

Article

A Lightweight Fingerprint Recognition Pipeline Based on Hierarchical Energy-Feature Decomposition

Yajuan Sun ¹, Maolin Li ², Qinge Wu ^{2,*}  and Shuyan Wu ^{1,3,*}¹ College of Information Engineering, Henan University of Animal Husbandry and Economy, Zhengzhou 450046, China; 80374@hnuahe.edu.cn² School of Computer and Information Engineering, Shanghai Polytechnic University, Shanghai 201209, China; 20251513019@sspu.edu.cn³ Henan Key Laboratory of Livestock and Poultry Genetic Improvement and Healthy Breeding, Zhengzhou 450046, China

* Correspondence: qewu@sspu.edu.cn (Q.W.); 80042@hnuahe.edu.cn (S.W.)

Abstract

Fingerprint recognition remains challenging when ridge structures are degraded by noise, weak contrast, translation, rotation, and local deformation during acquisition. Although deep-learning approaches have improved biometric recognition, they often require large labeled datasets and carefully specified training protocols, which can limit their use in small-data or resource-constrained scenarios. This paper presents a lightweight fingerprint recognition pipeline based on hierarchical energy-feature decomposition. The pipeline integrates Template Integrated Mean (TIM) preprocessing, region-of-interest localization, coefficient-feature extraction, energy-feature extraction, and two-stage template matching. Coefficient features are used for coarse candidate screening, whereas energy features are used for fine matching within the reduced candidate set. On the evaluated fingerprint dataset, the proposed method achieves a closed-set identification accuracy of 97.86% under the reported gallery/probe protocol. Additional aggregate-level statistical checks and baseline configuration details are provided to clarify the evaluation scope and comparison protocol.

Keywords: fingerprint recognition; closed-set identification; hierarchical feature representation; energy-feature decomposition; template matching; image preprocessing

1. Introduction

Fingerprint recognition is widely used in biometric authentication for access-control systems, online identity verification, secure terminals, financial services, and intelligent information infrastructures. Practical applications include identity management in health and security systems, ID-card verification, securities trading, examinations, temporary residence management, access control, e-commerce, salary management, confidential rooms, key-protected units, driver authentication, and child-transfer management [1]. In these scenarios, recognition methods must distinguish identity reliably while maintaining low computational cost and transparent decision logic. This requirement is especially important for small-data or resource-constrained systems, where large training sets and complex model tuning may not be available. Therefore, lightweight and interpretable fingerprint recognition remains a practical research topic in secure identity recognition.

Recent studies have improved fingerprint recognition through enhanced feature representation, multimodal cues, and learning-based models. Texture-based image recognition has also improved in terms of accuracy and efficiency [2]. Fingerprint recognition, as a



Academic Editor: Kewei Sha

Received: 13 June 2026

Revised: 8 July 2026

Accepted: 15 July 2026

Published: 17 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

representative biometric texture-recognition task, has benefited from both local minutiae descriptors and global ridge-pattern representations [3]. Zhang et al. [4] proposed an incomplete fingerprint recognition method based on feature fusion and pattern entropy. Jiang et al. [5] developed a fingerprint recognition technique using long digital line segments. Kizrak et al. [6] introduced a fingerprint recognition algorithm based on median filtering. Nanni et al. [7] formulated fingerprint verification as a binary classification problem. Moros et al. [8] used laser-induced breakdown spectroscopy and machine learning to identify explosive residues in fingerprints on object surfaces. Lin et al. [9] combined optical sensors and fractal classifiers for fingerprint recognition. Wang et al. [10] used high-performance liquid chromatography fingerprinting and pattern recognition to identify and differentiate Chinese medicinal materials. These studies demonstrate the breadth of feature-based recognition, but low-quality fingerprint images, partial impressions, weak ridge structures, and limited training data still affect recognition stability in practical biometric authentication systems.

Segmentation and preprocessing are essential before reliable fingerprint templates can be extracted. Xie et al. [11] constructed a two-stage fast segmentation method based on multi-scale texture features and non-local active contours. Edward et al. [12] used bi-binary complex wavelets to extract rotation-invariant texture features. Zhou et al. [13] fused graph cuts and active contours for texture-aware segmentation. Yu introduced gray-level co-occurrence matrix information and spatial information into fuzzy C-means clustering [14]. Long et al. [15] constructed a multi-scale segmentation framework based on a contourlet hidden Markov model. Inyang et al. [16] proposed an ANFIS-based hybrid system for texture classification. Wu et al. [17] further explored texture segmentation and showed the potential of texture-based feature analysis. These methods improve region localization and texture representation, but they may be sensitive to initialization, parameter selection, image noise, or scale variation. For fingerprint images, preprocessing should suppress noise and irrelevant background regions while preserving ridge-valley structures needed for matching.

Learning-based methods have further expanded the representation ability of image restoration and biometric recognition. Wei et al. [18] proposed a robust corner detector in continuous space, which improves feature-point localization for texture analysis. Liu et al. [19] used multi-scale image patches and sparse representation models to enhance latent fingerprints. Scabini et al. [20] compared visual Transformer architectures for texture feature extraction and reported their advantages in complex texture modeling. Li et al. [21] proposed a structure-preserved network combined with a residual diffusion model for detail-aware image denoising. Ghalati et al. [22] reviewed texture-analysis applications in biomedical imaging and demonstrated the broader value of texture descriptors. Zhang et al. [23] proposed DnCNN for image denoising. Zhang et al. [24] further developed FFDNet for flexible and fast image denoising. Liang et al. [25] proposed SwinIR based on a shifted-window Transformer architecture for image restoration. Zamir et al. [26] introduced Restormer for efficient Transformer-based restoration of high-resolution images. Although these learning-based methods are powerful, they often require large training datasets, careful training protocols, and matched testing domains. Their black-box nature and computational cost may also limit their use in small-data or resource-constrained fingerprint authentication scenarios.

Recent work has also explored hybrid features, neural rendering, lightweight deployment, and image-quality enhancement for recognition systems. Artabaz and Sliman compared handcrafted-feature fusion and deep-learning strategies in multimodal hand biometric recognition, providing useful ideas for hierarchical biometric representation [27]. Huang et al. [28] proposed NeRF-Texture for synthesizing high-quality neural radiance

field textures. Zhang et al. [29] implemented lightweight neural networks on edge devices, offering a reference for embedded recognition systems. Cho et al. [30] applied a generative adversarial network to blur recovery and improved the recognition performance of low-quality fingerprint images. However, end-to-end learning methods remain dependent on the scale of labeled data, lightweight models may sacrifice representation ability, and generative restoration methods may suffer from unstable training or weak controllability. For secure biometric authentication in information infrastructures, a practical fingerprint recognition workflow should integrate robust preprocessing, compact identity-related representation, and efficient template matching. This study addresses this need by organizing these components into a unified, lightweight fingerprint authentication workflow.

The fingerprint recognition pipeline studied in this paper consists of three main stages: image preprocessing, feature representation, and hierarchical matching. In the preprocessing stage, acquired fingerprint images are enhanced and divided into regions of interest. In the feature representation stage, bifurcation and intersection points are extracted, and coefficient and energy features are constructed through hierarchical energy-feature decomposition. In the matching stage, coefficient features are used for coarse candidate screening, and energy features are used for fine matching through a distance-based decision rule. The overall workflow is shown in Figure 1.

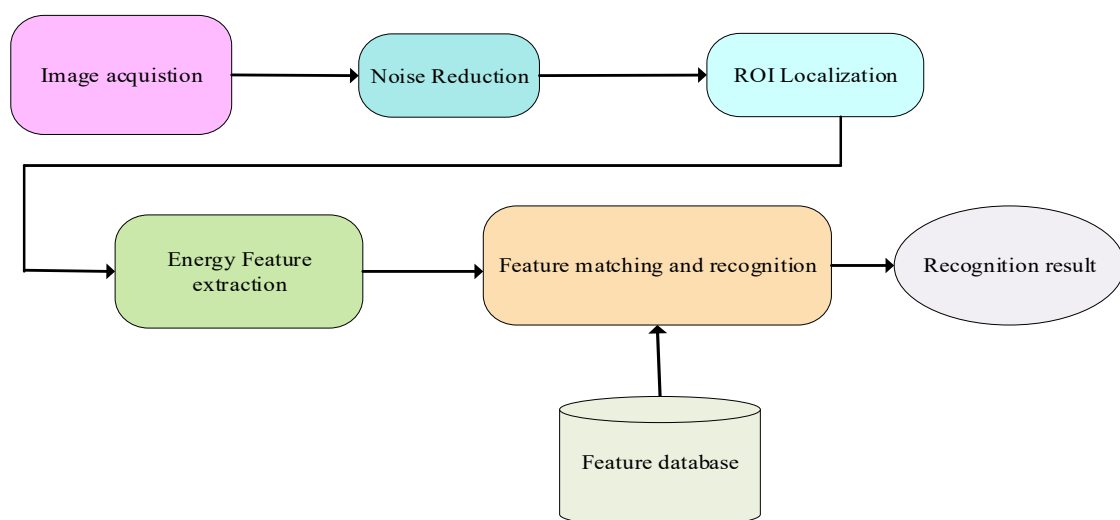


Figure 1. Construction flow of the proposed closed-set fingerprint recognition pipeline.

In summary, although various fingerprint recognition methods have been proposed, there remains a need for lightweight and interpretable workflows that can handle noisy fingerprint images under limited-data conditions. The main contributions of this paper are fourfold: (1) a reproducible TIM preprocessing formulation for fingerprint denoising; (2) a hierarchical coefficient-and-energy feature representation for fingerprint templates; (3) a two-stage matching rule that uses coefficient features for candidate screening and energy features for final matching; and (4) an explicitly specified gallery/probe evaluation protocol with baseline settings and aggregate-level statistical comparison. These contributions position the method as a lightweight fingerprint recognition pipeline and provide a clearer basis for later extension to verification and open-set authentication scenarios.

2. Feature Analysis of Fingerprint Images

Fingerprint recognition determines identity by comparing discriminative local minutiae and global ridge-pattern features across fingerprint images. Because different fingers exhibit different ridge structures, bifurcations, endpoints, and ridge orientations, these

features can be encoded as biometric templates for authentication. Compared with many other identity-recognition techniques, fingerprint recognition offers stable physical characteristics, convenient acquisition, and relatively low implementation cost.

A fingerprint consists of the ridge patterns on the skin of the fingertip, which increase friction to assist in grasping objects. Though merely a small skin region, a fingerprint holds abundant information. The ridge morphology, including break points and intersections, differs for every individual; these unique skin patterns are termed “features” in information processing. Medical evidence confirms that fingerprint patterns are unique and remain invariant over a person’s lifetime, thus offering strong uniqueness and stability. This allows a person to be matched with his fingerprint by comparing a query print with previously stored records to verify identity. As a result, identification based on ridge features has become a central focus of fingerprint recognition research. While recent work has drawn significant interest and made notable advances, continued exploration is still required.

Compared with many other recognition techniques, fingerprint recognition benefits from relatively stable ridge patterns and convenient image acquisition. In practical environments, however, the captured image can still be affected by pressure, rotation, translation, dry or wet fingers, sensor differences, and background noise. Therefore, reliable recognition requires preprocessing and feature extraction methods that preserve discriminative ridge structures while reducing acquisition artifacts.

High-resolution fingerprint sensors capture rich ridge and valley details, but they also include background regions that are not useful for matching. This paper therefore extracts a region of interest before feature analysis. The ROI step focuses subsequent processing on informative ridge structures and reduces the influence of irrelevant background pixels. Figure 2 illustrates two representative fingerprint minutiae, namely the endpoint and the intersection.

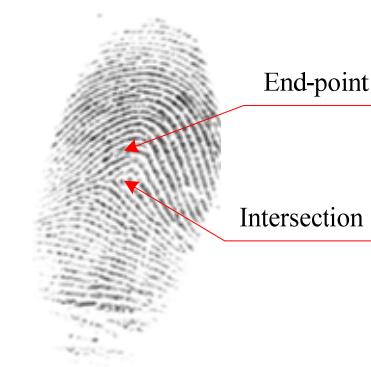


Figure 2. Detailed features of a fingerprint.

The SUSAN operator is used to detect candidate corner points. However, some edge points may also be detected as candidate points. Therefore, a circular transition analysis is further used to distinguish endpoints and intersection points.

For each candidate point O , several concentric circles are constructed with O as the center. The pixels on each circle are compared with the gray value of O . Pixels with similar gray values are assigned ‘1’, while the others are assigned ‘0’. Then, the binary sequence on each circle is scanned clockwise, and the number of transitions between ‘0’ and ‘1’ is counted. If the number of transitions is exactly two, the candidate point is regarded as an endpoint. If the number of transitions is greater than two, it is regarded as an intersection point. Other points are treated as edge or non-feature points and are discarded.

3. Fingerprint Image Preprocessing

The quality of acquired fingerprint images is often degraded by noise, background interference, translation, and rotation during capture. These artifacts hinder feature extraction and template matching. Therefore, segmentation, calibration, normalization, and denoising are performed before feature extraction.

3.1. Image Noise Reduction Processing

3.1.1. Noise Type Analysis

Fingerprint images are susceptible to various noise interferences during acquisition, primarily including: Gaussian noise, caused by sensor thermal and electronic noise, which follows a normal distribution; salt-and-pepper noise, caused by defects in the acquisition equipment, dust, or dry skin surface, manifesting as randomly distributed bright and dark spots; speckle noise, caused by uneven skin moisture, changes in lighting, so on; and quantization noise, the error generated when converting analog signals to digital signals. The mixed-noise model is summarized in Equation (1).

$$I_{noisy}(m, n) = I_{clean}(m, n) + N_G(m, n) + N_{SP}(m, n) + N_Q(m, n) \quad (1)$$

where N_G is Gaussian noise, N_{SP} is salt-and-pepper noise, and N_Q is quantization noise. While deep learning-based denoising methods (e.g., DnCNN, U-Net with attention) have shown impressive performance, they often require large training sets, struggle with unseen noise patterns, and may over-smooth fine ridge/valley structures critical in fingerprint analysis. Furthermore, generative models (e.g., GANs) risk introducing hallucinated textures. Hence, in scenarios with limited training data and high demand for texture fidelity—such as forensic or legacy datasets—a non-learning, highly interpretable approach remains valuable.

3.1.2. Proposed Noise Reduction Algorithm

This paper uses a Template Integrated Mean (TIM) method for fingerprint image denoising. TIM is a deterministic local filtering operation that integrates multi-scale template statistics and an adaptive residual decision rule. The mathematical formulation of TIM is given in Equations (2)–(5), including local template mean estimation, multi-scale fusion, residual calculation, and residual-threshold-based pixel replacement. Let $I(x, y)$ denote the input grayscale fingerprint image, and let $W_s(x, y)$ be a square local window centered at pixel (x, y) , where s in $\{3, 5, 7\}$. The local template mean at scale s is defined in Equation (2):

$$T_s(x, y) = |W_s|^{-1} \sum_{(u, v) \in W_s(x, y)} I(u, v) \quad (2)$$

The fused local estimate and residual are then calculated using Equations (3) and (4):

$$T(x, y) = \sum_s w_s T_s(x, y), w_s \geq 0, \sum_s w_s = 1 \quad (3)$$

$$r(x, y) = |I(x, y) - T(x, y)| \quad (4)$$

Given two thresholds τ_l and τ_h , the denoised image Q is obtained by the residual-based rules in Equation (5):

$$Q(x, y) = \begin{cases} T_7(x, y), & r(x, y) > \tau_h \\ T_3(x, y), & \tau_l < r(x, y) \leq \tau_h \\ I(x, y), & \text{otherwise} \end{cases} \quad (5)$$

Here, $T_3(x,y)$ and $T_7(x,y)$ denote the local template means computed with 3×3 and 7×7 windows, respectively; the 5×5 template contributes to the fused estimate $T(x,y)$ in Equation (3). In this implementation, equal weights are used unless otherwise specified. Reflection padding is used at image boundaries. The thresholds are estimated from the residual distribution of the reference/training images and are not tuned using recognition test results. This residual-based rule allows TIM to handle mixed noise without assuming a single noise distribution.

The TIM denoising procedure is summarized in Algorithm 1.

Algorithm 1. Template Integrated Mean (TIM) denoising.

Item	Description
Input	Grayscale fingerprint image I , window sizes $S = \{3, 5, 7\}$, weights w_s , and thresholds τ_l and τ_h .
Procedure	Step 1: Compute local template means T_3, T_5 , and T_7 for every pixel. Step 2: Compute the fused local estimate T and residual r . Step 3: Replace strong outliers with the large-window estimate, correct moderate deviations with the small-window estimate, and retain pixels with small residuals.
Output	Denoised image Q .

The method contains no trainable neural parameters and is therefore reproducible from the stated window sizes, weights, thresholds, and boundary rule.

3.1.3. Noise Reduction Performance Evaluation Indicators

Three objective evaluation metrics were used. For the signal-to-noise ratio (PSNR), it was used to evaluate the resemblance between the denoised image and the original clean image.

$$\text{PSNR} = 10 \log_{10} \left(\frac{255^2}{\text{MSE}} \right)$$

$$\text{MSE} = \frac{1}{MN} \sum_{x=1}^M \sum_{y=1}^n [P(i, j) - Q(i, j)]^2$$

Similarity is used to assess the degree to which structural information of an image is preserved.

$$\text{SSIM}(A, B) = \frac{(2\mu_A\mu_B + C_1)(2\sigma_{AB} + C_2)}{(\mu_A^2 + \mu_B^2 + C_1)(\sigma_A^2 + \sigma_B^2 + C_2)}$$

where μ is the mean, σ is the standard deviation, and C_1 and C_2 are constants.

Mean absolute error is used to measure the average pixel difference between the denoised image and the original image.

$$\text{MAE} = \frac{1}{MN} \sum_{x=1}^M \sum_{y=1}^n |P(x, y) - Q(x, y)|$$

where P denotes the original clean image, Q denotes the denoised image, M and N denote the image dimensions, and $|\cdot|$ denotes the absolute value operation.

3.2. Noise Reduction Simulation Experiment and Result Analysis

3.2.1. Data Source

FVC2004 was used for the denoising and preprocessing evaluation. It contains four subsets, DB1, DB2, DB3, and DB4, acquired or generated under different sensor conditions: DB1 uses the Crossmatch V300 optical sensor, DB2 uses the Digital Persona U.are.U

4000 optical sensor, DB3 uses the Atmel FingerChip thermal sensor, and DB4 contains synthetic fingerprints. Each subset contains 10 fingers with eight images per finger. For PSNR, SSIM, and MAE calculation, the original FVC2004 image was treated as the clean reference image, and degraded inputs were generated by adding controlled synthetic noise, including Gaussian noise, salt-and-pepper noise, and speckle-like multiplicative noise. The same degraded inputs and the same noise settings were used for all compared denoising methods. Therefore, the denoising metrics evaluate performance under a controlled synthetic-degradation benchmark. The degradation protocol is summarized in Table 1.

Table 1. Controlled degradation protocol for the denoising evaluation.

Item	Setting in This Study	Purpose
Reference image	Original FVC2004 image before synthetic degradation	Provides the reference for PSNR, SSIM, and MAE calculation
Gaussian noise	Controlled additive Gaussian noise; the same setting was applied to all compared denoising methods	Simulates sensor/electronic noise under a reproducible condition
Salt-and-pepper noise	Controlled impulse noise; the same setting was applied to all compared denoising methods	Simulates isolated bright/dark pixel corruption
Speckle-like noise	Controlled multiplicative noise; the same setting was applied to all compared denoising methods	Simulates local intensity disturbance
Metric calculation	Denoised image compared with the corresponding pre-degradation reference image	Ensures that PSNR/SSIM/MAE are computed against a traceable reference

3.2.2. Comparison of Images Before and After Noise Reduction

To illustrate the denoising behavior of the proposed TIM method, sample 101_2.tif from FVC2004 was used as a representative example. The compared methods were applied to the same degraded input image, and the qualitative comparison is shown in Figure 3.

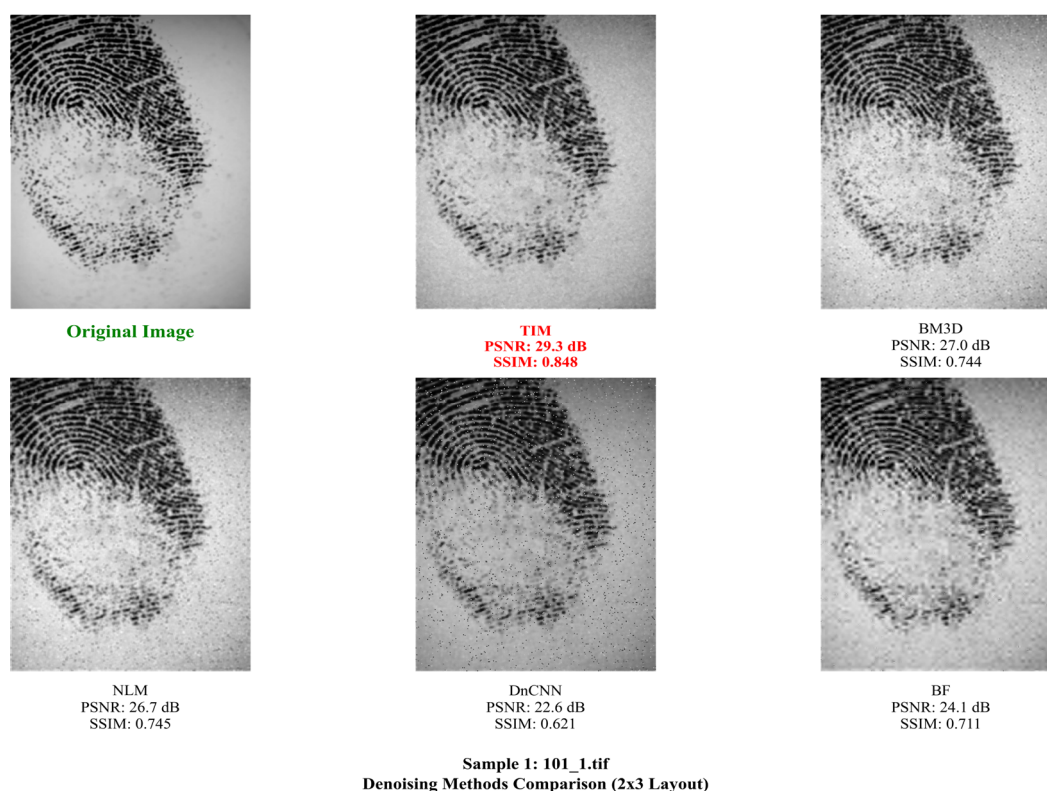


Figure 3. Comparison of the proposed TIM method with existing methods.

Figure 3 provides a qualitative comparison of denoising results produced by the evaluated methods. The TIM result preserves clearer ridge continuity and fewer visible noise artifacts in the selected example. The averaged quantitative results are reported in Table 2 and Figures 4 and 5.

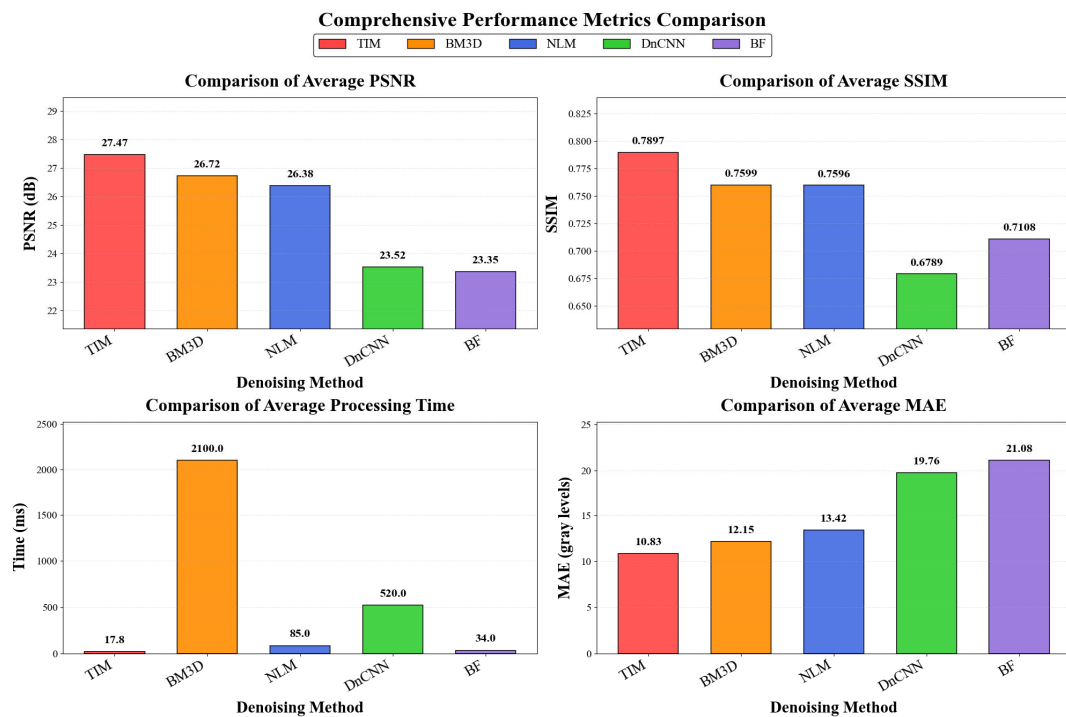


Figure 4. Comparison of average performance for different noise reduction methods.

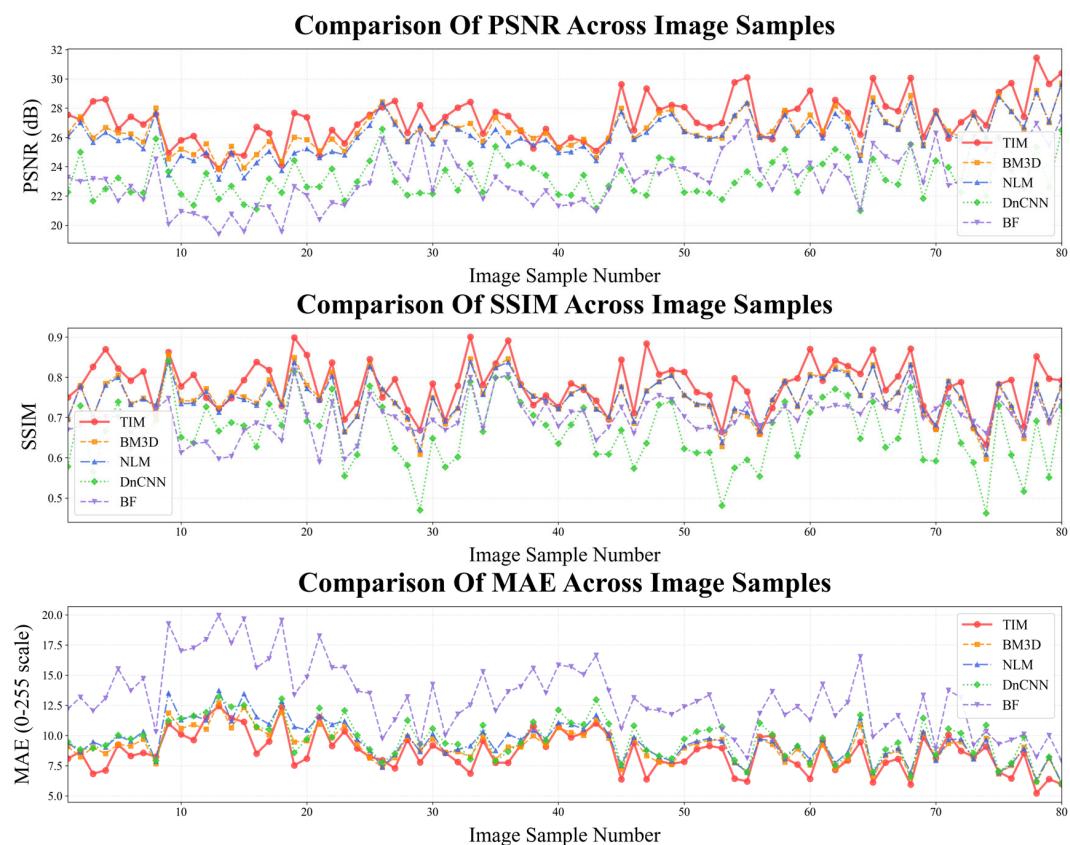


Figure 5. Comparison of signal-to-noise ratio and similarity of different noise reduction methods.

3.2.3. Performance Comparison of Noise Reduction Methods

To evaluate the performance of the proposed Template Integrated Mean (TIM) method in fingerprint image denoising, four representative methods were selected for comparison: BM3D, DnCNN, non-local means filtering, and bilateral filtering. These methods represent classical model-based denoising, learning-based denoising, and local/non-local filtering strategies. All methods were evaluated using the same degraded inputs and the same reference images. Parameter settings followed commonly used default values when no dataset-specific tuning was available.

Table 2. Comparison of performance for different noise reduction methods.

Method	Average PSNR (dB)	Average SSIM	Average MAE	Average Time (ms)
TIM	27.47	0.7897	10.83	17.8
BM3D	26.72	0.7599	12.15	2100
NLM	26.38	0.7596	13.42	85
DnCNN	23.52	0.6789	19.76	520.0
BF	23.35	0.7108	21.08	34

The experimental results indicate that TIM achieves the highest observed average PSNR and SSIM among the compared methods under the reported controlled benchmark. As shown in Table 2 and Figure 4, TIM obtains an average PSNR of 27.47 dB, an average SSIM of 0.7897, and an average MAE of 10.83. The method also shows lower processing time than the compared alternatives in this experimental setting. These results suggest that TIM can preserve ridge structures while suppressing synthetic noise in the evaluated benchmark.

Overall, the proposed TIM achieves optimal or near-optimal performance across all four dimensions: denoising quality, structure preservation, computational efficiency, and stability. It strikes an effective balance between high denoising quality and low computational overhead, making it particularly suitable for resource-constrained embedded fingerprint recognition systems. Compared with deep-learning methods, the proposed method requires neither a GPU nor training data and possesses domain adaptability. Compared with traditional methods, it achieves higher denoising accuracy through multi-scale templates and adaptive thresholding strategies.

3.3. Feature Preprocessing

(1) Extraction of fingerprint edge contour

An image boundary represents a discontinuity in local image properties, such as abrupt variations in gray level, color, or texture, marking the transition from one region to another. For computational convenience, first- or second-order derivatives are commonly adopted to detect these boundaries, as derivative-based methods readily capture gray-level discontinuities. Edge detection is typically implemented through convolution with spatial differential operators. Prior to boundary extraction, appropriate smoothing and binarization steps must be applied.

Assume the original fingerprint image is $f'(x, y)$. First, smooth filtering is performed in the spatial domain to suppress noise and enhance the continuity of fingerprint ridges. After filtering, a relatively complete fingerprint edge image can be obtained. Then, edge detection is carried out on the binarized image along both the image-width and image-height directions by using gray-level mutation information. Specifically, abrupt gray-level changes between neighboring pixels are used to locate potential fingerprint edge points, and the final edge map is obtained by combining the detection results from the two directions.

Based on these prior works, a fitting curve that the position points x_k and y_k satisfy can be derived. The borderline of the fingerprint region is then obtained through texture boundary modeling. This texture segmentation technique is referred to as the gray-level mutation algorithm. Figure 6 illustrates the extraction process of fingerprint edges using this mutation algorithm and provides a comparison between the proposed algorithm and existing fingerprint extraction methods.

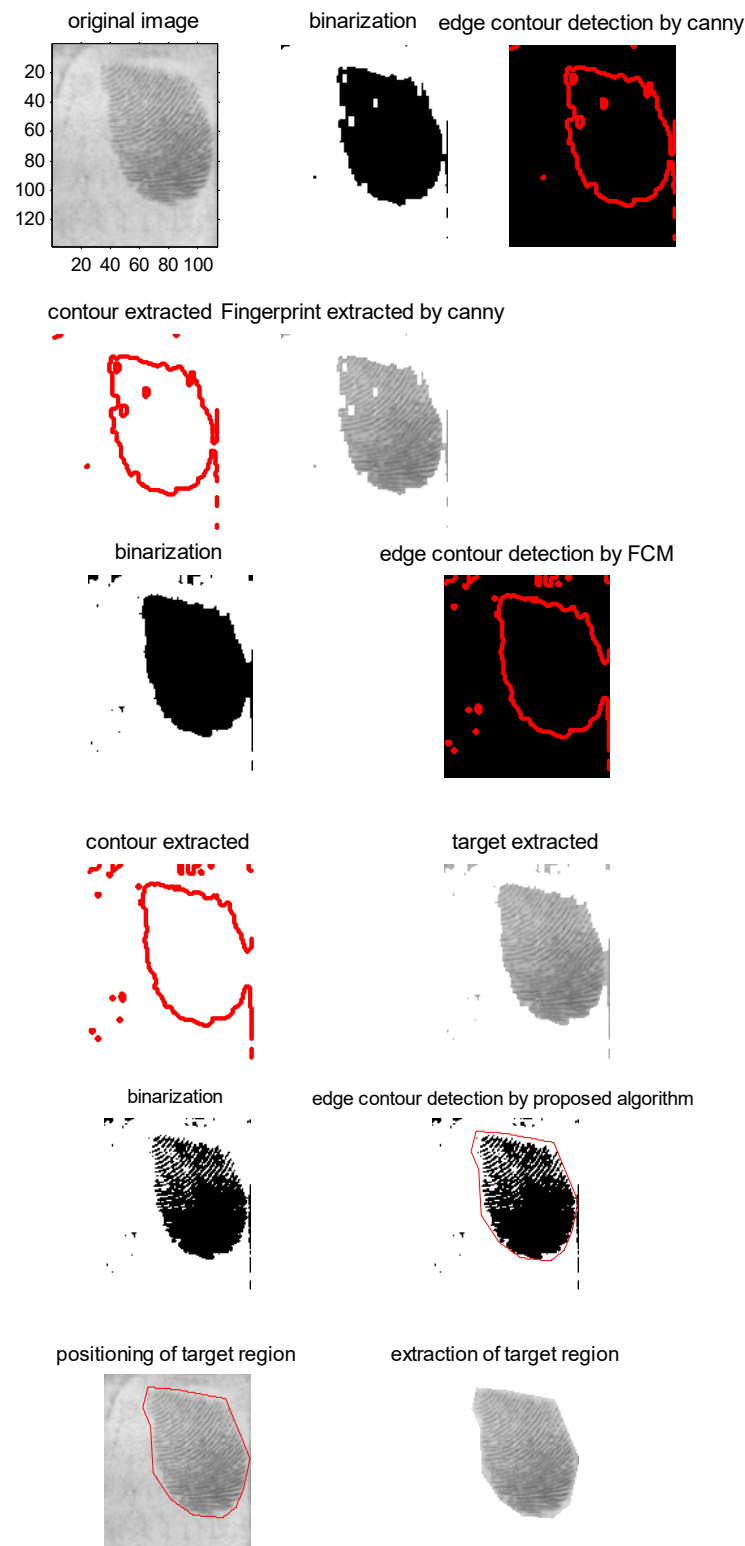


Figure 6. Extraction of fingerprint edge contour by several algorithms.

As shown in Figure 6, the Canny and FCM algorithms produce contours with many marginal branches, whereas the proposed method yields smoother edge contours that more accurately outline the fingerprint region.

(2) Extraction of the interest area

This section gives the method for the extraction of the interest area as follows:

In the coordinate system shown in Figure 7, in order to efficiently isolate a fingerprint region that is both informative and easy to process, this paper presents a rapid segmentation method. The procedure is as follows: first, along the OP direction along x axis, a sharp increase h_1 and a sharp decrease h_2 in gray-level values are detected, thereby determining the left boundary l_1 and right boundary h_2 of the region of interest. Next, along the OQ direction along y axis, similar sharp variations w_1 and w_2 are detected to establish the upper boundary l_3 and lower boundary l_4 . These abrupt changes in gray-level values are related to the pixel width of the fingerprint edge, the edge height, and the edge position; that is, the mutation of gray-level values is determined by these two factors.

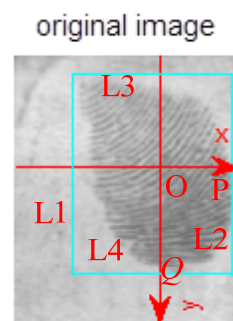


Figure 7. Construction of a coordinate system.

The area of the image falling within this region is extracted as the interest area, as shown in Figure 8. The interest area obtained in this way has eliminated the influence of noise, rotation, translation, and so on. Experimental results show that the ROI extracted in this way is more suitable for subsequent fingerprint feature extraction.

extraction of target region by proposed algorithm



Figure 8. Positioning and extraction of the target region.

To clearly illustrate each stage of the ROI extraction pipeline, Figure 9 presents a comprehensive visualization of the process. The boundary scan map is first computed to distinguish the high-texture fingerprint region from the smooth background. A binary mask is then generated through adaptive thresholding and morphological operations.

Finally, the extracted ROI undergoes contrast normalization and median filtering for ridge enhancement.

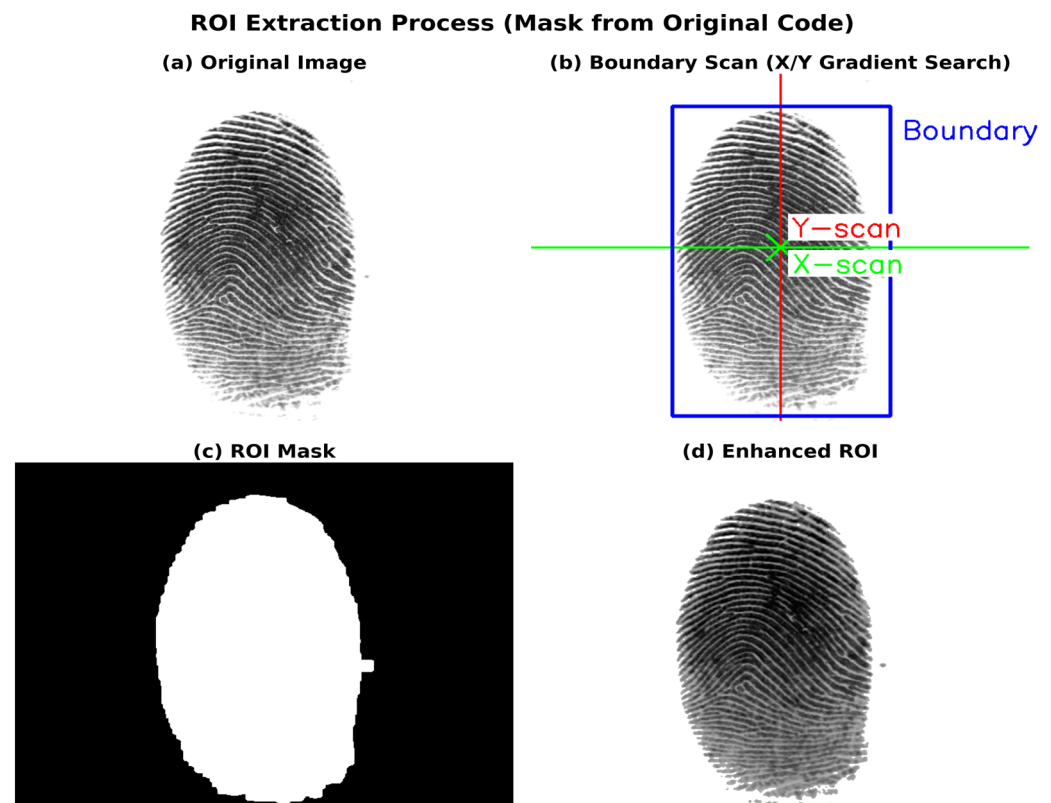


Figure 9. Visual illustration of the ROI extraction process.

The segmentation algorithm produces a sufficiently large region of interest that almost fully retains the ridge structure, pattern, bifurcations, and endpoints, while delivering sharper papillary ridges. This directly boosts the accuracy of downstream feature extraction and matching. However, the method demands a high-end capture device to record high-quality images and requires a large-scale, high-fidelity fingerprint database for conducting experiments.

In addition, a comparative evaluation of the proposed gray-level mutation-based extraction algorithm against existing methods is presented, as shown in Figure 10. The proposed approach clearly demonstrates superior extraction quality, as it almost entirely removes the background, suppresses noise, and comprehensively retains the fingerprint information.

3.4. Extraction of Characteristic Points

(1) Adjustment of gray

To obtain a clearer texture pattern of the finger, the original RGB image is first converted to grayscale, and its intensity range is then stretched from $[0.4, 0.6]$ to $[0, 1]$. Figure 11 displays the histograms before and after this adjustment. The histogram reveals that the transformed image possesses a uniform grayscale distribution, thereby producing an image with enhanced texture detail.

(2) Detection of characteristic points

The above processing steps produce a clear-textured fingerprint image, significantly easing the task of feature point detection and tracking. Beyond edge detection, the gray-level jump algorithm outlined earlier is also capable of locating corner points, endpoints,

and bifurcations. An example of corner point detection based on this algorithm is shown in Figure 12.

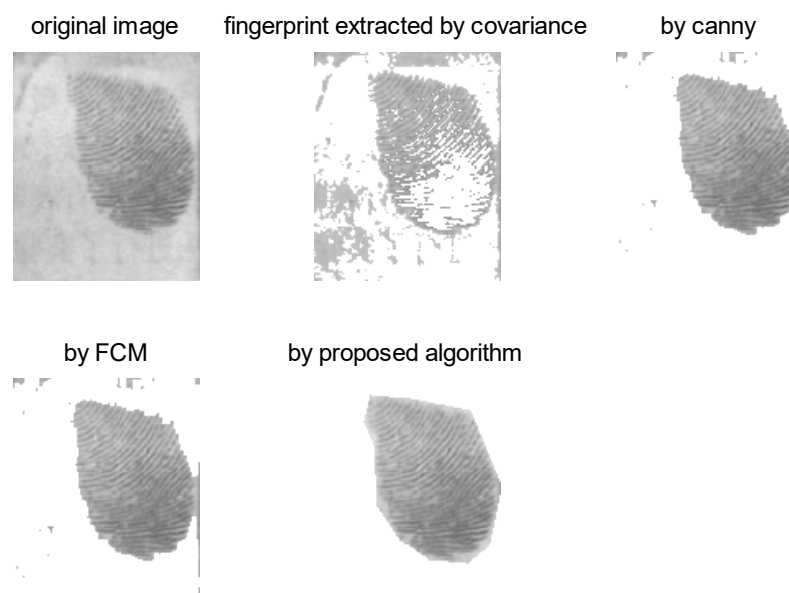


Figure 10. Comparison of fingerprint extraction.

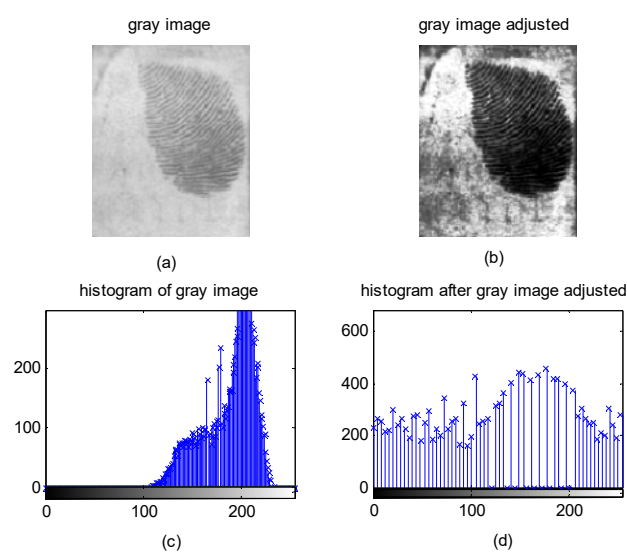


Figure 11. Grayscale range adjustment and histogram comparison. (a) Original grayscale fingerprint image. (b) Fingerprint image after grayscale range adjustment. (c) Histogram of the original grayscale image. (d) Histogram after grayscale range adjustment.

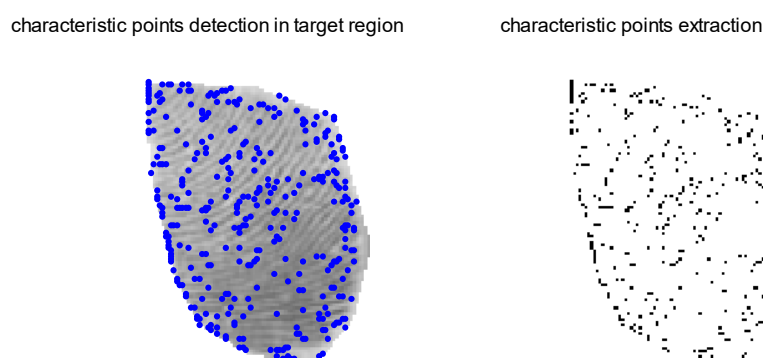


Figure 12. Detection and extraction of corner points based on the gray-level jump algorithm.

The characteristic points are the endpoints, corner points, bifurcation points, etc., which the papillary ridges form on the fingerprint. The type of corner points detected by the gray level jump algorithm is varied, so it requires filtering the detected corner points. A circle with a sufficiently small radius is centered at each detected corner point, and the gray-level variations along its circumference are examined. A point where three abrupt increases or decreases in gray level occur is classified as a characteristic point.

(3) Energy Decomposition and Feature Point Visualization

To provide an intuitive understanding of the energy-based feature extraction mechanism, Figure 13 presents the multi-orientation Gabor decomposition process and the resulting energy spectrum analysis. The original ROI is convolved with a bank of 8-direction Gabor filters (0° to 157.5° with 22.5° step size). The total energy map aggregates responses across all orientations, while individual orientation responses reveal directional texture characteristics.

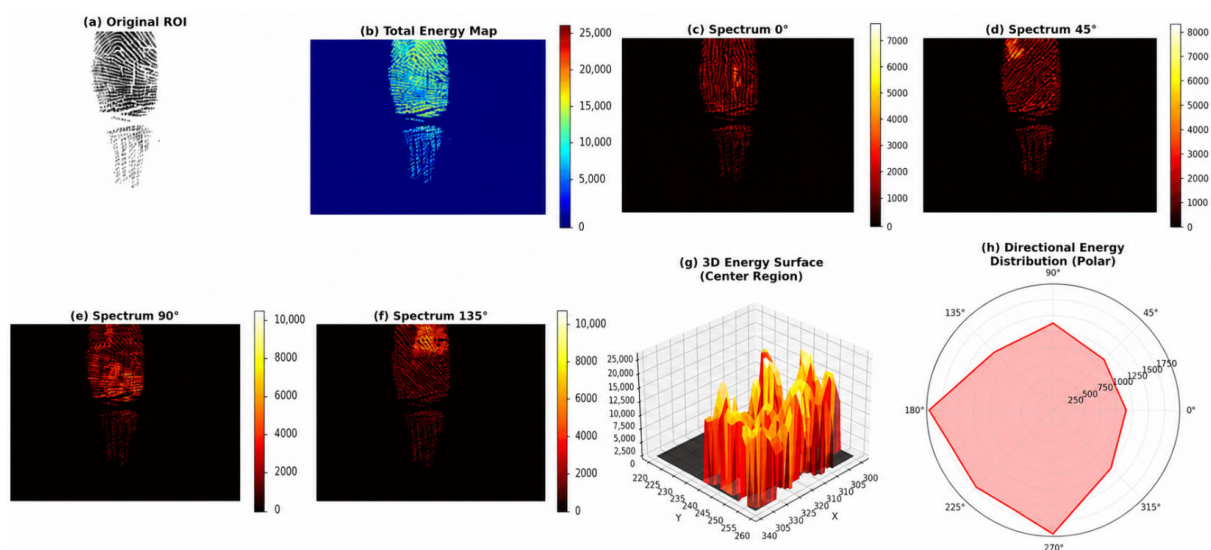


Figure 13. Gabor-based coefficient and energy response maps of the fingerprint ROI.

4. Hierarchical Fingerprint Feature Representation

In this study, coefficient decomposition, Gabor decomposition, and energy decomposition are treated as consecutive steps of one unified feature-extraction procedure. A multi-orientation Gabor filter bank is first applied to the fingerprint ROI to obtain coefficient response maps. The coefficient feature vector is constructed from normalized statistics of these response maps, whereas the energy feature vector is constructed from the squared response magnitudes. Therefore, the coefficient and energy features are complementary descriptors derived from the same Gabor response bank rather than two unrelated feature representations.

Fingerprint feature extraction is performed using the energy decomposition transform. Choose a function $\psi(\vec{t})$ to construct $\psi_{j,k}(\vec{t}) = 2^{j/2} \psi(2^j \vec{t} - \vec{k})$, and make it do an inner product with the image $p(x, y)$, i.e., $\langle \psi_{j,k}(\vec{t}), p(x, y) \rangle$, which can carry out some processing for a fingerprint image, such as smoothing, denoising, enhancing, compressing, and so on, where $\vec{t} = (x, y)$, $\vec{j} = (j_1, j_2)$, $\vec{k} = (k_1, k_2)$. An alternative way is usually to choose a two-dimensional Gabor function as the coefficient decomposition, as shown in Equation (6):

$$g(x, y) = \frac{1}{2\pi\delta_x\delta_y} \cdot e^{-\frac{1}{2}[(\frac{x}{\delta_x})^2 + (\frac{y}{\delta_y})^2] + j(ux + vy)} \quad (6)$$

The image $p(x, y)$ is processed by the specific algorithm $W_{\vec{j}, \vec{k}} = \langle \psi_{\vec{j}, \vec{k}}(\vec{t}), p(x, y) \rangle$, which performs decomposition and extraction of energy coefficient features. For texture regions with rich features, the energy becomes denser, the amplitude difference remains small, and the energy distribution grows more regular as the decomposition level increases. Conversely, for regions with sparse texture, the energy exhibits a jumping and scattered pattern with deeper decomposition levels.

Based on the decomposition computation of coefficients, the i th level energy feature is derived from the i th level energy decomposition coefficients. This energy feature reflects the detection of texture characteristic points during the extraction of target region characteristic points in a fingerprint image at the 2^{-i} scale, across different directions and positions.

4.1. Coefficient-Based Feature Extraction

Assuming $g_j^n(t) \in U_j^n$, it can be expressed as $g_j^n(t) = \sum_l d_l^{j,n} \psi_n(2^j t - l)$. According to the calculation of energy decomposition, the decomposition coefficient is given by Equation (7):

$$\begin{cases} d_l^{j,2n} = \sum_k a_{k-2l} d_k^{j+1,n} \\ d_l^{j,2n+1} = \sum_k b_{k-2l} d_k^{j+1,n} \end{cases} \quad (7)$$

where $a_n = \frac{1}{2} \bar{p}_n$ and $b_n = \frac{1}{2} \bar{q}_n$, $\bar{p}_n$ and $\bar{q}_n$ are the conjugation of p_n and q_n , respectively.

By the calculation of energy reconstruction, the reconstruction coefficient is given by Equation (8):

$$d_l^{j+1,n} = \sum_k [p_{l-2k} d_k^{j,2n} + q_{l-2k} d_k^{j,2n+1}] \quad (8)$$

Equations (7) and (8) are used here to describe the recursive organization of decomposition coefficients, rather than to introduce an independent wavelet reconstruction module. In the proposed implementation, the coefficient and energy features are both derived from the Gabor response maps described above.

First, the preprocessed images undergo a two-level decomposition via the energy decomposition function $\psi(t)$, with results based on a given energy basis shown in Figure 14. The function outputs $\psi_n(t)$ a tree structure of energy decomposition that serves as a user interface, allowing detailed node images to be viewed by clicking individual nodes. Second, the energy decomposition coefficients at each node are calculated using the reconstruction coefficient function $d_l^{j+1,n}(t)$ at each layer, as presented in Figure 15. Finally, all coefficients are normalized to construct the characteristic vectors. The characteristic vector of energy decomposition coefficients at the layer i th is defined as X_i , given by Equation (9) below

$$X_i = (a_{i0}, a_{i1}, a_{i2}, \dots, a_{2^{2i}-1}) \quad (9)$$

where i denotes the number of layers of energy decomposition, a_j denotes the normalization coefficient values of energy decomposition at each layer.

4.2. Energy-Based Feature Extraction

The energy feature is computed using the energy decomposition results of the image. According to the algorithm described in this paper, the energy of a coefficient-decomposed image is defined in Equation (10):

$$V_{(i,j)} = \sum_{x=1}^{n \times 2^{-i}} \sum_{y=1}^{m \times 2^{-i}} [H_{(i,j)}(x, y)]^2 \quad (10)$$

where $V_{(i,j)}$ is the image energy at node (i, j) , with each node corresponding to a position in the quad-tree of the energy decomposition. For a two-level coefficient decomposition,

$i \in \{0, 1, 2\}$ and $j \in \{0, 1, 2, \dots, 15\}$. $H_{(i,j)}(x, y)$ denotes the energy of the detailed image at that node, and m, n are the image height and width. The energy feature of the coefficient-decomposed image is established by extracting the detailed image energy from each layer, as given in Equation (11):

$$V_i = (V_{(i,0)}, V_{(i,1)}, V_{(i,2)}, \dots, V_{(i,2^{2i}-1)}) \quad (11)$$

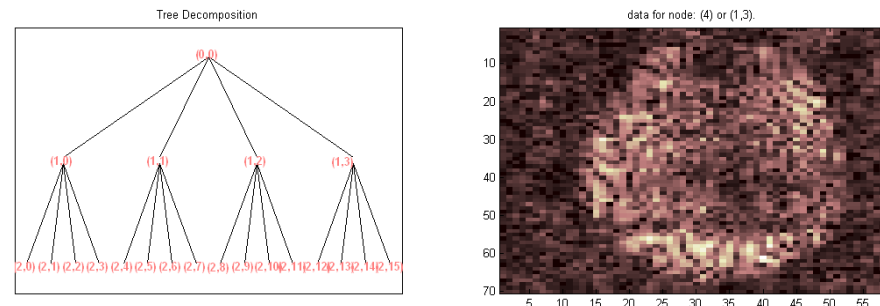


Figure 14. Nodes of energy decomposition and the detailed image at node (1, 3).

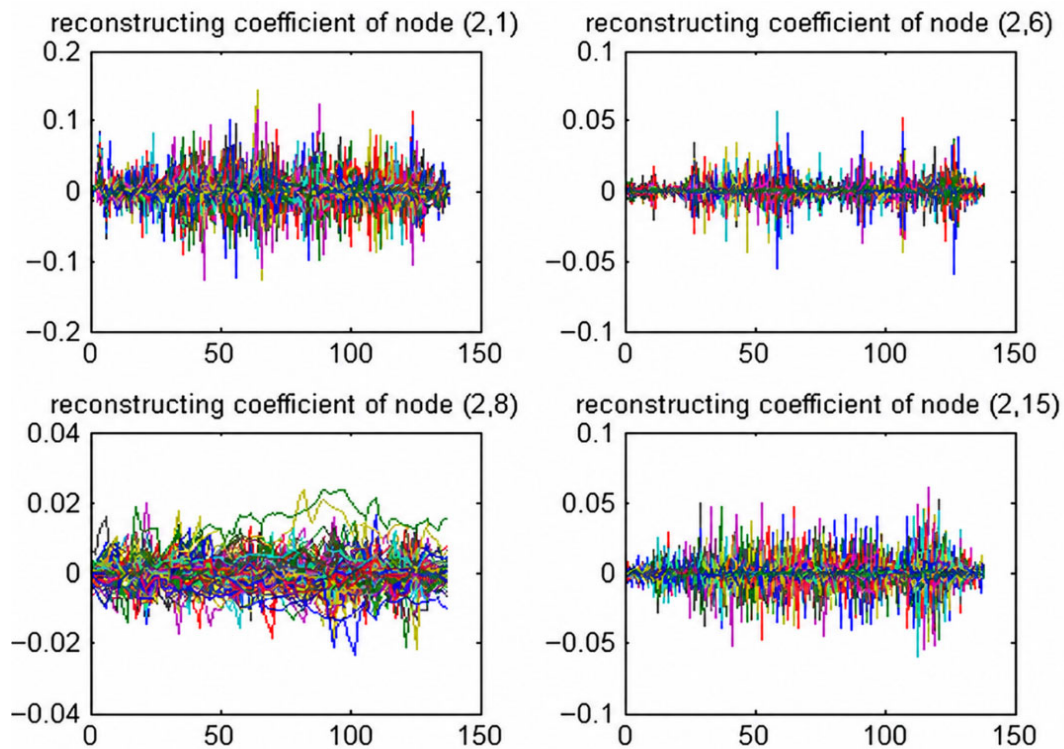


Figure 15. Reconstruction coefficients at selected nodes. Different colors indicate reconstructed coefficient components.

Therefore, E_i is referred to as the energy feature of the i th layer image decomposition, normalized as shown in Equation (12):

$$V_j = \frac{V_{(i,j)}}{\sum_{x=1}^n \sum_{y=1}^m [H_{(0,0)}(x, y)]^2} \quad (12)$$

where $j \in \{0, 1, 2, \dots, 2^{2i} - 1\}$. Further simplification can be obtained as Equation (13):

$$V_i = (V_0, V_1, V_2, \dots, V_{2^{2i}-1}) \quad (13)$$

4.3. Visualization of Extracted Minutiae Points

After computing the multi-orientation energy responses, local maxima are detected in each directional energy map and merged to form the final set of characteristic points. Figure 16 shows the extracted feature points overlaid on the original ROI image, marked with red asterisks.

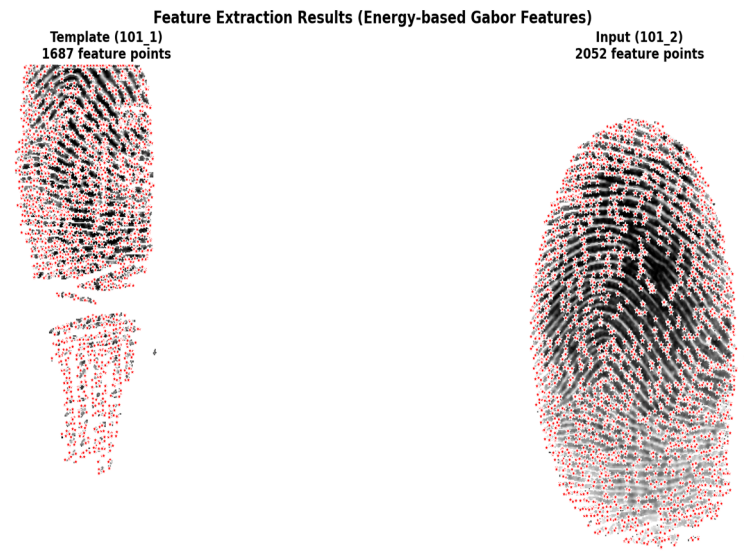


Figure 16. Extracted characteristic points from the fingerprint ROI.

5. Closed-Set Fingerprint Template Matching and Recognition

5.1. Biometric Template Database and Matching Strategy

To efficiently support fingerprint authentication, this study constructs a structured biometric template database for storing fingerprint images, coefficient features, energy features, and identity labels.

5.1.1. Fingerprint Database Construction and Storage

P1. Image Storage Structure: Original fingerprint images are stored in a hierarchical directory structure. For example, all image files for the user are stored under '/Database/User001/', named as 'User ID Serial Number.bmp'.

P2. Feature Vector Storage: Extracted feature vectors are stored in '.npy' format. Each user has a corresponding feature file containing the feature matrix of multiple images.

P3. Index File Creation: An 'index.csv' file is created to record information such as user ID, image path, feature file path, and acquisition device type for easy retrieval.

5.1.2. Hierarchical Template Matching

After preprocessing and feature extraction, each gallery fingerprint is represented by normalized coefficient and energy feature vectors. These vectors are stored as standard biometric templates and used for hierarchical matching.

P1. During the recognition process, the feature vectors of known fingerprint classes are first retrieved from the template library using database retrieval commands, such as index retrieval or primary key matching.

The query fingerprint is compared with the stored templates. Under the closed-set identification protocol used in this study, the query is assigned to the identity class whose template has the smallest matching distance after the two-stage matching process.

P3. If the matching result is positive, the system outputs the identity information corresponding to the fingerprint; if it is negative, it continues to match with other standard

templates until all comparisons are completed, thereby achieving recognition. The specific matching algorithm is as follows:

Let C_q and E_q denote the normalized coefficient and energy feature vectors of a query fingerprint q , and let C_i and E_i denote the corresponding vectors of the i -th gallery template. The coefficient distance $D_c(q, i)$ is used for coarse screening, and the energy distance $D_e(q, i)$ is used for fine matching. In this implementation, both distances are computed as Euclidean distances between the corresponding normalized feature vectors, as shown in Equation (14). The first stage selects a reduced candidate set according to $D_c(q, i)$, and the second stage identifies the final match according to $D_e(q, i)$.

$$d_l = \|V - W_l\| = \sum_{i=1}^n |V_i - W_{li}| \quad (14)$$

where $|V_i - W_{li}| = \frac{1}{2^{2i}} \sum_{j=0}^{2^{2i}-1} |c_{ij}^{V_i} - c_{ij}^{W_{li}}|$ represents the difference between the characteristic vectors of the coefficient for the image, or $|V_i - W_{li}| = \frac{1}{2^{2i}} \sum_{j=0}^{2^{2i}-1} |E_{ij}^{V_i} - E_{ij}^{W_{li}}|$ is the difference between the characteristic vectors of energy of the image decomposed by the coefficient. $c_{ij}^{V_i} \in V_i, c_{ij}^{W_{li}} \in W_{li}, E_{ij}^{V_i} \in E_i^{V_i}, E_{ij}^{W_{li}} \in E_i^{W_{li}}, i$ is the number of layers of the coefficient in the image decomposition. The recognition principle is described as follows:

If the final energy distance is below the selected threshold, the query fingerprint is assigned to the corresponding identity class; otherwise, it is rejected under the threshold-based decision rule. The threshold is estimated from distances among gallery or training templates and is not tuned on the test images. This setting reduces the risk of using test information during evaluation.

The hierarchical matching process reduces unnecessary fine comparisons. Stage 1 compares the query coefficient vector with all gallery coefficient vectors and selects a candidate set with the smallest coefficient distances. Stage 2 compares the query energy vector only with the energy vectors in this candidate set. This structure retains the discriminative energy-based comparison while reducing the number of final matching operations. This structure retains the discriminative energy-based comparison while reducing the number of final matching operations, as illustrated in Figure 17.

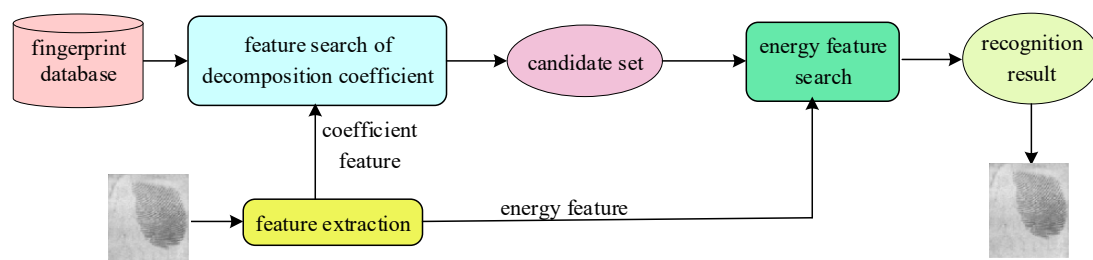


Figure 17. Flow chart of hierarchical recognition.

5.1.3. Visualization of Two-Stage Matching

To illustrate the two-stage hierarchical matching process, Figure 18 provides a visual comparison between the initial coefficient-based coarse matching (Stage 1) and the energy-based fine matching (Stage 2).

Stage 1 employs the features of image decomposition coefficients to rapidly search the database and generate a candidate set of fingerprints with similar coefficient characteristics. Due to the relatively low computational complexity of coefficient feature comparison, this stage significantly improves the overall recognition speed. As shown in

Figure 18a, this initial matching produces numerous candidate correspondences based on coefficient similarity.

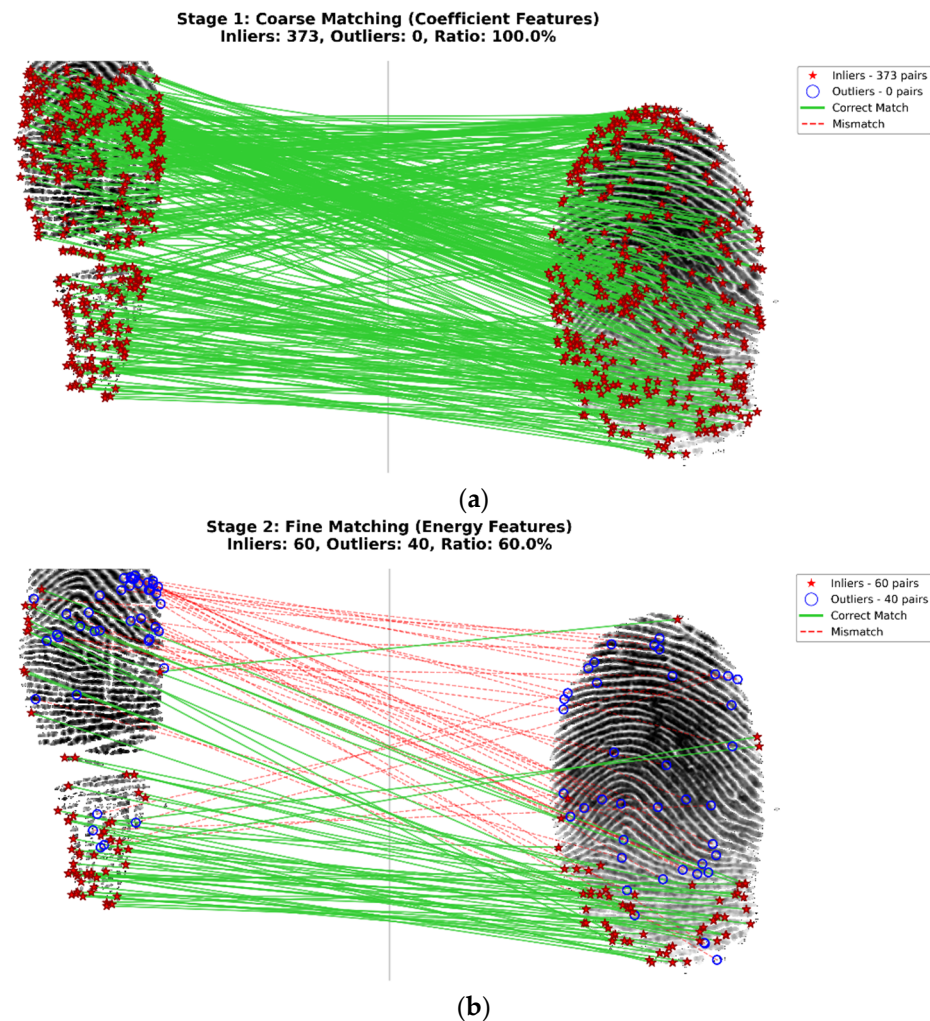


Figure 18. (a). Coefficient-feature-based coarse candidate screening. (b). Energy-feature-based fine matching and rejection of mismatched pairs.

Stage 2 applies the energy features derived from image decomposition to perform fine matching within the candidate set. Since the number of samples in the candidate set is substantially smaller than that in the entire database, the overall recognition efficiency is further enhanced. Based on the minimum difference principle, the unknown fingerprint is identified as belonging to the class whose energy feature vector yields the smallest difference. To clearly demonstrate the effectiveness of the fine matching stage in rejecting incorrect candidates, mismatched point pairs from different fingers are introduced as negative examples, marked with blue circles (○) and red dashed lines. Figure 18b shows the final matching result, where correctly matched points are marked with red asterisks (*) connected by green solid lines, and the rejected mismatched points are marked with blue circles (○) connected by red dashed lines.

Additionally, Figure 19 shows the spatial distribution of unmatched feature points, marked with blue circles (○). These points typically occur near ROI boundaries or in regions with significant non-linear deformation, where local energy descriptors are insufficiently distinctive for reliable matching.

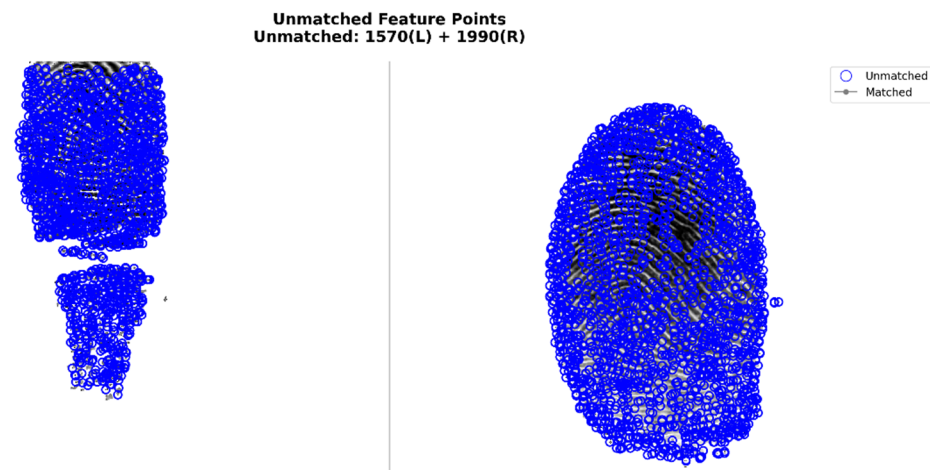


Figure 19. Spatial distribution of unmatched feature points.

The visualization supports the interpretation of the two-stage matching strategy. Stage 1 narrows the search space using coefficient features, whereas Stage 2 performs a more detailed comparison using energy features. This design is intended to improve matching efficiency while preserving closed-set identification accuracy under the evaluated protocol.

5.2. Closed-Set Fingerprint Identification Experiment

The identification experiment used a separate in-house fingerprint dataset containing 150 finger identities, with six images collected for each identity. This dataset was used only for recognition evaluation, whereas FVC2004 was used only for denoising and preprocessing analysis. The recognition dataset was created by an imaging institute within a biometric research center and acquired using an optical fingerprint scanner at 500 dpi, with an image size of 640×640 pixels and 256 gray levels. It contains thumbs, index fingers, and middle fingers from subjects aged 20 to 45 years, with an approximate male-to-female ratio of 3:2. For each finger identity, six images were captured in two sessions separated by at least one week. The same sensor type was used for this experiment; therefore, the experiment evaluates within-sensor closed-set identification rather than cross-sensor recognition.

For the identification experiment, each of the 150 fingers was treated as one identity class. For each identity, one image was randomly selected as the gallery template, and three remaining images were used as probe images, resulting in 150 gallery images and 450 probe images per split. Gallery and probe images from the same finger identity are expected in closed-set fingerprint identification; however, the same image was never used in both sets. To reduce dependence on a single random split, the random selection was repeated five times, and the average results were reported. The same distance score can be used in a verification-oriented setting by comparing it with a validation-derived threshold, but the main experiment in this paper reports closed-set identification because the available dataset is organized as enrolled identities with probe images.

P1. One gallery image and three probe images were selected for each of the 150 finger identities. The selected gallery images formed the template database, and the selected probe images formed the test set for closed-set identification.

P2. The first group of images is preprocessed, and feature extraction is performed using a 3-level given coefficient decomposition. The 150 coefficient characteristic vectors and 150 energy characteristic vectors of the 3-level coefficient decomposition of the image are acquired, respectively, and are archived as training samples.

P3. Probe images are preprocessed and represented using the same three-level decomposition procedure. For each probe image, the coefficient feature vector is first compared with all gallery coefficient vectors to form a candidate set. The energy feature vector of the

probe image is then compared with the energy feature vectors in the candidate set, and the identity is assigned according to the minimum final distance. The random selection and matching procedure is repeated according to the protocol described above.

The matching threshold is determined using gallery/training fingerprints only. Pair-wise distances among gallery or validation templates are computed after preprocessing and feature extraction, and the threshold is estimated from the intra-class distance distribution. Test/probe images are not used to select or tune this threshold. In the two-stage matching process, this threshold controls whether a candidate proceeds from coarse coefficient screening to fine energy-feature matching.

P4. Each probe image is matched against the gallery according to the two-stage procedure. Correct and incorrect identifications are recorded, and the closed-set identification accuracy is calculated.

The proposed hierarchical method was compared with four representative fingerprint recognition approaches: a Siamese Network with a ResNet50 backbone, MRF [15], ANFIS [16], and the eight-neighbor deviation technique [7]. The configuration of the Siamese Network with a ResNet50 backbone is summarized in Table 3. For the five methods reported in Figure 20 and Tables 4–8, the same in-house fingerprint dataset, 150-gallery/450-probe split, identity definition, and closed-set identification metric were used. Method-specific internal parameters were retained only where required by the corresponding algorithm. The average correct identification rates of the proposed method, Siamese Network with ResNet50, MRF, ANFIS, and the eight-neighbor deviation technique were 97.86%, 96.41%, 93.55%, 90.74%, and 86.23%, respectively, as shown in Figure 20.

Table 3. Siamese Network with ResNet50 baseline setting.

Setting	Description
Training/evaluation data	Same in-house fingerprint dataset; 150 gallery images and 450 probe images per split
Backbone	ResNet50 feature extractor in a Siamese two-branch architecture
Input size	224×224 grayscale fingerprint images, resized from the preprocessed ROI
Positive pairs	Image pairs from the same finger identity
Negative pairs	Image pairs from different finger identities; positive/negative pairs were balanced during training
Loss function	Contrastive loss with margin = 1.0
Optimizer	Adam optimizer with an initial learning rate of 1×10^{-4}
Epochs and batch size	400 training epochs; batch size = 32
Augmentation	Small rotation, translation, and contrast perturbation applied to training pairs only
Threshold/tuning	Threshold and model-selection settings were determined from training/validation data only; probe images were used only for final testing

Table 4. Protocol consistency for compared recognition methods.

Protocol Item	Setting Used in This Manuscript
Dataset	Same in-house fingerprint recognition dataset for the five compared methods
Identity definition	Each finger is treated as one identity class
Gallery/probe split	Same 150-gallery/450-probe split per repeated evaluation
Probe usage	Probe images used only for final testing, not for threshold tuning
Preprocessing	Same TIM and ROI preprocessing output used before feature extraction, when compatible with the method
Metric	Average closed-set correct identification rate, recall, precision, and F1-score
Method-specific settings	Only algorithm-internal parameters, such as Siamese training settings or MRF/ANFIS model parameters, were method-specific

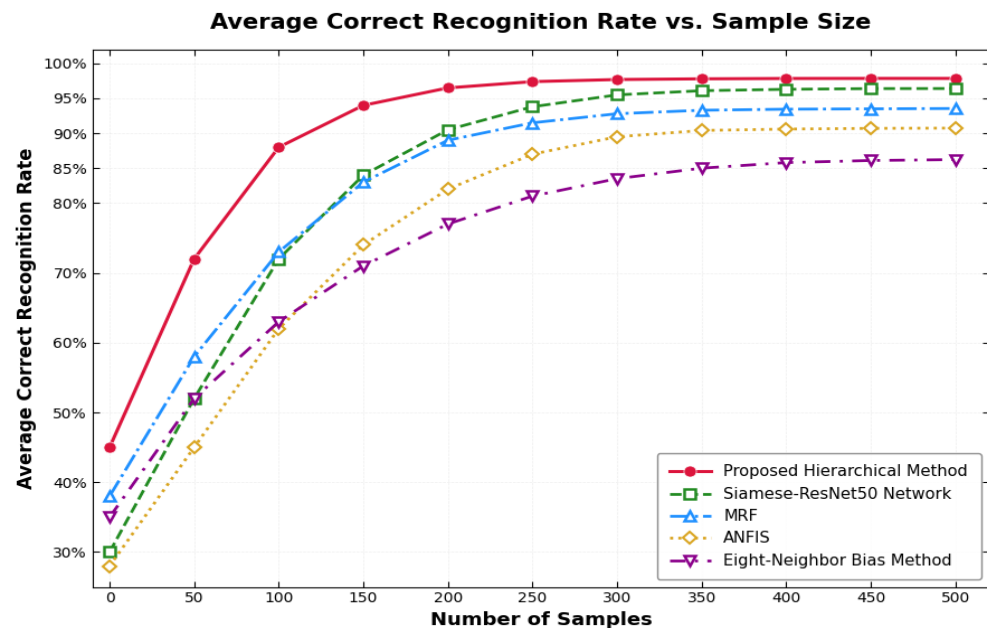


Figure 20. Comparison of the correct recognition rate for the proposed and other methods.

As shown in Figure 20, the proposed method attains the highest average correct identification rate among the compared methods under the reported closed-set protocol. Because the accuracy difference between the proposed method and the Siamese Network with ResNet50 is relatively small, Table 9 further reports Wilson 95% confidence intervals and an independent two-proportion test for the aggregate closed-set identification results. The training and evaluation settings of the Siamese Network with ResNet50 baseline are summarized in Table 3.

To improve the transparency of baseline comparison, the protocol used for the five methods reported in this manuscript is summarized in Table 4.

5.2.1. Comprehensive Comparison of Various Recognition Methods

To evaluate the overall behavior of the compared fingerprint recognition algorithms, this section summarizes closed-set identification accuracy, computational-efficiency trend, noise resistance, environmental adaptability, and parameter-setting requirements. The proposed hierarchical method is compared with four representative methods, including the Siamese Network with ResNet50, MRF, ANFIS, and the eight-neighbor bias method. The comparison protocol is summarized in Table 4, and the comparison results are summarized in Table 5.

Table 5. Comprehensive comparison of various fingerprint recognition methods.

Algorithm	Average Correct Identification Rate (%)	Recognition-Efficiency Trend	Noise/Perturbation Robustness	Applicable Condition
Proposed hierarchical method	97.86%	The increase continues as the sample size increases, and then tends to stabilize after reaching a certain level.	Strong under the reported protocol	Closed-set fingerprint identification
Siamese Network with ResNet50	96.41%	Gradually increase with the increase in sample size	Strong under the reported protocol	General fingerprint image comparison
MRF method	93.55%	Relatively slow increase and early stabilization	medium	Clear fingerprint structure
ANFIS method	90.74%	Dependent on sample size, growth is non-linear	Weak	Small-sample fingerprint comparison
Eight-neighbor bias system method	86.23%	Slow growth, early saturation	Weak	High-resolution fingerprint

The average correct identification rate in Table 5 is calculated from the reported recognition experiments under different sample sizes. The recognition-efficiency trend is observed using the average-correct-identification-rate curve. The proposed method shows a faster increase and a higher final stable value in the reported setting. Its noise resistance and environmental adaptability are interpreted from the preprocessing and feature-representation behavior observed in the experiments; broader robustness claims would require additional cross-sensor and cross-database tests.

5.2.2. Component-Level and Metric Comparison

To further describe recognition behavior, four metrics—accuracy, precision, recall, and F1-score—were used for quantitative comparison among the evaluated methods. All methods were assessed on the same fingerprint dataset under the reported experimental setting. The performance curves are illustrated in Figures 21–24, and the final results are summarized in Table 6.

Table 6. Final metric comparison of the evaluated methods.

Method	R (%)	P (%)	F1 (%)	Accuracy (%)
Eight-Neighbor Bias	84.31	88.15	86.17	86.23
ANFIS	88.44	92.95	90.65	90.74
MRF	93.9	93.89	93.53	93.55
Siamese Network + ResNet50	96.83	94.15	95.45	96.41
Proposed Hierarchical Method	97.95	97.68	97.81	97.86

Figure 21 shows the performance curves of the five evaluated methods over repeated evaluation iterations. A common trend is observed: the reported accuracy increases and then stabilizes as the number of evaluated samples or iterations grows. The proposed method reaches a stable accuracy of 97.86%, followed by the Siamese Network with ResNet50 at 96.41%, MRF at 93.55%, ANFIS at 90.74%, and the eight-neighbor bias method at 86.23%. These results describe comparative behavior under the reported fingerprint recognition protocol.

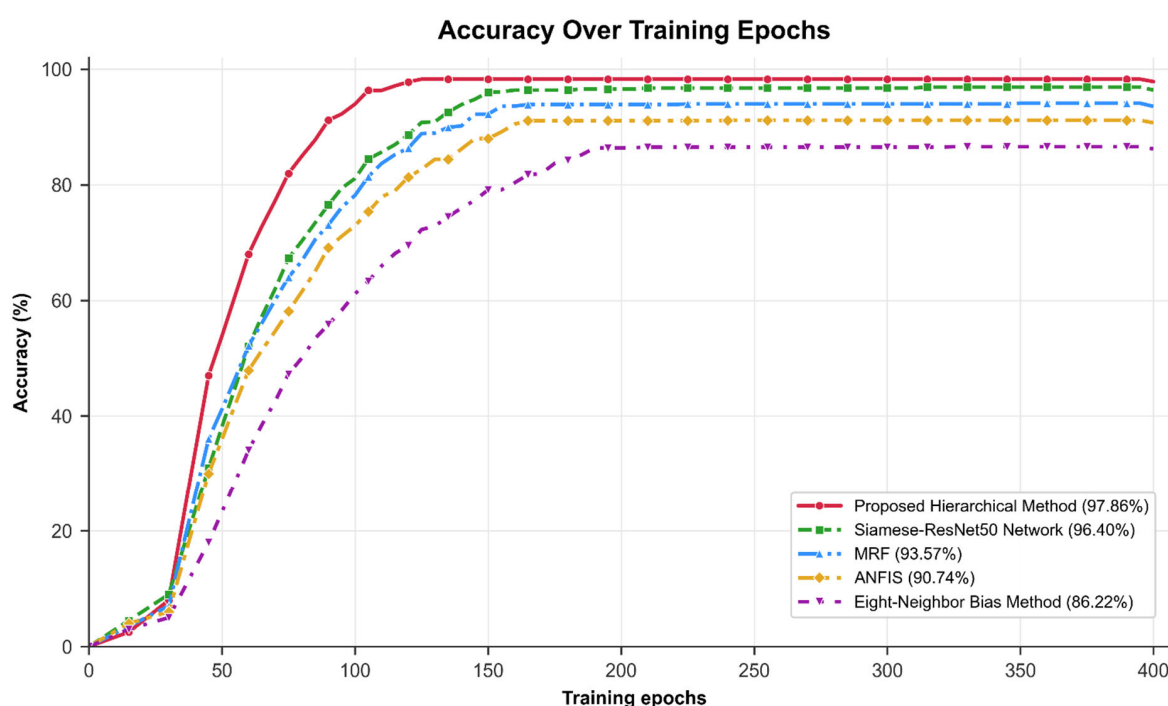


Figure 21. Comparison of overall accuracy rates of several methods.

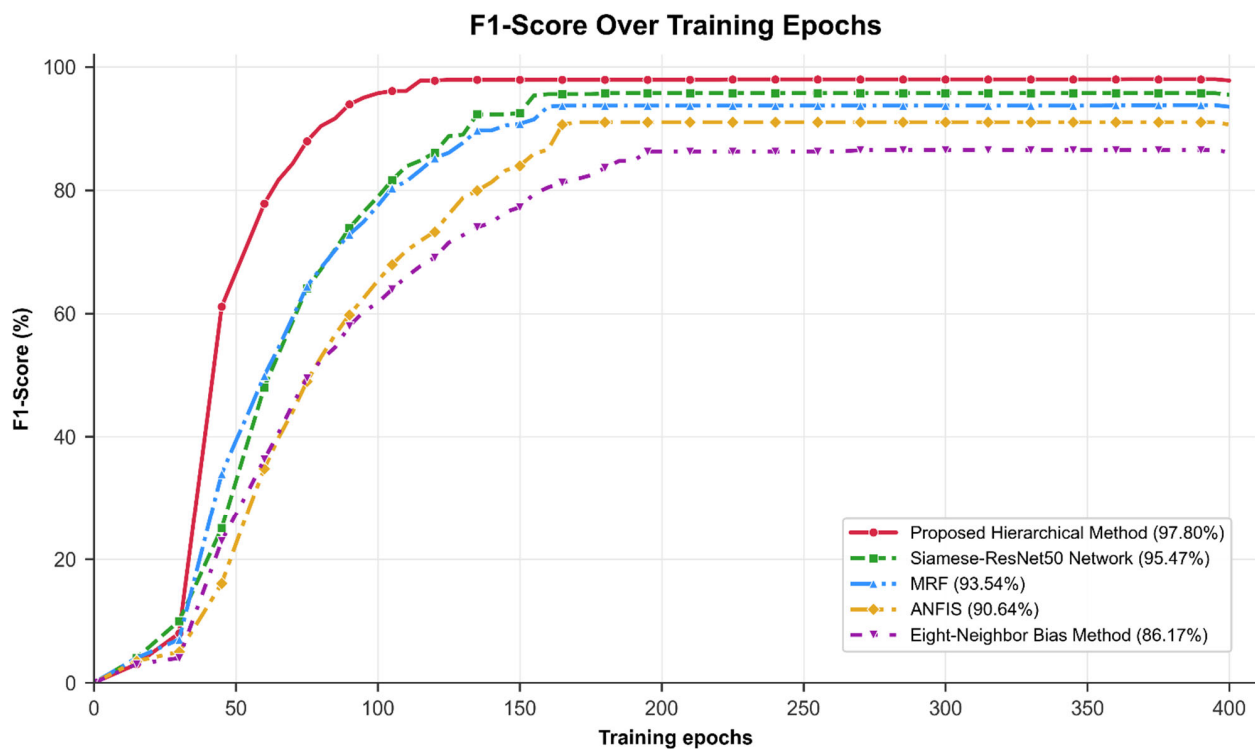


Figure 22. Comparison results of the F1-score of different methods.

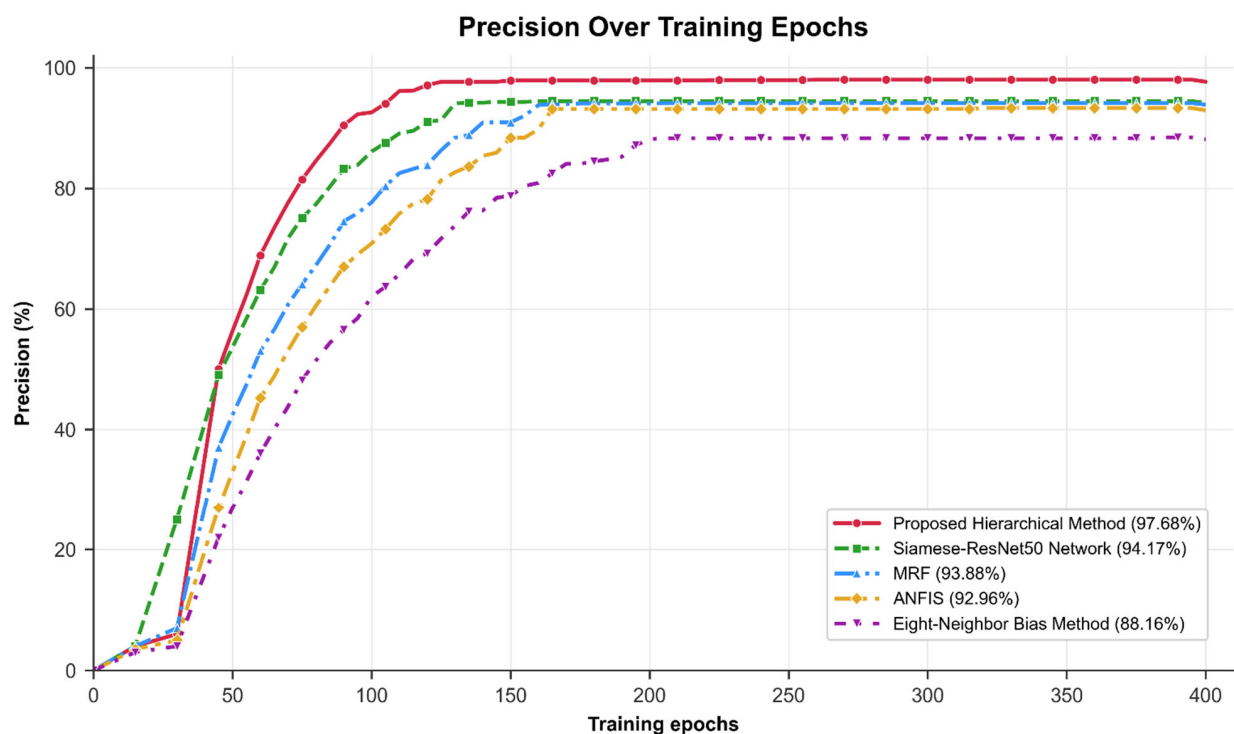


Figure 23. Comparison results of the precision of different methods.

Figures 22–24 summarize recall, precision, and F1-score for the compared methods. The proposed method obtains the highest reported values in this experiment, with a recall of 97.95%, a precision of 97.68%, and an F1-score of 97.81%. These results are consistent with the closed-set identification results and suggest that the hierarchical representation contributes to stable recognition performance under the evaluated protocol.

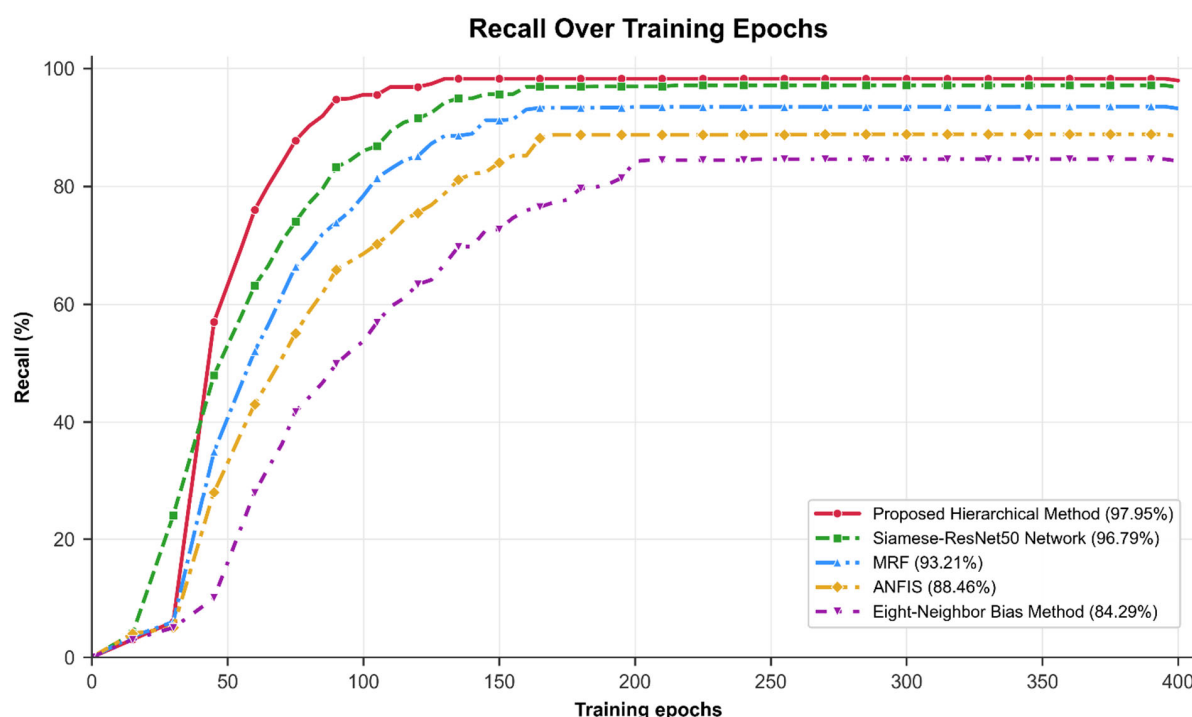


Figure 24. Comparison results of the recall of different methods.

To further examine the contribution of the main processing stages, a component-level comparison was conducted for the proposed method. The results are presented in Table 7.

Table 7. Component-level contribution analysis of the proposed method.

Component Setting	TIM Preprocessing	Coefficient Screening	Energy Fine Matching	Interpretation
Complete pipeline	Included	Included	Included	Reference configuration used for the main recognition result
Without TIM preprocessing	Removed	Included	Included	Tests sensitivity to preprocessing and ridge-noise suppression
Coefficient-only matching	Included	Included	Removed	Tests the contribution of coarse candidate screening without energy refinement
Energy-only matching	Included	Removed	Included	Tests the contribution of fine energy matching without coefficient preselection

Table 7 is used to clarify the role of the main processing stages rather than to introduce an additional set of incompatible performance numbers. The complete pipeline combines TIM preprocessing, coefficient-based candidate screening, and energy-feature fine matching. The component settings show how each stage contributes to either input stabilization, search-space reduction, or final matching discrimination.

Table 8 presents matching accuracy on the gallery/training and probe/test sets.

Table 8. Matching accuracy on the gallery/training and probe/test sets.

Method	R (%)	P (%)
Eight-Neighbor Bias	86.85	85.47
ANFIS	91.26	90.12
MRF	93.91	93.18
Siamese Network + ResNet50	96.75	96.08
Proposed Hierarchical Method	98.12	97.63

The proposed method achieves the highest reported accuracy on both the gallery/training and probe/test sets, with a small difference between them. This result suggests that the complete two-stage architecture provides more stable recognition behavior than the tested component variants under the reported protocol.

5.2.3. Implementation Details and Statistical Interpretation

All denoising and recognition algorithms were executed under the same software environment and data split. The experiments were run on a workstation equipped with an 11th Gen Intel(R) Core(TM) i5-11300H CPU at 3.10 GHz with four physical cores and eight logical processors. All methods were evaluated on the same CPU-based workstation without GPU acceleration. The reported runtime values therefore reflect the computational cost of the evaluated methods under the stated implementation environment, rather than deployment performance on embedded hardware. The aggregate-level statistical comparison of closed-set identification accuracy is summarized in Table 9.

Table 9. Aggregate-level statistical comparison of closed-set identification accuracy.

Method	Accuracy	Wilson 95% CI	<i>p</i> -Value vs. Proposed Method
Proposed hierarchical method	97.86%	96.06–98.85%	Reference
Siamese Network with ResNet50	96.41%	94.26–97.77%	0.192
MRF	93.55%	90.89–95.47%	0.001
ANFIS	90.74%	87.71–93.08%	<0.001
Eight-neighbor bias method	86.23%	82.74–89.11%	<0.001

To address the statistical interpretation of the accuracy differences, an aggregate-level comparison was performed using the 450 probe images in each split as the evaluation unit. Wilson 95% confidence intervals were calculated for each reported closed-set identification accuracy. In addition, an independent two-proportion test was used as an approximate aggregate-level comparison between the proposed method and each baseline. This test is less informative than a paired test based on image-level predictions, but it provides a transparent statistical check using the available aggregate results.

5.2.4. Scope and Limitations of the Evaluation

The current evaluation uses 150 enrolled finger identities and 450 probe images per split, with five repeated gallery/probe selections. This scale is appropriate for assessing the proposed lightweight pipeline under controlled closed-set fingerprint recognition conditions. The aggregate-level confidence intervals in Table 9 further quantify the uncertainty of the reported accuracy values. Larger databases, cross-sensor matching, open-set unknown-user rejection, and full verification metrics remain important directions for future evaluation.

6. Conclusions

This paper presents a lightweight fingerprint recognition pipeline for closed-set identity recognition. The workflow integrates TIM-based preprocessing, ROI extraction, minutiae analysis, hierarchical coefficient and energy feature representation, and two-stage template matching. The experimental results indicate that the proposed method achieves competitive closed-set identification accuracy on the evaluated fingerprint dataset while retaining an interpretable processing structure. These findings support the feasibility of using hierarchical energy-feature decomposition for small-data fingerprint recognition scenarios.

Future work will extend the evaluation to larger cross-sensor fingerprint databases, introduce open-set unknown-user rejection and verification-oriented metrics such as FAR, FRR, EER, ROC/DET, and TAR@FAR, and examine robustness under controlled rotation, translation, partial fingerprint, pressure distortion, dry/wet finger conditions, and sensor variation.

Author Contributions: Y.S. and M.L. conceived and designed the study framework, performed the feature analysis, and drafted the manuscript. Q.W. developed the noise reduction algorithm and conducted the simulation experiments. S.W. carried out the feature preprocessing, performed the feature extraction analysis, and participated in the design of the hierarchical recognition experiments. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the Key Science and Technology Project of Henan Province University (26B413002) and Henan Provincial Key Laboratory of Livestock and Poultry Genetic Improvement and Healthy Breeding (C81JX260004), respectively.

Data Availability Statement: The datasets used and/or analyzed during the current study are available from the corresponding author, Qing Wu (qewu@sspu.edu.cn), on reasonable request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Caldwell, T. Tabletop combines image display and fingerprint recognition. *Biom. Technol. Today* **2013**, *2013*, 12–13. [\[CrossRef\]](#)
2. Cimpoi, M.; Maji, S.; Kokkinos, I.; Vedaldi, A. Deep Filter Banks for Texture Recognition, Description, and Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3828–3836. [\[CrossRef\]](#)
3. Cao, K.; Jain, A.K. Automated Latent Fingerprint Recognition. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*; IEEE: New York, NY, USA, 2019; Volume 41, pp. 788–800. [\[CrossRef\]](#) [\[PubMed\]](#)
4. Zhang, J.; Jing, X.J.; Chen, N.; Wang, J.L. Incomplete fingerprint recognition based on feature fusion and pattern entropy. *J. China Univ. Posts Telecommun.* **2013**, *20*, 121–128. [\[CrossRef\]](#)
5. Jiang, X.; You, X.; Yuan, Y.; Gong, M. A method using long digital straight segments for fingerprint recognition. *Neurocomputing* **2012**, *77*, 28–35. [\[CrossRef\]](#)
6. Kızrak, A.M.; Özen, F. A new median filter based fingerprint recognition algorithm. *Procedia Comput. Sci.* **2011**, *3*, 859–865. [\[CrossRef\]](#)
7. Nanni, L.; Lumini, A. A novel method for fingerprint verification that approaches the problem as a two-class pattern recognition problem. *Neurocomputing* **2006**, *69*, 846–849. [\[CrossRef\]](#)
8. Moros, J.; Serrano, J.; Gallego, F.; Macías, J.; Laserna, J. Recognition of explosives fingerprints on objects for courier services using machine learning methods and laser-induced breakdown spectroscopy. *Talanta* **2013**, *110*, 108–117. [\[CrossRef\]](#) [\[PubMed\]](#)
9. Lin, C.H.; Chen, J.L.; Tseng, C.Y. Optical sensor measurement and biometric-based fractal pattern classifier for fingerprint recognition. *Expert Syst. Appl.* **2011**, *38*, 5081–5089. [\[CrossRef\]](#)
10. Wang, L.; Wang, X.; Kong, L. Automatic authentication and distinction of *Epimedium koreanum* and *Epimedium wushanense* with HPLC fingerprint analysis assisted by pattern recognition techniques. *Biochem. Syst. Ecol.* **2012**, *40*, 138–145. [\[CrossRef\]](#)
11. Xie, X.; Wu, J.; Jing, M. Fast two-stage segmentation via non-local active contours in multiscale texture feature space. *Pattern Recognit. Lett.* **2013**, *34*, 1230–1239. [\[CrossRef\]](#)
12. Lo, E.H.S.; Pickering, M.R.; Frater, M.R.; Arnold, J.F. Image segmentation from scale and rotation invariant texture features from the double dyadic dual-tree complex wavelet transform. *Image Vis. Comput.* **2011**, *29*, 15–28. [\[CrossRef\]](#)
13. Zhou, H.; Zheng, J.; Wei, L. Texture aware image segmentation using graph cuts and active contours. *Pattern Recognit.* **2013**, *46*, 1719–1733. [\[CrossRef\]](#)
14. Yu, J. Texture segmentation based on FCM algorithm combined with GLCM and space information. In *2011 International Conference on Electric Information and Control Engineering*; IEEE: New York, NY, USA, 2011; pp. 4569–4572.
15. Long, Z.; Younan, N.H. Multiscale texture segmentation via a contourlet contextual hidden Markov model. *Digit. Signal Process.* **2013**, *23*, 859–869. [\[CrossRef\]](#)
16. Inyang, U.G.; Akinyokun, O.C. A hybrid knowledge discovery system for oil spillage risks pattern classification. *J. Artif. Intell. Res.* **2014**, *3*, 77–86. [\[CrossRef\]](#)
17. Wu, Q.E.; Wang, J.; Yang, C.; Cui, G.; Yang, W. Target recognition by texture segmentation algorithm. *Expert Syst. Appl.* **2016**, *46*, 394–404. [\[CrossRef\]](#)

18. Wei, X.; Dong, Y.; Liu, Q.; Wang, L.; Lou, L. Robust corner detection in continuous space. *Vis. Comput.* **2025**, *41*, 759–772.
19. Liu, M.; Chen, X.; Wang, X. Latent fingerprint enhancement via multi-scale patch based sparse representation. *IEEE Trans. Inf. Forensics Secur.* **2014**, *10*, 6–15. [[CrossRef](#)]
20. Scabini, L.; Sacilotti, A.; Zielinski, K.M.; Ribas, L.C.; De Baets, B.; Bruno, O.M. A comparative survey of vision transformers for feature extraction in texture analysis. *J. Imaging* **2025**, *11*, 304. [[CrossRef](#)] [[PubMed](#)]
21. Wu, J.; Wu, H.; Yuan, G. Detail-aware image denoising via structure preserved network and residual diffusion model. *Vis. Comput.* **2025**, *41*, 639–658.
22. Ghalati, M.K.; Nunes, A.; Ferreira, H.; Serranho, P.; Bernardes, R. Texture analysis and its applications in biomedical imaging: A survey. *IEEE Rev. Biomed. Eng.* **2021**, *15*, 222–246. [[CrossRef](#)]
23. Zhang, K.; Zuo, W.; Chen, Y.; Meng, D.; Zhang, L. Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising. *IEEE Trans. Image Process.* **2017**, *26*, 3142–3155. [[CrossRef](#)] [[PubMed](#)]
24. Zhang, K.; Zuo, W.; Zhang, L. FFDNet: Toward a Fast and Flexible Solution for CNN-Based Image Denoising. *IEEE Trans. Image Process.* **2018**, *27*, 4608–4622. [[CrossRef](#)]
25. Liang, J.; Cao, J.; Sun, G.; Zhang, K.; Van Gool, L.; Timofte, R. SwinIR: Image Restoration Using Swin Transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*; IEEE: New York, NY, USA, 2021; pp. 1833–1844.
26. Zamir, S.W.; Arora, A.; Khan, S.; Hayat, M.; Khan, F.S.; Yang, M.-H. Restormer: Efficient Transformer for High-Resolution Image Restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: New York, NY, USA, 2022; pp. 5728–5739.
27. Artabaz, S.; Sliman, L. Feature fusion and selection using handcrafted vs. deep learning methods for multimodal hand biometric recognition. *Sci. Rep.* **2025**, *15*, 29237. [[CrossRef](#)] [[PubMed](#)]
28. Huang, Y.H.; Cao, Y.P.; Lai, Y.K.; Shan, Y.; Gao, L. NeRF-Texture: Synthesizing Neural Radiance Field Textures. *IEEE Trans. Pattern Anal. Mach. Intell.* **2024**, *46*, 5986–6000. [[CrossRef](#)] [[PubMed](#)]
29. Huang, C.Y.; Zhang, Y.L.; Liao, S.Y.; Huang, W.C.; Chen, J.H.; Dong, B.C.; Hsu, C.J.; Huang, C.Y. Real-Time Cell Gap Estimation in LC-Filled Devices Using Lightweight Neural Networks for Edge Deployment. *Nanomaterials* **2025**, *15*, 1289. [[CrossRef](#)] [[PubMed](#)]
30. Cho, N.S.; Kim, C.S.; Park, C.; Park, K.R. GAN-based blur restoration for finger wrinkle biometrics system. *IEEE Access* **2020**, *8*, 49857–49872. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.