International Journal of Computer Sciences and Engineering
Vol.**13**, Issue.**8**, pp.**58-67**, August **2025**
ISSN: 2347-2693 (Online)
Available online at: www.ijcseonline.org

Review Article

A Comprehensive Review of Fingerprint Recognition Systems, Methods, and Algorithms

Himanshi Sharma[1], Hardeep Singh[2*]

[1,2]Dept. of Computer Engineering and Technology, Guru Nanak Dev University, Amritsar, India

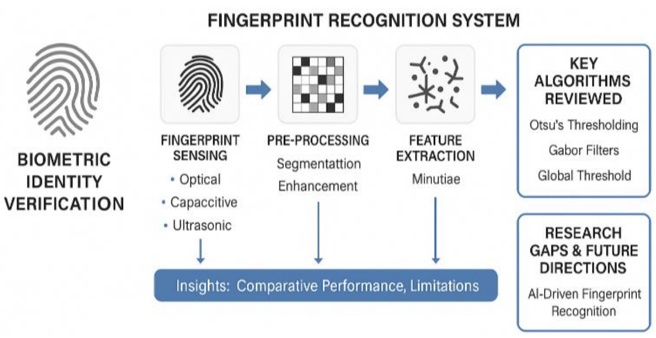
*Corresponding Author:

Received: 23/Jun/2025; **Accepted**: 25/Jul/2025; **Published**: 31/Aug/2025. **DOI**: https://doi.org/10.26438/ijcse/v13i8.5867

Abstract: Today, we need security in every system, be it our personal laptop, computer system, ATM system, etc. We need to ensure that our data is well protected from misuse by imposters by verifying the identity of a user. Traditional systems of verification using credentials are not foolproof and face many challenges. These systems cannot meet the growing security demands in applications such as border crossing. Therefore, biometric recognition is becoming popular day by day. Biometrics is a way of verifying a user's identity using innate traits that define the individual. This work gives a review of biometrics, particularly fingerprint recognition systems. The study also assesses popular methods, including the Bozorth3 algorithm, Otsu's Thresholding, Gabor Filters, and Global Threshold Segmentation. There is a discussion of comparative insights into their limitations and performance. In addition to providing a comprehensive literature review, this paper identifies current research gaps and suggests potential directions for future advancements in fingerprint recognition technology.

Keywords: Biometric, Bozorth3, Otsu's Thresholding, Gabor Filters, Feature Extraction, Segmentation

Graphical Abstract-The four stages of the fingerprint recognition process are summarized in Figure sensing, pre-processing, feature extraction, and matching. It highlights key algorithms such as Otsu's Thresholding, Gabor Filters, Global Thresholding, and the Bozorth3 algorithm. The workflow emphasizes their role in enhancing accuracy and reliability. The graphic also includes insights into limitations and proposes future directions for advancing fingerprint recognition technology.



1. Introduction

Conventional credential-based verification systems are problematic because of problems like reused or weak passwords and ID card loss or forgetfulness. As a result, biometrics have become more popular, allowing users to avoid carrying ID cards or remembering passwords [1]. A popular biometric authentication method is fingerprint recognition, which uses distinctive physical characteristics such as voice, iris patterns, fingerprints, and face features to identify users [18].

1.1 Process of Fingerprint Recognition

A fingerprint system has two phases: modeling and recognition. The modeling phase includes capturing the fingerprint, preprocessing, feature extraction, template creation, and storage [2]. The recognition phase is divided into identification (1: N matching) and verification (1:1 matching). Identification compares a new fingerprint against a database to determine a match, while verification compares it with a specific stored template, returning a similarity score [3]. The different processes of fingerprint recognition are displayed in Figure 1.

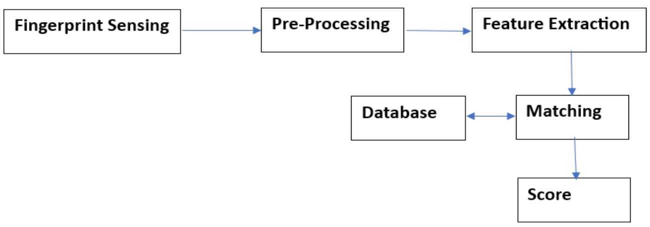


Figure 1. Processes involved in fingerprint recognition

Fingerprint Sensing. It can be done online using fingerprint sensors or offline by inking the fingertip and scanning it and is the initial stage in verification.

Pre-processing. It uses segmentation, enhancement, and ridge analysis to improve fingerprint images by eliminating noise and fine-tuning ridge patterns. Because of its fragility, greyscale depiction is frequently avoided. The detailed pre-processing is shown in Figure 2.

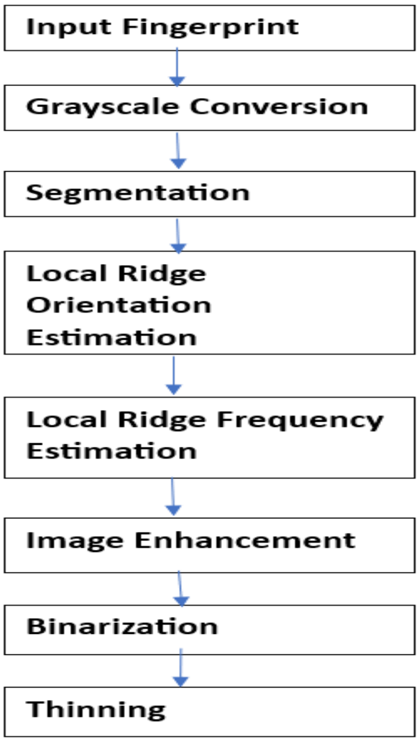


Figure 2. Detailed Pre-processing

Local ridge orientation. It is described as the inclination of the ridge with the horizontal plane. The angle ridge makes with the horizontal plane is important when using the Gabor filter to enhance images. The three primary approaches are learning-based, mathematical model-based, and gradient-based.

Local ridge frequency. The number of ridges per unit length perpendicular to ridge orientation [4]. Common estimation methods include (a) Block-Based Method and (b) Fourier Transform

Image Enhancement. It is one of the most critical processes as the quality of the image has an important impact on the success rate of the systems. The purpose of image enhancement is to enhance structural integrity while preserving local details [5].

Segmentation. This step isolates the fingerprint area from the background to prevent extracting features from noise. Some techniques used are Thresholding, Watershed Segmentation, Edge detection, etc [37].

Binarization. An improved gray-level image is transformed into a binary image for feature extraction using this method [30]. It usually uses thresholding algorithms to differentiate between fingerprint troughs (background) and ridges (foreground). Valleys are denoted by 0 (white) and ridges by 1 (black).

Thinning. By eliminating unnecessary pixels, this technique thins the ridges to a singular pixel, simplifying the identification of minute details. It facilitates memory efficiency, pattern matching, and feature extraction. Guo-Hall and Zhang-Suen are examples of common algorithms for thinning [11].

Feature Extraction. In order to generate templates, this process finds distinctive fingerprint characteristics. The most popular technique, the minutiae extraction algorithm, finds ridge ends and bifurcations [13].

Postprocessing. After feature extraction, postprocessing is required to remove erroneous minutiae points like spur, hole, spike structure, etc. For instance, a ridge ending that is within a threshold distance of a bifurcation is deleted. [7].

Matching. In this phase, the previously retrieved features from the inputted fingerprint are compared with database templates. Here, we ascertain how similar the input fingerprint and the template are. Minutiae-based matching, correlation-based matching, and pattern-based matching are some of the several matching techniques.

1.2 Objective of the study

In this research study, the main objectives are as follows:

- To review the basic operations and stages of fingerprint recognition systems.
- To study pre-processing techniques such as segmentation, orientation and frequency estimation, image enhancement, and thinning.
- To review feature extraction and matching algorithms like Crossing Number and Bozorth3.
- To highlight the advantages, limitations, and performance aspects of these methods.

1.3 Organization

The remainder of the paper is structured as follows: Section 2 contains the related work, Section 3 contains preliminaries, Section 4 contains the results and discussion, Section 5 contains the research gaps, and Section 6 summarizes the research work with future scope.

2. Related Work

The ability to recognize fingerprints has greatly advanced. Table 1 gives an overview of the major advancements in fingerprint recognition systems and related publications.

Table 1. Literature Review

Sr. No.	Author	Year	Technique	Description	Advantages	Limitations
1.	Gnanasivam and Muttan [14]	2010	An effective algorithm for preprocessing fingerprints and extracting features.	Algorithm includes two stages: Getting the vertically orientated fingerprint image and detecting fingerprint's core point.	94% of rotations and 92% of core detections are accurate.	Ridge smoothness is decreased by the algorithm's vertical orientation procedure.
2.	Pillai et al. [26]	2011	Fast Fourier Transform (FTT), Fourier Mellin Transform.	The study introduces two image recognition techniques: parallel neural networks and image subtraction with thresholding.	For the image subtraction method accuracy of comparison of high-quality fingerprints is 97%.	Accuracy dropped to 93% with poor-quality fingerprints.
3.	Carsten Gottschlich [15]	2012	Curved Gabor filters.	This paper presents an approach to image improvement via curved Gabor filters and ridge frequency estimation utilizing curved areas.	Greater processing speed and increased matching accuracy.	Error in orientation field estimation can affect the performance.
4.	Josphineleela and Ramakrishnan [35]	2012	SVM-Kernel classification with orientation field extraction.	This paper proposes a method to detect altered fingerprints using SVM-Kernel along with orientation field extraction.	Greater accuracy than current techniques, useful for huge datasets of clustered fingerprints.	High processing complexity and poor performance for images that are badly damaged.
5.	Babatunde et al. [6]	2013	A uniformity-level approach for estimating fingerprint ridge frequency.	The enhanced algorithm using a block-processing approach is applied to both real and synthetic fingerprint images.	The enhanced algorithm exhibits greater speed and efficiency.	Accuracy decreases as the noise level increases, irrespective of size.
6.	Quadri and Choudhary [28]	2014	Fast fingerprint enhancement algorithm.	This paper describes a quick fingerprint enhancement technique that improves clarity using local ridge orientation and ridge frequency.	Improved goodness index and verification performance.	Block processing can be replaced with pixel processing to expedite calculation and lower complexity.
7.	Affan Bin Muhammad Sidek [31]	2015	Discrete Wavelet Transform (DWT) and Gray-Level Co-occurrence Matrices (GLCM).	Uses MATLAB to examine fingerprint databases along with use of GCLM and DWT algorithms.	DWT compression maintains image quality and maximizes memory use.	The dissimilarity value of DWT is higher and more sensitive than that of GLCM, and it rises with additional noise.

8.	Pavol Marák-Alexander Hambalík [16]	2016	An artificial neural network as a feature extractor, and BOZORTH3 matching.	Two neural networks — one for level-2 feature extraction and another for coarse categorization— are used in the identification mode.	Achieved an accuracy of 98.64% on a mixed high and low-quality fingerprint dataset.	Expanding the training database can increases accuracy.
9.	Sudiro et al. [32]	2017	Hardware-based artificial neural network device.	This research describes a fingerprint recognition system that incorporates minutiae point extraction algorithm with a hardware-based artificial neural network device.	The false Acceptance Rate (FAR) is 41.57%, the False Rejection Rate (FRR) is 41.13% and the Equal Error Rate is 41.35%.	Over 40% FAR and FRR result in frequent recognition errors.
10.	Sharma and Kaur [36]	2018	Random Decision Forest classification.	This study employs a machine learning approach to extract features, then separates the two fingerprints to automatically partition the overlapped fingerprint regions.	False Rejection Rate (FRR) is 5.9%.	Investigating several other strategies could enhance segmentation outcomes even more. Improvement is still required to completely automate the process and distinguish between more than two photographs that overlap.
11.	Kuchana and Malneni [20]	2020	Harris Corner Detection and Principal Component Analysis.	This work studies minutiae matching using FLANN, Brute Force, and Harris Corner Detector and introduces a technique for region of interest detection utilizing Histogram of Orientated Gradients.	Processing time is decreased by concentrating the algorithm on the area of interest rather than examining the full image.	Principal Component Analysis doesn't give satisfying results.
12.	Appati et al. [5]	2021	Transform -Minutiae Fusion-Based Model.	This work describes a fingerprint identification model based on transform-minutiae fusion. Wavelets are used to extract features, whereas wave atom transforms are used to smooth data.	Recognition accuracy for FVC2002-DB2_B, DB3_B, and DB4_B is 95%, 100%, and 95% respectively.	Prediction accuracy is 50% while testing with only minute features, and it increases when additional features are added.
13.	Mua'ad et al. [23]	2021	K-means, local binary pattern (LBP), Wavelet Packet tree (WPT), and minutiae methods for feature extraction.	Several fingerprint images were taken and treated using the methods mentioned in this paper.	K-means and minutiae methods exhibit robustness against image rotation and provide distinct features for each fingerprint image.	WPT and LBP struggle with image rotation; nevertheless, a histogram guarantees rotation robustness and maintains

						distinguishing characteristics.
14.	Abraham Junior Abbey [1]	2022	Fisherface and Eigenface methods from facial recognition.	A non-minutiae-based algorithm adopted from facial recognition is used to develop a system that is not affected by partial occlusion.	The fisherface and eigenface approaches have respective accuracy rates of 90% and 86.7%.	The major restriction is the geometry of the dataset images used.
15.	Agha et al. [2]	2023	SSO (Single Sign-On).	The usage of SSO-based fingerprint authentication for cloud services for businesses is suggested in this paper.	SSO-based fingerprint authentication reduces security risks in cloud computing.	Including finger pulse rates in biometric security checks makes it harder for unauthorized users to access systems.
16.	Rajaram et al. [29]	2023	Convolutional neural network (Child-CLEF).	Child-CLEF, a CNN-based fingerprint recognition system for kids is presented in this work. While BOZORTH3 manages recognition and Child-CLEF Net extracts features.	The accuracy achieved with the proposed system is 98.46% with a 1.99% error rate in comparison.	Patches should be utilized instead of the whole fingerprint to create networks that are not dependent on the fingerprint size.
17.	Mao et al. [22]	2024	SLA (Stereo-lithography Apparatus) technology, a form of 3D printing.	This study contrasts fingerprint impressions versus SLA 3D printing. It emphasizes SLA printing as a better option than conventional techniques.	The unlocking success rate of the fingerprint imprint method is as low as 24.4% and as high as 35.6%.	Using image processing and optimizing ridge height parameters can result in an optimal unlocking success rate of up to 91.1%.

3. Preliminaries

3.1 Pre-Processing

Segmentation. It helps extract relevant features. It divides the image's pixels into two categories: foreground and backdrop [12]. Thresholding is one of the common techniques for segmentation. 255 and 0 are the two typical grey values that make up the output image after threshold segmentation [24].

Using Threshold Segmentation. Using a threshold, this method divides a greyscale image into foreground and background [8]. Pixels below or equal to the threshold are categorized as background (0), while pixels above threshold are categorized as foreground (1). Simple pictures with a light object on a dark backdrop work well with it.

$$g(c,d) = \begin{cases} 1 & if\ f(c,d) > T \\ 0 & if\ f(c,d) \le T \end{cases} \quad (1)$$

where $g(c,d)$ is the output image, $f(c,d)$ is the input image and T is the threshold intensity value.

Global Threshold Segmentation Algorithm.

1. Input the grayscale image.
2. Select the initial threshold value (T).
3. Divide the pixels into two classes based on the value of their intensities.
   a) Class 1- comprises all pixels with intensity exceeding the threshold value (I > T).
   b) Class 2- includes every pixel having intensity below or equal to the threshold value (I ≤ T).
4. Calculate the average intensity value for the two classes.

$$M1 = \frac{Sum\ of\ intensities\ of\ pixels\ of\ class\ 1}{Number\ of\ pixels\ in\ class\ 1} \quad (2)$$

$$M2 = \frac{Sum\ of\ intensities\ of\ pixels\ of\ class\ 2}{Number\ of\ pixels\ in\ class\ 2} \quad (3)$$

5. Calculate the new threshold value(T).

$$T = \frac{M1+M2}{2} \quad (4)$$

6. Iterate through steps 2-4 until the difference between the T values is smaller than the initial value.

Otsu's Threshold Technique. By splitting the image into two classes and disregarding the spatial correlations between pixels, Otsu's threshold approach improves global threshold segmentation [10].

Otsu's Threshold Algorithm.

1. Calculate the input image's normalized histogram.
2. Select the initial threshold value (T).
3. Divide the pixels into background (0) and foreground (1) based on the threshold value (T)
   - Background- every pixel having intensity below or equal to the threshold.
   - Foreground- every pixel having intensity more than the threshold.
4. Compute weights w_b and w_f

$$w_b = \frac{No.\ of\ pixels\ in\ the\ background}{Total\ no.of\ pixels} \quad (5)$$

$$w_f = \frac{No.\ of\ pixels\ in\ foreground}{Total\ no.of\ pixels} \quad (6)$$

5. Compute the mean intensity of the background (μ_b) and foreground (μ_f)

$$\mu_b : \frac{Sum\ of\ intensities\ of\ pixels\ in\ background}{Total\ no.\ of\ pixels\ in\ background} \quad (7)$$

Ridge Orientation Estimation. A gradient-based method is the most straightforward way to estimate orientation. Using this method, gradient vectors were computed by calculating partial derivatives of the grey intensities at every pixel.

$$(x, y) = \begin{pmatrix} G_x(x,y) \\ G_y(x,y) \end{pmatrix} = \begin{pmatrix} \frac{\partial I(x,y)}{\partial x} \\ \frac{\partial I(x,y)}{\partial x} \end{pmatrix} \quad (8)$$

where $G_x(x,y)$ and $G_y(x,y)$ are derivatives of I at pixel (x,y) with respect to the x-axis and y-axis and I represents a grayscale fingerprint image [9].

The local ridge orientation at pixel (x,y) is represented by the angle $\theta(x,y)$. Since the ridge line is undirected it does not have any start or end point value of $\theta(x,y)$lies in the interval [0, π). Also, observe that the ridge orientation and the gradient orientation is perpendicular.

Ridge Orientation Algorithm.

1. Segment the image into blocks that overlap by w*w.
2. Compute the horizontal and vertical gradients $\partial_x(x,y)$ and $\partial_y(x,y)$ at every pixel of the image $I(x,y)$ using Sobel operators.
3. If the gradient values are the same i.e. $\partial_x(x,y) = \partial_y(x,y)$, then add random noise ± 1 to one of the gradient values. If one of the gradient values is equal to 0, then add random noise ± 1. Random noise is added to improve robustness and prevent numerical issues.
4. Compute the average gradient vectors to estimate the dominant ridge orientation in each block [25].

$$V_l(l,m) = \sum_{c=l-\frac{n}{2}}^{l+\frac{n}{2}} \sum_{d=m-\frac{n}{2}}^{m+\frac{n}{2}} 2\partial_l(c,d)\partial_m(c,d) \quad (9)$$

$$V_m(l,m) = \sum_{c=l-\frac{n}{2}}^{l+\frac{n}{2}} \sum_{d=m-\frac{n}{2}}^{m+\frac{n}{2}} (\partial_l^2(c,d) - \partial_m^2(c,d)) \quad (10)$$

5. Compute $\Phi(l,m)$ i.e. gradient orientation. (2013) [28]

$$\Phi(l,m) = \frac{1}{2}tan^{-1}\left(\frac{V_l(l,m)}{V_m(l,m)}\right) \quad (11)$$

6. Calculate the final local ridge orientation. (2013) [28]

$$\theta_{gr}(l,m) = \Phi(l,m) + k\pi \quad (12)$$

$$\text{Where } k = \begin{cases} \frac{1}{2} & (\Phi(l,m) < 0) \vee (\Phi(l,m) \geq 0 \wedge V_l(l,m) > 0) \\ 1 \ \ for & \Phi(l,m) < 0 \wedge V_l(l,m) \geq 0 \\ 0 & \Phi(l,m) \geq 0 \wedge V_l(l,m) \leq 0 \end{cases}$$

Ridge Frequency Estimation. It is the number of ridges per unit length along a hypothetical section perpendicular to the ridge orientation. The most widely employed approach for estimating ridge frequency is Fourier-based [6].

Ridge Frequency Algorithm.

1. Define a small window around the region of interest containing ridge patterns using algorithms like a hamming window, gaussian window, etc.
2. Convert the spatial domain representation to the frequency domain by applying the 2D Fourier transform on this windowed region.
3. Calculate the power spectrum to represent the energy distribution across various frequencies in a fingerprint image using Fourier coefficients.

$$Power\ Spectrum = (Magnitude\ of\ Fourier\ coefficients)^2 \quad (13)$$

4. Identify and extract the dominant ridge frequency corresponding to the spacing between adjacent ridge lines in the fingerprint image.

Image Enhancement. This determines whether or not correct details are extracted. It aims to improve image quality and minimize noise so that it doesn't interfere with minutiae extraction. Contextual filters are often used for enhancement, especially Gabor filters.

Gabor Filters. These are commonly employed due to their frequency-selective and orientation-selective features [15]. Users can select frequency and orientation as per their requirements. Because of these properties.

Gabor Filters Algorithm.

Input a fingerprint image $I(x,y)$.

Now, establish three parameters for employing the Gabor filter on the fingerprint image:

- ω - filter orientation
- f - frequency of sinusoidal plane wave
- σ_x, σ_y- standard deviations of Gaussian kernel

Generate a 2D Gabor filter kernel $g(x,y : \omega, f)$ based on specified parameters.

Choosing σ_x and σ_y involves a trade-off: higher values improve noise resistance but may introduce spurious ridges and valleys, while smaller values fail to remove noise effectively. Typically, both are set to 4.0 [25].

Convolve the fingerprint image with the Gabor filter kernel.

The enhanced image E is obtained as [19].

$$E(k,l) = \sum_{u=-\omega_g/2}^{\omega_g/2} \sum_{v=-\omega_g/2}^{\omega_g/2} G(u, v{:}\, \omega(k,l), f(k,l)).I(k-u, l-v) \quad (14)$$

Binarization. Using the threshold technique, binarization turns a colored or greyscale image into a binary one. Local thresholds adapt to specific areas of the image, whereas global thresholds apply a single value. Background pixels are those that are below the threshold, while foreground pixels are those that are above it (1).

Thinning. It reduces the number of pixels by removing redundant pixels. One of the most common algorithms used is Zhang-Suen's Algorithm.

Zhang-Suen's Algorithm. It is a type of iterative algorithm i.e. it continuously eliminates pixels near the border until the thinning image is reduced to just one pixel. It is also called the 2-pass algorithm because for every iteration it performs two checks to remove pixels. Except for the skeleton's contour points, this algorithm eliminates every contour point in the picture [17]. This algorithm works for a 3x3-sized block as shown in Figure 3.

The thinning algorithm is usually performed on binarized images. A binary image can be defined as a matrix where the binary value of each pixel is 1 if it is black and 0 if it is white. For example: - consider the given block (Figure 3) where pixel P1 is black with eight neighbors.

P4	P5	P8
P3	P1	P2
P7	P9	P6

Figure 3. 3x3-sized block for Zhang-Suen's Algorithm.

Algorithm.

$A(x,y)$ = amount of white to black transitions in the series of a pixel's eight neighbors (x,y). For example, for $A(P1)$ sequence will be P5 P8 P5 P6 P9 P7 P3 P4 P5.
$B(x,y)$ = The number of black pixels among the eight neighboring pixels (x, y).

1. Test all the pixels and mark the pixels that satisfy the conditions of Pass1 which are:
   - The pixel should be black having eight neighbors.
   - $2 \leq B(x,y) \leq 6$ (For example, $2 \leq B(P1) \leq 6$).
   - $A(x,y) = 1$ (For example, $A(P1) = 1$).
   - A minimum of one neighboring pixel in the north, east, and south is white. (For example, at least one of the P5, P2, and P9 pixels is white).
   - A minimum of one neighboring pixel in the east, south, and west is white. (For example, at least one of the P2, P9, and P3 pixels is white).
2. All the pixels that satisfy these above conditions of Pass1 are set to white i.e. 0.
3. Test all the pixels and mark the pixels that satisfy the conditions of Pass2 which are:
   - The pixel is black and has eight neighbors.
   - $2 \leq B(x,y) \leq 6$ (For example, $2 \leq B(P1) \leq 6$)
   - $A(x,y) = 1$ (For example, $A(P1) = 1$).
   - At least one of the North, east, and west neighbor pixels is white. (For example, at least one of the P5, P2, and P3 pixels is white).
   - At least one of the North, south, and west neighbor pixels is white. (For example, at least one of the P5, P9, and P3 pixels is white).
4. All the pixels that satisfy these above conditions of Pass2 are set to white i.e. 0.
5. Both the above passes are repeated continuously until no more pixel is chosen for removal from both passes.

3.2 Feature Extraction

Fingerprint features are extracted for recognition at this step. Frequency domain, minutiae-based, ridge-based, and texture-based methods are common approaches; the minutiae-based method is the most widely used. Bifurcations and endpoints are the two main sites it identifies. By examining pixel neighbors, the Crossing Number Algorithm finds minutiae in images that have been binarized and thinned [14]. A bifurcation is indicated by a CN value of 3, whereas an endpoint is indicated by a CN value of 1.

Crossing Number Algorithm.

1. Define a 3*3 window for each ridge pixel.
2. Examine the surrounding area of each pixel in their window.
3. Calculate the CN (crossing number) value for each pixel. Half of the sum of the variations between neighboring pixel pairs in each of the eight neighborhoods is used to compute CN [7].
$$CN = 0.5 \sum_{i=1}^{8} |P_i - P_{i+1}| \quad (15)$$
4. Refer to Table 2 to determine what kind of pixel it is.

Table 2. Properties of crossing number [7]

CN	PROPERTY
0	Isolated Point
1	Ridge Ending Point
2	Continuing Ridge Point
3	Bifurcation Point
4	Crossing Point

3.3 Matching

This is the last stage. Usually, a matching score is computed to determine if the fingerprint matches the template [21]. The most common matching technique is minutiae-based matching. Bozorth3 algorithm is a widely used algorithm for minutiae-based matching.

Bozorth3. To ascertain whether the two fingerprints are the same, this algorithm calculates matching scores between minute details from the two fingerprints. By applying rotation and translation invariants, minutiae points are stored as xyt files and matched based on their location (x, y) and orientation (θ) [33].

Bozorth3 Algorithm.

1. For both test and template fingerprints, construct comparison tables for intra-fingerprint minutiae that include relative measures (distance and angle) for each minutia and every other minutia. Translation and rotation invariance are provided by these tables.
2. List compatible minutiae pairs between the test and template fingerprints in an inter-fingerprint compatibility table. Compatible entries are those that have comparable distances and related angles within tolerance [4].
3. Go through the inter-fingerprint compatibility table to determine the match score. The probability that the test and template fingerprints belong to the same person rises with a greater match score.

3.4 Analysis of Methods

The analysis of well-known fingerprint recognition methods is presented in Table 3.

Table 3. Analysis of Methods

Sr. No.	Author(s)	Year	Method	Description	Advantages	Limitations
1.	Wang et al. [34]	2007	Gradient-based method.	By obtaining partial derivatives of the grey intensities at every pixel, this approach creates gradient vectors.	Preserves ridge details while reducing noise.	Sensitive to errors in ridge orientation estimation.
2.	Al-Amri and Kalyankar [3]	2010	P-Tile Technique.	Using a percentile threshold, this segmentation technique separates pixels into foreground and background.	Simple to implement and effective for high-contrast images.	Struggles with low contrast and noisy images.
3.	Popović et al. [27]	2012	Averaging squared gradients methods.	To identify ridge flow, pixel intensity fluctuations are analyzed, squared and cut down on noise, and the prevailing ridge direction in each area is detected.	Robust to noise and suitable for local orientation analysis.	Computationally expensive compared to simpler methods.
4.	Chaudhari et al. [11]	2014	Crossing Number Algorithm.	Each pixel is examined by this program to ascertain if it is an endpoint, a bifurcation, or neither.	Simple and computationally efficient.	Can misidentify minutiae in low-quality images.
5.	Bhargavi and Jyothi [8]	2014	Otsu's threshold technique.	The algorithm divides the image into two classes and returns the optimal threshold intensity.	Resolves the problem of uneven light.	It is computationally expensive.
6.	Ashutosh Kumar Chaubey [10]	2016	Global Threshold Segmentation.	A threshold separates a greyscale image into two sections: the foreground and the background.	It is simple and fast.	Not very effective with uneven lighting or noise.
7.	Supatmi and Sumitra [33]	2020	Bozorth3.	Matches minutiae points using rotation and translation invariants.	High accuracy and efficiency.	Limited performance with low-quality images.
8.	Fakhrina and Fakhry [13]	2024	Zhang-Suen's Algorithm.	The thinning image is reduced to only one pixel by use of this iterative technique.	Fast and efficient for thinning.	May lose fine ridge details in some cases.

4. Results and Discussion

Studies show that Otsu's thresholding is a reliable method, especially for noisy fingerprint images, and the Crossing Number algorithm proves to be a very effective minutiae extraction technique. For large-scale identification systems without resource-constrained environments, the Bozorth3 algorithm gives good accuracy.

Research Gaps

The literature on fingerprint recognition emphasizes the need for more research by pointing out gaps and limitations. A significant problem is the scarcity of trustworthy testing datasets. While Abraham Junior Abbey's study [1] employed a dataset without defined geometry, Pavol Marák and Alexander Hambalík's research [16] might increase their dataset. Furthermore, the hierarchical Hough transform method is more complicated and computationally demanding than the generalized Hough transform, despite being quicker and more accurate. Some research gaps in fingerprint recognition systems are:

- Although it analyses images rapidly, the Zhang-Suen thinning technique has trouble with line endings, diagonal connectivity, and 2x2 grids. The minutiae extraction method and hardware-based ANN devices cause failures in fingerprint recognition as well, with FAR and FRR surpassing 40%.
- When utilizing the Gabor filter to enhance fingerprints, it is crucial to determine the frequency and direction of the ridges. Employing a curved Gabor filter for ridge frequency estimation is sensitive to errors in ridge orientation estimation.
- Maheshwar Kuchana, and Asish Malneni in their research paper [20] proposed Principal Component Analysis which does not give satisfactory results.
- A common technique used for segmentation is thresholding which is a powerful approach but the threshold parameter is difficult to adjust automatically. Furthermore, thresholding is only effective on a subclass of photos where the intensity of the objects is different from that of the backdrop [24].

5. Conclusion and Future Scope

Because fingerprints are unique, fingerprint recognition systems are frequently employed in biometrics. This review paper describes the fundamental workings of one such system. The four main procedures are fingerprint sensing, feature extraction, pre-processing, and matching. The paper's examination of key algorithms used at various stages highlights the necessity of pre-processing for feature extraction. Preprocessing is divided into further sub-stages which include:

- Segmentation is achieved using thresholding algorithm i.e. Global Thresholding and Otsu's Thresholding. While the former technique is simple and fast, it fails with uneven lighting or noise. The latter, however, resolves this problem even if it is computationally expensive.
- Local orientation estimation and local frequency estimation which are required parameters for image

enhancement. The most common method used for orientation estimation is the gradient-based method while for frequency estimation method is the Fourier-based method.

- Image enhancement is achieved using contextual filters. The Gabor filter is the most commonly used.

This paper has provided an overview of the major steps and algorithms used in fingerprint recognition systems. Having discussed several paradigms for pre-processing, segmentation, feature extraction, and matching, the work also outlines the advantages and disadvantages of e xisting approaches. Future studies can concentrate on enhancing recognition performance in difficult scenarios, as with imperfect, partial, or changed fingerprints. Accurate feature extraction and matching may be improved by combining deep learning and hybrid techniques.

Conflict of Interest
The authors state that there are no conflicts of interest pertinent to this work.

Funding Source
This research did not receive any grant from anywhere.

Author's Contribution
Both authors contributed equally to this work.

Acknowledgements
The authors are grateful for the reviewer's valuable comments that improved the manuscript.

References

[1] A. J. Abbey, "Fingerprint recognition system", **2022**.
[2] A. Z. Agha, R. K. Shukla, R. Mishra, R. S. Shukla, and P. Campus, "SSO Based Fingerprint Authentication of Cloud Services for Organizations", **2023**.
[3] S. S. Al-Amri and N. V. Kalyankar, "Image segmentation by using threshold techniques", *arXiv preprint*, arXiv:1005.4020, **2010**.
[4] F. Alonso-Fernandez, J. Bigun, J. Fierrez, H. Fronthaler, K. Kollreider, and J. Ortega-Garcia, "Fingerprint recognition", *Guide to Biometric Reference Systems and Performance Evaluation*, pp.**51–88**, **2009**.
[5] J. K. Appati, P. K. Nartey, E. Owusu, and I. W. Denwar, "Implementation of a Transform-Minutiae Fusion-Based Model for Fingerprint Recognition", *International Journal of Mathematics and Mathematical Sciences*, Vol.**2021**, No.**1**, **2021**.
[6] I. G. Babatunde, A. O. Charles, and O. Olatunbosun, "Uniformity level approach to fingerprint ridge frequency estimation", *International Journal of Computer Applications*, Vol.**62**, No.**22**, **2013**.
[7] R. Bansal, P. Sehgal, and P. Bedi, "Minutiae extraction from fingerprint images—a review", *arXiv preprint*, arXiv:1201.1422, **2011**.
[8] K. Bhargavi and S. Jyothi, "A survey on threshold based segmentation technique in image processing", *International Journal of Innovative Research and Development*, Vol.**3**, No.**12**, pp.**234–239**, **2014**.
[9] W. Bian, D. Xu, Q. Li, Y. Cheng, B. Jie, and X. Ding, "A survey of the methods on fingerprint orientation field estimation", *IEEE Access*, Vol.**7**, pp.**32644–32663**, **2019**. doi: 10.1109/ACCESS.2019.2903601.
[10] A. K. Chaubey, "Comparison of the local and global thresholding methods in image segmentation", *World Journal of Research and Review*, Vol.**2**, No.**1**, pp.**1–4**, **2016**.
[11] A. S. Chaudhari, G. K. Patnaik, and S. S. Patil, "Implementation of minutiae based fingerprint identification system using crossing number concept", *Informatica Economica*, Vol.**18**, No.**1**, pp.**17**, **2014**.
[12] W. El-Hajj-Chehade, R. A. Kader, R. Kassem, and A. El-Zaart, "Image segmentation for fingerprint recognition", *In the Proceedings of the IEEE Applied Signal Processing Conference (ASPCON)*, pp.**314–319**, **2018**. doi: 10.1109/ASPCON.2018.8748534
[13] F. A. Fakhrina and M. Fakhry, "Fingerprint pattern feature extraction for loop fingerprint pattern identification by Zhang-Suen and Stentiford thinning method", *Journal of Computer Science and Information Technology*, Vol.**16**, pp.**84–91**, **2024**. doi: 10.18860/mat.v16i2.28892
[14] P. Gnanasivam and S. Muttan, "An efficient algorithm for fingerprint preprocessing and feature extraction", *Procedia Computer Science*, Vol.**2**, pp.**133–142**, **2010**.
[15] C. Gottschlich, "Curved-region-based ridge frequency estimation and curved Gabor filters for fingerprint image enhancement", *IEEE Transactions on Image Processing*, Vol.**21**, No.**4**, pp.**2220–2227**, **2011**. doi: 10.1109/TIP.2011.2170696
[16] P. M. A. Hambalík, "Fingerprint recognition system using artificial neural network as feature extractor: design and performance evaluation", *Tatra Mt. Math. Publ.*, Vol.**67**, pp.**117–134**, **2016**.
[17] L. Hong, Y. Wan, and A. Jain, "Fingerprint image enhancement: algorithm and performance evaluation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.**20**, No.**8**, pp.**777–789**, **1998**. doi: 10.1109/34.709565
[18] A. K. Jain, A. Ross, and S. Prabhakar, "An introduction to biometric recognition", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol.**14**, No.**1**, pp.**4–20**, **2004**. doi: 10.1109/TCSVT.2003.818349
[19] N. N. B. Karo, A. Y. Sari, N. Aziza, and H. K. Putra, "The enhancement of fingerprint images using Gabor filter", *In Journal of Physics: Conference Series*, Vol.**1196**, No.**1**, **2019**.
[20] M. Kuchana and A. Malneni, "Fingerprint matching—An experimental approach", *International Journal of Research in Applied Science and Engineering Technology*, Vol.**8**, No.**6**, pp.**1734–1742**, **2020**.
[21] C. Liu, V. K. Mago, R. Cappelli, M. Ferrara, and D. Maltoni, "Minutiae-based fingerprint matching", *Cross Disciplinary Biometric Systems*, pp.**117–150**, **2012**.
[22] W. Mao, Y. Zhao, P. Pavlenko, Y. Chen, and X. Shi, "Innovative solutions for worn fingerprints: A comparative analysis of traditional fingerprint impression and 3D printing", *Sensors*, Vol.**24**, No.**8**, pp.**2627**, **2024**.
[23] M. Mua'ad, Z. A. Alqadi, and K. Aldebei, "Comparative analysis of fingerprint features extraction methods", *Journal of Hunan University Natural Sciences*, Vol.**48**, No.**12**, **2021**.
[24] Z. Niu and H. Li, "Research and analysis of threshold segmentation algorithms in image processing", *In Journal of Physics: Conference Series*, IOP Publishing, Vol.**1237**, No.**2**, pp.**022122**, **2019**.
[25] T. Orczyk and L. Wieclaw, "Fingerprint ridges frequency", *In Third World Congress on Nature and Biologically Inspired Computing*, IEEE, pp.**558–561**, **2011.** doi: 10.1109/NaBIC.2011.6089649
[26] A. Pillai, S. Mil'shtein, and M. Baier, "Algorithms for binarizing, aligning and recognition of fingerprints", *In VISAPP*, pp.**426–432**, **2011**.
[27] B. M. Popović, M. V. Banđur, A. M. Raičević, and D. Ranđelović, "Different methods for fingerprint image orientation estimation", *In 2012 20th Telecommunications Forum (TELFOR)*, IEEE, pp.**662–665**, **2012**.
[28] N. Quadri and S. S. Choudhary, "Performance analysis and designing of fingerprints enhancement technique based on segmentation, of estimation and ridge frequency, Gabor filters with wavelet transform", *Image*, Vol.**16**, pp.**18**, **2014**.

[29] K. Rajaram, N. B. Amma, and S. Selvakumar, "Convolutional neural network based children recognition system using contactless fingerprints", *International Journal of Information Technology*, Vol.**15**, No.**5**, pp.**2695–2705**, **2023**.

[30] K. Sasirekha and K. Thangavel, "A comparative analysis on fingerprint binarization techniques", *International Journal of Computational Intelligence and Informatics*, Vol.**4**, No.**3**, pp.**163–168**, **2014**.

[31] M. A. Sidek, "Fingerprint recognition using gray level co-occurrence matrices (GLCM) and discrete wavelet transform (DWT)", *In IRC*, **2015**.

[32] S. A. Sudiro, I. P. Wardhani, B. A. Wardijono, and B. Handias, "Fingerprint matching application using hardware based artificial neural network with MATLAB", *In 5th International Conference on Electrical, Electronics and Information Engineering (ICEEIE)*, IEEE, pp.**66–70**, **2017**, doi: 10.1109/ICEEIE.2017.8328764

[33] S. Supatmi and I. D. Sumitra, "Fingerprint matching using Bozorth3 algorithm and parallel computation on NVIDIA compute unified device architecture", *In IOP Conference Series: Materials Science and Engineering*, IOP Publishing, Vol.**879**, No.**1**, pp.**012109**, **2020**.

[34] Y. Wang, J. Hu, and F. Han, "Enhanced gradient-based algorithm for the estimation of fingerprint orientation fields", *Applied Mathematics and Computation*, Vol.**185**, No.**2**, pp.**823–833**, **2007**.

[35] R. Josphineleela and M. Ramakrishnan, "A new approach of altered fingerprints detection on the altered and normal fingerprint database", *Indian Journal of Computer Science and Engineering (IJCSE)*, Vol.**3**, No.**6**, pp.**818–821**, **2012**.

[36] A. Sharma and M. Kaur, "Automatic segmentation for separation of overlapped latent fingerprints", *International Journal of Computer Sciences and Engineering*, Vol.**6**, No.**7**, pp.**484–490**, **2018**.

[37] P. E. Gundgurti and S. B. Kulkarni, "Latent fingerprint enhancement and segmentation through advanced deep-learning techniques", *International Journal of Computer Sciences and Engineering*, Vol.**16**, No.**1**, pp.**72–93**, **2025**.

AUTHORS PROFILE

Himanshi Sharma completed her B.Tech. degree in Computer Science and Engineering from Guru Nanak Dev University, Amritsar, Punjab, India, in 2025. She has a strong interest in software development, web technologies, and database management. With a keen focus on full-stack development, she works with modern frameworks and architectures to build scalable applications. Additionally, she explores data science and machine learning, leveraging AI-driven approaches to optimize software systems. Actively involved in research and technical projects, she continues to expand her knowledge and contribute to innovative solutions in the field of computing.

Hardeep Singh received the B.Tech. degree and the M.Tech. degree in Computer Science and Engineering from Guru Nanak Dev University, Amritsar, Punjab, India, in 2010 and 2011, respectively, where he recently completed the Ph.D. degree with the Computer Science department. Since 2012, he has been an assistant professor with the Computer Engineering and Technology Department at Guru Nanak Dev University, Amritsar, Punjab, India. He has published more than 21 research articles. His research interests include software engineering, social network analysis, and data mining.

