Investigating the impact of thresholding and thinning methods on the performance of partial fingerprint identification systems: a review

Mahesh Joshi⬤,* Bodhisatwa Mazumdar, and Somnath Dey
Indian Institute of Technology Indore, Department of Computer Science and Engineering, Indore, Madhya Pradesh, India

Abstract. A fingerprint identification system is an application of pattern recognition and image processing. The performance of a fingerprint-based biometric system relies on pre-processing techniques employed on fingerprint images. Especially, thresholding and thinning methods are used to detect minutiae points representing local features and are often utilized to identify a person uniquely. However, studies on partial fingerprints exposed the MasterPrint vulnerability for partial fingerprint identification systems wherein the system performs non-unique user identification. The thresholding and thinning techniques may lead to spurious minutiae generation and stimulate huge MasterPrints. Here, we investigate the impact of thresholding and thinning methods on identification accuracy and the percentage of MasterPrint generated using a partial fingerprint identification method. The experiments comprise four thresholding methods, namely, iterative optimal thresholding, Otsu's global image thresholding, Niblack local thresholding, and Bernsen's local image thresholding. Furthermore, it employs four thinning methods, namely, Khalid, Mariusz, Marek thinning algorithm, Khalid, Marek, Mariusz, Marcin thinning algorithm, Hilditch thinning algorithm, and Stentiford thinning algorithm. The results demonstrate that the identification accuracy and percentage of MasterPrint generated varies significantly by replacing the underlying pre-processing methods. Consequently, each combination of thresholding and thinning methods might not be suitable for user identification in high-security applications using a partial fingerprint identification method. The investigation outcomes provide the guidelines to demonstrate the robustness of a partial fingerprint identification method. © *2023 Society of Photo-Optical Instrumentation Engineers (SPIE)* [DOI: 10.1117/1.JEI.32.1.010901]

1 Introduction

Fingerprint biometric systems (FBSs) require minimal involvement from an individual to capture the sensed finger impression. Also, these systems have high accuracy and are affordable in implementation. Hence, FBSs are widely used as commercial applications for access control and user identification. An FBS is employed for user authentication and identification applications. During the enrollment process, the system administrator enrols a legitimate person by storing multiple samples of their finger into the database as encrypted templates. In the authentication scenario, the system verifies the user's claim by comparing the recently acquired fingerprint template with only those stored templates, which possess the claimed identity. The claim may get accepted or rejected based on the similarity score between the templates. However, an identification system fetches all the stored templates to compare with the recently created template. The identity of the stored template generating the highest similarity score becomes the unknown user's identity.

In general, FBSs perform ridge pattern recognition utilizing the features extracted from the fingerprint image for access control, security applications, user authentication, and identification.

---

*Address all correspondence to Mahesh Joshi, phd1701101004@iiti.ac.in

Fingerprint of a user captured by the sensors at different occasions usually differs due to various physiological factors of an individual and environmental conditions. Hence, these images are pre-processed for better recognition accuracy. Thresholding and thinning are among the predominantly employed pre-processing methods in fingerprint recognition. Usually, fingerprint images are enhanced using the thresholding technique to ensure that samples of the same finger captured at multiple instances appear similar. Thresholding enhances the dark region from the fingerprint termed as ridges and reduces the intensity of the remaining portion called valleys. Minutiae are the local fingerprint features formed at the point where three ridges emerge, or a single ridge ends.[1] However, as minutia represents a single pixel within a ridge, a binarized fingerprint image obtained by thresholding may not facilitate accurate minutiae detection in multiple samples of the same fingerprint.[2] Therefore, a single-pixel version of fingerprint ridges is obtained using the thinning of the binarized image. Consequently, a combination of thresholding and thinning methods for minutiae-based feature extraction improves person recognition accuracy of an FBS. Ratha et al.[3] employed projection-based thresholding approach[3] and Human Information Processing Laboratory's Image processing System library[4] for thinning purpose. Feng[5] performed thinning of fingerprints using local threshold-based thresholding method and Guo and Hall method[6] was used for thinning the binarized fingerprints. Joshi et al.[7] used adaptive thresholding approach[8] and Zhang–Suen thinning algorithm[9] for thresholding and thinning of partial fingerprints in their experimentation.

FBS are among the most reliable means of individual authentication, authorization, and identification due to low-cost devices, user convenience, low response time, and high accuracy. However, Roy et al.[10] investigated the partial fingerprint identification systems and observed that partial fingerprints might not be unique to every individual. The authors termed MasterPrint to those partial fingerprints identifying at least 4% distinct subjects from the enrolled database. The authors experimented on partial fingerprints cropped from FVC 2002 full fingerprint dataset using commercial VeriFinger software development kit. The results of their investigation concluded that a dictionary of the top five MasterPrints could disclose the identity of more than 60% unique subjects. Bontrager et al.[11] investigated the feasibility of using latent variable evolution to generate complete image-level synthetic MasterPrints, termed as DeepMasterPrints. The probability of a successful attack using a dictionary of DeepMasterPrints was demonstrated to be significantly high on NIST Special Database 9 fingerprint dataset[12] and FingerPass DB7 dataset.[13] These statistics proved the severity of the MasterPrint vulnerability.

Pre-processing fingerprint grayscale images is a crucial step in fingerprint recognition. Due to environmental and physiological conditions, the fingerprint images of the same finger captured at different instances are not similar. Hence, pre-processing in fingerprint recognition was less explored as it was a common practice in the literature to allow tolerance during the feature matching step. The investigation by Roy et al.[10] on partial fingerprint identification disclosed the MasterPrint vulnerability. The investigators reported accepting approximately similar features as one of the reasons for generating MasterPrint.[10] The first method to address the MasterPrint vulnerability by opting for strict feature matching was presented by Joshi et al.[7] The method performed feature extraction based on geometric constructs formed using adjacent minutiae. Hence, the method was expected to produce marginally varying identification rate and percentage of MasterPrint generated if the underlying pre-processing approaches were replaced with other thresholding and thinning algorithms in the literature. The work in this paper was carried out to investigate how far the identification accuracy and percentage of MasterPrint generated varies by replacing the pre-processing steps with other thresholding and thinning methods in the literature.

Joshi et al.[7] proposed a minutiae geometry-based MasterPrint mitigation method, and employed adaptive thresholding approach[8] for thresholding of the partial fingerprints and Zhang–Suen algorithm[9] to obtain thinned fingerprints. The approach achieved up to 97% accuracy and generated 0.1% MasterPrints. However, as the image processing literature contains numerous techniques for thresholding and thinning, it is imperative to investigate the robustness of these methods in terms of identification accuracy and addressing the MasterPrint vulnerability for a partial fingerprint identification system. In this regard, the paper provides an exhaustive experimentation comprising four thresholding and four thinning methods to study their impact on the partial fingerprint identification system performance.

The highlight of the investigations from the paper is to answer the following questions,

1. Given a rotation invariant, robust, minutiae-based local feature extraction method, how far do the results vary if the fingerprint pre-processing is carried out using diverse thresholding and thinning methods?
2. What is the more acceptable measure for a fingerprint recognition method to signify appropriate pre-processing, the average count of minutiae in the fingerprint or average minutiae density?
3. Do false minutiae detection and removal methods help improve the fingerprint identification system performance if the underlying pre-processing does not yield the thinned fingerprint image that preserves ridge connectivity and ridge pattern?
4. Are the existing image pre-processing methods beneficial for fingerprint images, or is there a requirement for a rigorously tested fingerprint pre-processing method?

The organization of the rest of the paper is as follows. Section 2 describes an introduction of various components involved in FBS. Section 3 introduces various thresholding approaches employed in the experimentation. Section 4 provides a brief introduction to the thinning approaches that follow the thresholding process in the investigation. The experimental setup to investigate the impact of various pre-processing methods on the performance of a partial fingerprint identification and MasterPrint mitigation scheme is presented under Sec. 5. The result analysis and performance evaluation of the possible combinations of thresholding and thinning approaches is given in Sec. 6. The discussion on the results follows in Sec. 7. Finally, Sec. 8 concludes the paper.

2 Fingerprint Biometric System: An Overview

An FBS comprises several components performing a dedicated function in fingerprint recognition. The steps involved in minutiae-based FBS are shown in Fig. 1. A sensor on an input device captures the fingerprint for the portion of a finger that touches the sensor. The area of a finger that touches the sensing device forms a dark region termed as ridges. However, the remaining portion of the finger results in valleys. Thus, fingerprint acquisition involves creating a ridge-valley pattern for an input finger. However, distortions due to varying pressure exerted, environmental factors, sweat, or dry fingers are certain when mapping a three-dimensional fingertip onto a two-dimensional plane.[14] Hence, fingerprint pre-processing is carried out to facilitate feature extraction. Fingerprint pre-processing usually involves enhancing the quality of the fingerprint image, i.e., producing high contrast between ridges and valleys, to facilitate minutiae-based local feature extraction.[15]
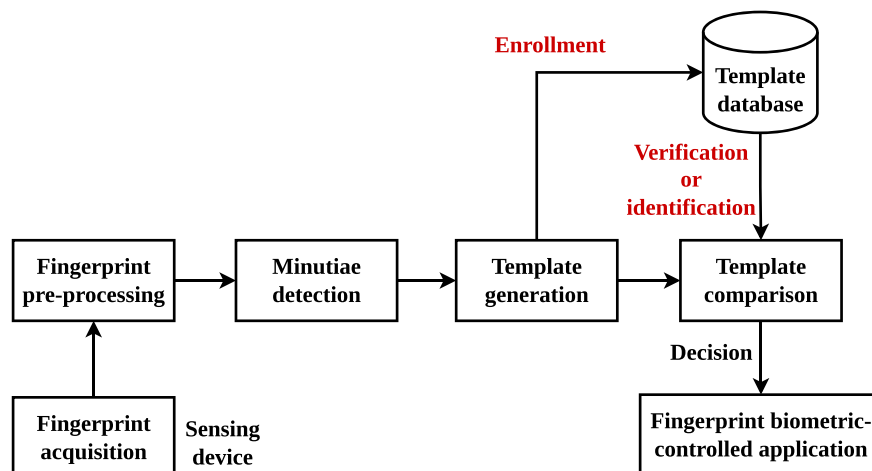

Fig. 1 Various components involved in minutiae feature-based FBS.

Image enhancement techniques modify the intensities of pixels in an image so that it can be more suitable for a specific application. An enhancement method appropriate for one application may not deliver the expected outcome for another. Fingerprint image enhancement approaches can be categorized as histogram-based, filtering-based, or transformation-based. Histogram specification is an image processing approach that utilizes image histograms to adjust contrast.[16] The histogram depicts the brightness distribution and is primarily used to enhance the local contrast without affecting the overall contrast. Image enhancement using histogram equalization (HE) represents the statistical relationship between each gray level in the image and the number of pixels that appear in the gray level. HE reflects the frequency of each gray level in the image. However, it is often necessary to consider combining a variety of simple and effective algorithms or fusing other enhancement techniques to achieve the final enhancement effect.[17] Furthermore, HE techniques add noise to the output image and increase background contrast.[18,19]

Usually, filter-based methods generate image frequency spectrum data. Filters are utilized in pre-processing to solve two objectives. First, to fill the small gaps, i.e., low-pass effect, in the ridge direction. Next, increase the band-pass effect, i.e., discrimination between ridges and valleys in the direction orthogonal to the ridges.[20] Gabor filters act as band-pass filters to remove the noise and preserve the true ridge-valley pattern. The convolution nature of the Gabor filter contributes to high computational complexity, leading to an overall increase in the running time of the user verification and identification process.[14] Furthermore, applying Gabor filters requires feeding reliable estimation inputs of the local context, i.e., the local orientation and ridge frequency to Gabor filters. Moreover, failing to estimate the local context correctly may lead to the creation of artefacts in the output image, consequently increasing the number of errors in user identification or verification.[14] The major drawbacks of Gabor filters include their limited, i.e., approximately one octave, maximum bandwidth, and when seeking broad spectral information with maximal spatial localization, they are not optimal.[14] Frequency domain techniques are computationally less efficient and require more processing resources to implement.[14] Thresholding is a type of transformation-based image enhancement method. Thresholding generally involves two steps, i.e., determining a gray threshold according to some objective criteria and assigning each pixel to one class of background or foreground. The objective criteria may consider neighboring pixel intensities or the intensities from the entire image. Consequently, a thresholding method can be a local or global thresholding approach. The thresholding of a digital image is beneficial for segmenting a region of interest from the background. When applied to fingerprint images, thresholding solves two purposes. It isolates a fingerprint from the rest of the untouched area of the sensing device and highlights ridge patterns. Usually, thresholding requires fewer computations and, therefore, is relatively easy to implement compared to other techniques.[21]

The fingerprint recognition involving minutiae-based features employs direct grayscale, binarized, and thinned images.[22] Since minutiae are single-pixel locations in the fingerprint image, a thinned fingerprint image facilitates minutiae detection. Hence, the fingerprint image after thresholding is often thinned before minutiae detection. The crossing number (CN) is the most widely used and accepted measure for minutiae detection.[23] The fingerprint template of a user comprises encrypted minutiae-based features. The templates for each user are stored in the template database during the enrollment phase. During the verification and identification of a user, the stored templates are retrieved and compared against the recently created fingerprint template. The templates corresponding to the identity claimed by a user are fetched from the database during the verification process. However, the newly acquired fingerprint template is compared with each stored template in the identification process. The system generates a similarity score, a numeric value, as the measure showing the chances that the templates under comparison belong to the same user. Thus, a high score corresponds to a higher probability that the templates are generated from the same finger. The user's claim is verified during verification as accepted or rejected. However, an unknown user's identity is declared in the identification scenario.

3 Thresholding Methods

Image pre-processing techniques improve the quality of an image while preserving its original contents.[24] However, it is not always useful to enhance an image as, more often, crucial
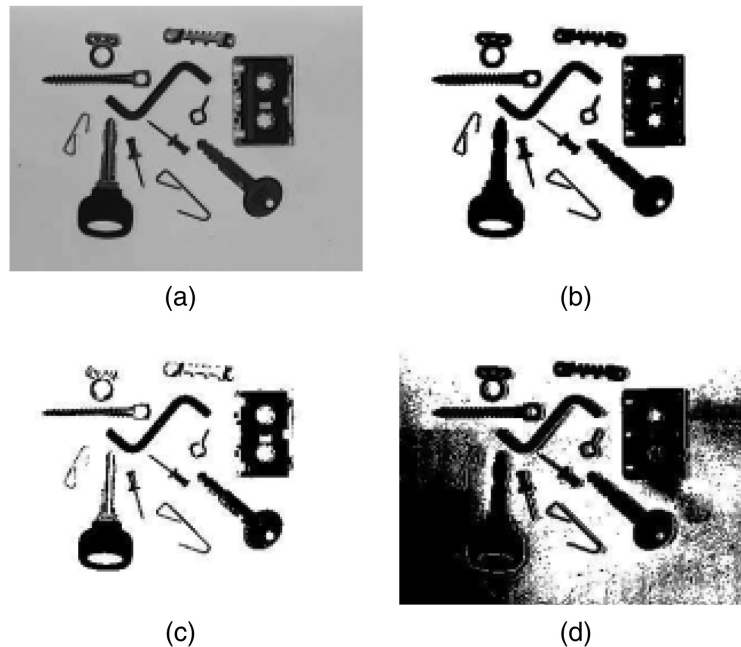
(a)

(b)

(c)

(d)

Fig. 2 Image thresholding (a) original image; (b) threshold segmentation; (c) low threshold; and (d) high threshold.[24]

information may get lost due to such techniques. There is an infinitesimally small probability of acquiring exactly the same fingerprints by the biometric system each time a user touches the biometric sensor. The weather conditions such as, moisture or heat, sweating around the fingertip, the orientation of the finger, and pressure exerted on the sensor surface are some of the reasons leading to dissimilar fingerprint samples of the same finger acquired at different instances. Therefore, thresholding approaches are generally employed to produce approximately similar ridge patterns from several samples of the same finger. These techniques are broadly categorized as global, local, and hybrid methods.[25] Figure 2 shows a grayscale image and the transformed images after applying threshold segmentation, low threshold, and high threshold. The figure shows that the threshold segmented version is the most appropriate for object recognition as it has retained the shapes and allowed negligible noise during the transformation. The fingerprint biometric researchers and vendors use image enhancement techniques for research activities and consumer products. However, every method may not be beneficial for a biometric application and may adversely affect the system accuracy by removing true minutiae or inserting false minutiae. Consequently, such practices leads to an incorrect and low identification rates and high MasterPrint generation. Therefore, it is imperative to investigate the impact of various combinations of pre-processing techniques and examine the system performance.

Shaikh et al.[26] evaluated six thresholding methods for performance bench-marking of various global and local thresholding methods toward fingerprint-based biometric recognition system. Their work forms the basis for selecting the thresholding approaches employed in our experiments. The thresholding algorithms used during the experiments include iterative optimal thresholding,[24] Otsu's global image thresholding,[27] Niblack local thresholding,[28] and Bernsen local image thresholding.[29] The following subsection briefly discusses these methods.

3.1 Iterative Optimal Thresholding

The iterative optimal thresholding approach models the image pixels as a histogram generating normal distributions for the area of interest, i.e., the ridge portion and the background region, also known as the valley portion.[30] The approach considers the minimum probability lying between the two distributions' maxima as the initial threshold, T_i. Furthermore, the method iteratively updates T_i to minimize the segmentation error.[24] The algorithm considers the value of T_i for which the segmentation error cannot be further minimized as the optimal threshold, T_o.
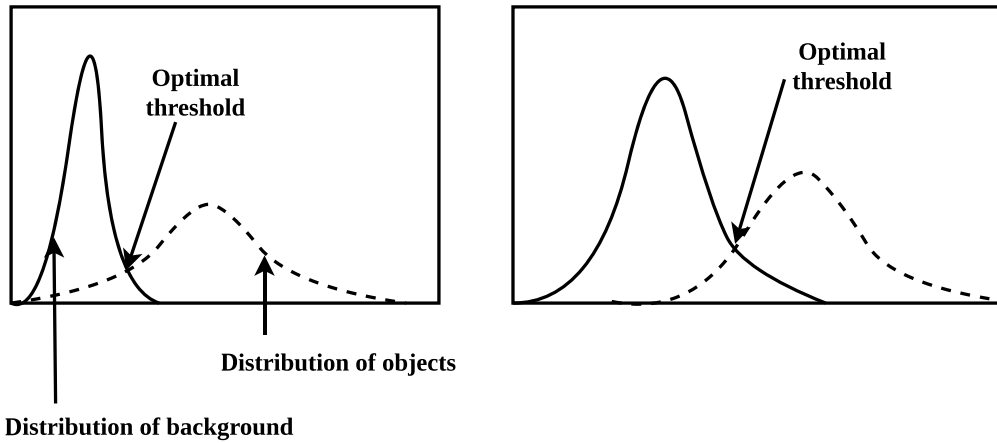
Fig. 3 Gray-level histograms approximated by two normal distributions—probability distributions of background and object of interest. The threshold is set to give minimum probability of segmentation error.[24]

Figure 3 shows the global level histograms to decide the initial optimal threshold. The figure depicts gray-level histograms approximated by two normal distributions—probability distributions of background and object of interest. The final optimal threshold is set to give minimum probability of segmentation error.[24]

3.2 Otsu's Global Image Thresholding

The approach returns an intensity as a threshold to divide the image pixels as background and foreground. The algorithm iteratively tries to maximize the inter-class intensity variance or minimise the intra-class intensity variance.[27] It computes histogram, $H(i)$, with L bins and probability, $p(i)$, for each intensity, i, within the image. Let t be the threshold under consideration. The probability of a pixel to be a background, W_b, and foreground, W_f, is computed as below,

$$W_b(t) = \sum_{i=0}^{t-1} p(i), \tag{1}$$

$$W_f(t) = \sum_{i=t}^{L-1} p(i). \tag{2}$$

The approach then calculates within-class variance σ_w as

$$\sigma_w^2(t) = W_b(t) \times \sigma_b(t) + W_f(t) \times \sigma_f(t), \tag{3}$$

where $\sigma_b(t)$ and $\sigma_f(t)$ are the background and foreground gray level variances, respectively. It returns the threshold corresponding to $\min(\sigma_w^2(t))$ as the desired threshold. Figure 4 shows a grayscale character A and its transformed version after applying Otsu's global image thresholding method. The difference in the foreground and background in noticeable after thresholding.

3.3 Niblack Local Thresholding

Niblack algorithm is a local thresholding approach.[28] It uses a fixed-sized rectangular window, w, surrounding a reference pixel, p, and slides the window over the entire image, I. The window size is application dependent and default value is 15. The approach computes the local mean, μ_w, and standard deviation, σ_w, for the window region.[31] The following equation decides the local threshold, T_w, for the given window, w

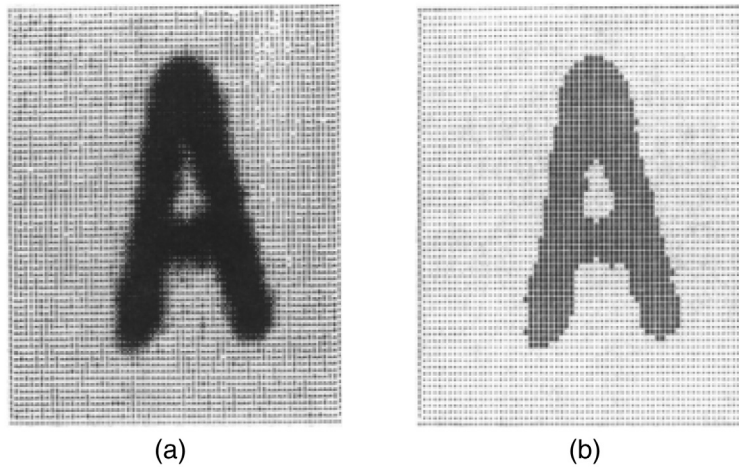$$T_w = \mu_w + (-0.2) \times \sigma_w. \tag{4}$$

Fig. 4 Image thresholding using Otsu's global image thresholding method.[27]
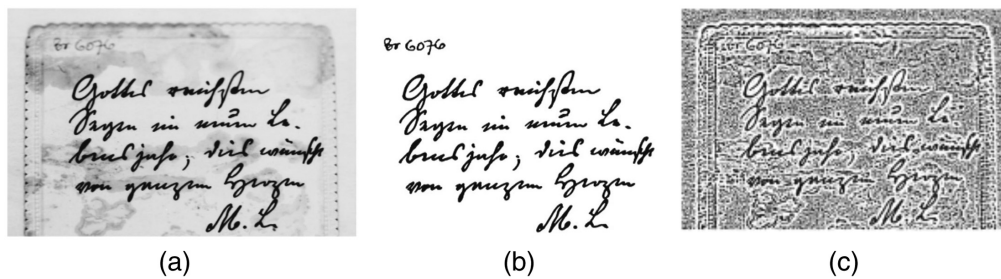

Fig. 5 Image thresholding using Niblack's local thresholding method: (a) test image; (b) ground truth; and (c) after applying Niblack thresholding method.[33]

The experimental results show that the approach generates thresholding noise in the non-desired gray region.[32] Figure 5 shows a test image, the ground truth, i.e., the reference image from the HDIBCO 2016 dataset, and the image after applying Niblack's thresholding. The resulted image after thresholding shows heavy background noise in the non-text, i.e., shadow, region.

3.4 *Bernsen's Local Image Thresholding*

Bernsen's approach is another local thresholding method.[29] For a given image, I, the approach initializes local contrast, l, and neighborhood window size, w, e.g., $l = 15$ and $w = 3$. The algorithm then assigns the lowest and highest gray levels within the window size $w \times w$ as $I_{\min}$ and $I_{\max}$, respectively. The local threshold, Th_l, and the contrast measure, C_m, are computed using the following equations [29]

$$Th_l = \frac{I_{\max} + I_{\min}}{2},\tag{5}$$

$$C_m = I_{\max} - I_{\min}.\tag{6}$$

If $C_m > l$, i.e., a non-uniform grayscale image, then the neighborhood belongs to the same class (background or foreground). Otherwise, a global thresholding approach decides the local threshold.[34] Figure 6 shows a sample text image before and after applying Bernsen's thresholding method. The difference between the background and the foreground is evident after thresholding. A sample grayscale fingerprint image and its various versions after applying the thresholding methods employed in the investigations are shown in Fig. 7. The figures show that different thresholds were applied by each method on the original image. Furthermore, the ridge

Fig. 6 Image thresholding using Bernsen's local thresholding method: (a) original image and (b) image after thresholding with $l = 15$ and $w = 3$.[35]
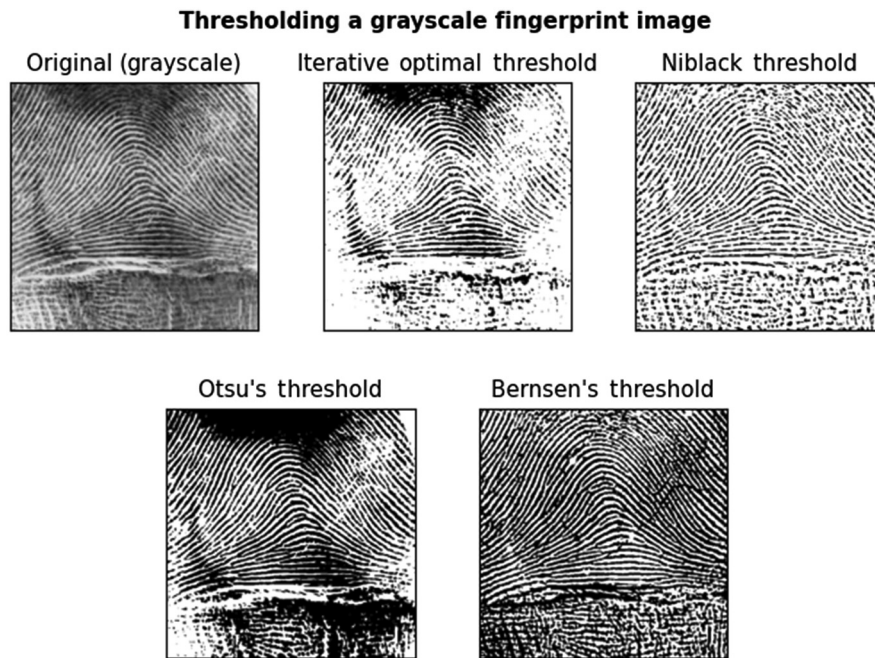

Fig. 7 A grayscale fingerprint image and its various versions after applying thresholding approaches employed in the investigation.

patterns in the image generated by Bernsen's method are clearly distinguishable compared to other methods.

4 Thinning Approaches

A thinning algorithm produces a single-pixel skeletal structure that highlights prominent features from the original image. In general, a binarized image is employed for the thinning process to ensure connectivity among various regions within the image. It helps in determining the topological and metric-based properties to count, measure, and classify relevant features. However, local noises in the image easily affect the resultant skeleton.[36] Thinning algorithms are mainly utilized for object representation, detection, manipulation, comparison, tracking, recognition, and compression.

Minutiae are the most widely exercised and accepted features utilized in FBSs.[11] In general, minutiae points are stored as their (x, y) coordinates, orientation angle, and type, i.e., ridge or bifurcation. Minutiae-based FBSs employ minutiae correlations within an image during their comparison. Therefore, locating minutiae most accurately within two samples of the same finger is highly desirable. A robust thinning approach accepting a correctly binarized fingerprint image can improve system performance in such circumstances. However, a given thinning approach
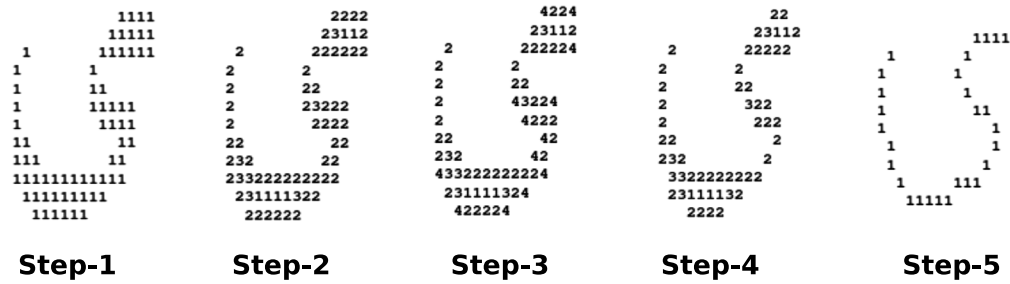
Fig. 8 Step involved in KMM thinning algorithm.[43]

may also adversely reduce the recognition accuracy if it produces substantial false minutiae due to several breaks in the ridge patterns.

Nazarkevych et al.[37] evaluated the effectiveness of image thinning methods in biometric security systems. The authors analyzed Zhang and Suen[9] and Hilditch thinning algorithm,[38] among others. Saha et al.[39] presented a comprehensive review of existing thinning methods and their applications. The authors discussed thinning approaches applicable to fingerprint analysis. The work by Nazarkevych et al.[37] and Saha et al.[39] encouraged us to experiment with Hilditch thinning algorithm[38] and Stentiford thinning method.[40] The designers of Saeed et al.[41] algorithm and its modified version by Tabedzki et al.[42] have claimed these approaches as a universal algorithm for image thinning. Hence, these two methods were utilized to verify the claim for their robustness in partial fingerprint identification. The following subsections briefly explain the thinning methods used in the investigation.

4.1 KMM Thinning Algorithm

The Khalid, Mariusz, Marek (KMM) approach accepts a binarized image wherein binary 1 represents the dark region to be thinned. Next, it converts the 1's adjacent to the boundary 0's and in the open elbow bends into 2 and 3, respectively.[43] The method considers non-zero positions in the image and figures out the locations, x, having 2, 3, or 4 sticking neighbors. It changes all such x to 4. A predefined table, Deletion Array, provides the sum for x that is the probable target for removal. It iteratively eliminates such x, assuring that the connectivity is intact. Finally, the approach excludes unnecessary 2's and 3's until it produces a single-pixel width thinned image.[44] Figure 8 shows various steps involved in thinning a binarized image using KMM thinning algorithm. The resultant thinned image preserves the pixel connectivity from the input image.

4.2 K3M Thinning Algorithm

The Khalid, Marek, Mariusz, Marcin (K3M) algorithm is a modified version of KMM.[41] The algorithm iterates over seven phases until it generates a thinned image. These phases can be summarised as below,

1. Mark boundry pixels, b
2. Remove b's with 3 adjacent neighbors
3. Remove b's having 3 or 4 adjacent neighbors
4. Remove b's with 3, 4, or 5 adjacent neighbors
5. Remove b's with 3, 4, 5, or 6 adjacent neighbors
6. Remove b's with 3, 4, 5, 6, or 7 adjacent neighbors
7. Unmark remaining boundary pixels

If the current iteration of these seven phases modifies the image, the image undergoes another iteration. Otherwise, the algorithm stops resulting in a thinned image.[42] Figure 9 shows several graphical symbols and their resultant thinned version after applying K3M thinning algorithm. The algorithm does not produce properly thinned images for small and almost thinned images.
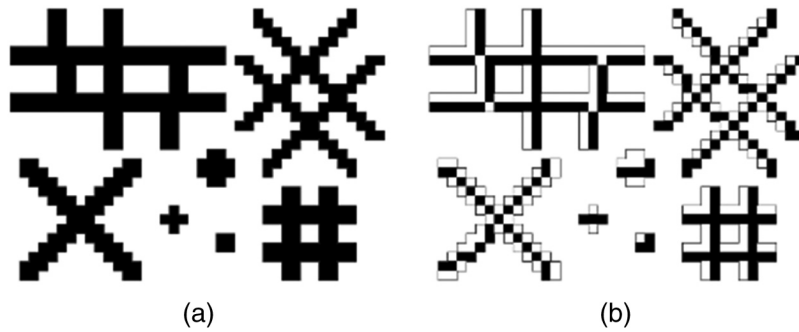
Fig. 9 Thinning graphical symbols with K3M: (a) original shapes and (b) corresponding thinned shapes.[41]

P_9	P_2	P_3
P_8	P_1	P_4
P_7	P_6	P_5

Fig. 10 Neighborhood pixel nomenclature in Hilditch approach.

4.3 *Hilditch Thinning Algorithm*

The Hilditch thinning algorithm has two variants; one uses a 3×3 window while the other uses 4×4 window.[38] The experiments in this paper employed 3×3 window. The neighborhood pixel nomenclature is as shown in Fig. 10. The algorithm iteratively decides if the reference pixel P_1 should be removed based on the following five conditions,[45]

1. Eliminate P_1, if it is a part of the skeleton.
2. Preserve P_1, if it lies on the border of a skeleton.
3. Preserve P_1, if it is an isolated pixel.
4. If P_1 is a connecting pixel, preserve it.
5. Remove P_1, if it has at least one neighbor.

The algorithm considers all the above conditions to decide if P_1 should be preserved or eliminated. It finally stops when the recent iteration encounters no pixels for removal. Figure 11 depicts a sample fingerprint image and its thinned version. The top right corner portion of the thinned image shows an improper thinning operation as a single ridge is split into several tiny ridges.

4.4 *Stentiford Thinning Algorithm*

The templates used in the Stentiford algorithm to decide if a pixel should be removed are shown in Fig. 12. It considers only three locations, marked with the circle, in the neighborhood of a pixel. The algorithm steps are as below,[40]

1. Traverse the image left-to-right downwards to locate pixels having T_1 pattern,
2. If the central pixel at such place is not an end point, i.e., the last pixel, and have connectivity value[47] as 1, mark it for removal,
3. Repeat step 1 and 2 for each pixel over the image,
4. Repeat steps 1, 2 and 3 for T_2 traversing upwards left-to-right, for T_3 traversing right-to-left upwards, and for T_4 traversing downwards right-to-left.
5. Remove all marked pixels.
6. If there was any deletion in step 5 of current iteration, repeat steps 1 to 5 else stop.

(a)   (b)
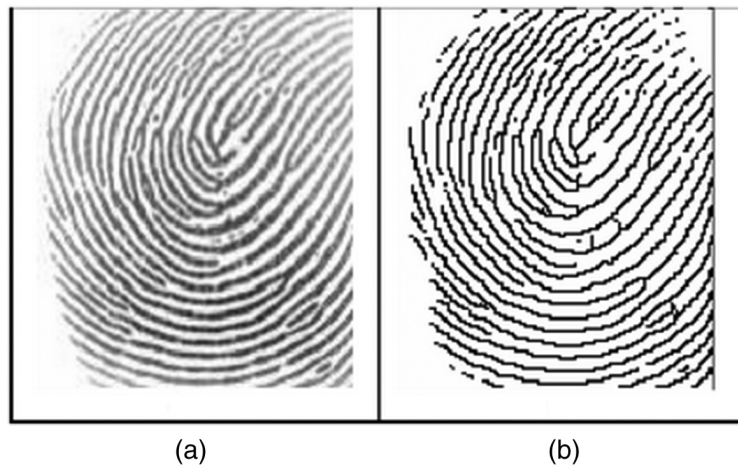
Fig. 11 Fingerprint thinning : (a) original image and (b) corresponding thinned image.[46]

T_1 T_2 T_3 T_4
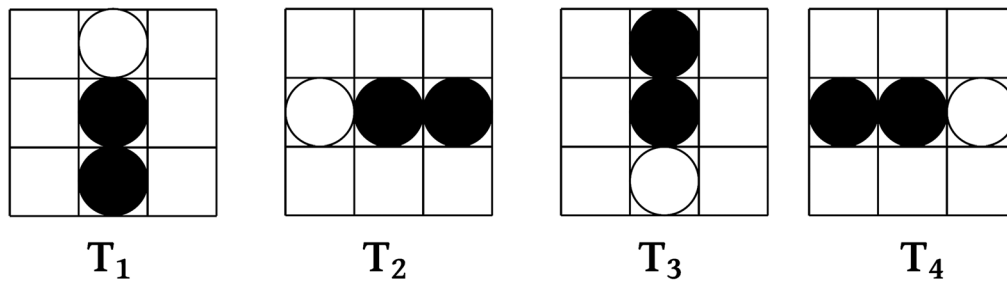
Fig. 12 The templates for deciding pixels for removal in Stentiford thinning algorithm. It considers only three locations, marked with circle, for the conclusion.

(a)   (b)
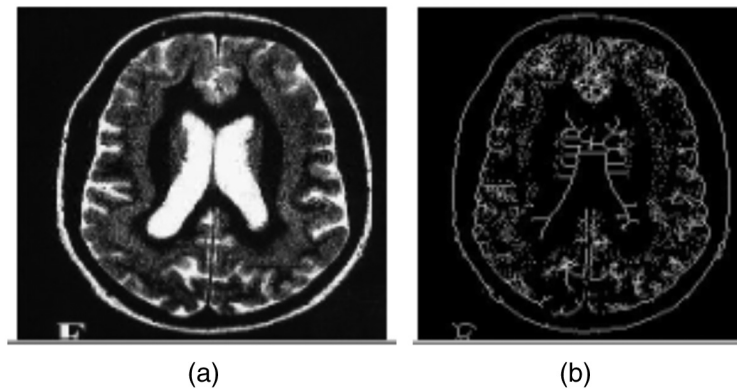
Fig. 13 Thinning of MRI image using Stentiford thinning algorithm: (a) original MRI image and (b) corresponding thinned image.[48]

Figure 13 shows a grayscale MRI image and corresponding thinned image obtained using Stentiford thinning algorithm. The thinned image accurately reflects the features from original image.[48]

5 Experimental Setup

Recently, Joshi et al.[7] proposed a minutiae geometry-based partial fingerprint identification approach targeted toward MasterPrint mitigation. The method experimented on partial fingerprint datasets cropped from five benchmark full fingerprint datasets delivered up to 97%

Table 1 Summary of the cropped partial datasets (150 × 150 px) used in the experiments.

Dataset	FVC2002 DB1_A	FVC2002 DB2_A	CrossMatch	sd302b	sd302d
Sensor technology	Optical sensor	Optical sensor	Optical sensor	Touch-free	Touch-free
Full fingerprints	800	800	408	920	1600
Total subjects	100	100	51	92	200
Samples per subjects	8	8	8	10	8
Image resolution	500 dpi	569 dpi	500 ppi	1000 ppi	500 ppi
Partial dataset size	3549	2767	2134	2098	2960

accuracy and generated 0.1% MasterPrints. The details of various datasets employed during the investigations are shown in Table 1. The authors employed adaptive thresholding approach[8] for thresholding and Zhang–Suen thinning algorithm[9] for thinning the binarized partial fingerprints. Minutiae detection was carried out by employing the metric, CN.[49] However, inappropriately binarized or thinned fingerprint may generate false minutiae affecting the system performance. Hence, Kim et al.[50] algorithm was employed to detect and remove false minutiae. The algorithm performs post-processing on the detected minutiae using various parameters, such as ridge flow, ridge orientation, connectivity, and distance between minutiae. It detects and eliminates five different types of false minutiae, namely broken ridge, bridge, short ridge, hole, and triangle.[50]

The investigation in this paper aims to study the impact of various combinations of thresholding and thinning methods in Joshi et al.[7] method toward identification accuracy and percentage of MasterPrint generated. The original papers that introduced the four thresholding and thinning methods employed during the investigation have shown satisfactory results. However, the impact of their cross combinations has not yet been reported. This work does not attempt to comment on a particular method or ascertain that a specific pair is preferable. Instead, this work evaluates the robustness of the Joshi et al.[7] method under diverse pre-processing conditions. The experiments were carried out using Joshi et al.[7] method on the partial datasets used in their paper. However, instead of adaptive thresholding approach[8] for thresholding and Zhang–Suen thinning algorithm[9] for thinning, 16 combinations of the selected thresholding and thinning methods were applied. So, there are 80 individual experiments as a part of the investigation.

The experiments were conducted on a desktop system with 64-bit Ubuntu 20.04.2 LTS (Focal Fossa) operating system having 64 GB internal memory (RAM) and Intel® Xeon(R) CPU E5-1620 v3 @ 3.50 GHz ×8 processor. The terminology used for various combinations of thresholding and thinning approaches on different datasets is given in Table 2. We followed D_B_T format for each combination, where D refers to the dataset, B specifies a thresholding approach, and T denotes the thinning approach. For example, entry 1_1_1 refers to the combination of iterative optimal thresholding and KMM thinning algorithm experimented on CrossMatch Sample DB dataset. Figure 14 shows a sample image from CrossMatch Sample DB dataset and its thinned version from each combination of thresholding and thinning approach as mentioned in Table 2. The thinned image for the same fingerprint generated from different combinations appears significantly diverse. This variation is expected to generate considerable differences in the identification performance of Joshi et al.[7] method.

6 Evaluation Metrics and Result Analysis

The MasterPrint vulnerability is a threat to an identification system. Hence, the investigation in this paper followed both closed-set and open-set identification set-up. The investigation involved two tests: an identification test[51] and a zero MasterPrint detection test. During the identification test, each template was compared with every other template from the dataset, and the similarity score for each comparison was computed. If the highest score corresponds to the actual subject sample, it was quoted as a correct detect and identify (CDI). In a false alarm (FA) scenario,

Table 2 Nomenclature for various combinations of thresholding and thinning approaches on different datasets. (TN_1 – KMM thinning algorithm, TN_2 – K3M thinning approach, TN_3 – Hilditch thinning algorithm, TN_4 – Stentiford thinning algorithm).

Thresholding approach	Thinning approach	CrossMatch Sample DB	FVC2002 DB1_A	FVC2002 DB2_A	NIST sd302b	NIST sd302d
Iterative optimal thresholding	TN_1	1_1_1	2_1_1	3_1_1	4_1_1	5_1_1
	TN_2	1_1_2	2_1_2	3_1_2	4_1_2	5_1_2
	TN_3	1_1_3	2_1_3	3_1_3	4_1_3	5_1_3
	TN_4	1_1_4	2_1_4	3_1_4	4_1_4	5_1_4
Otsu's method	TN_1	1_2_1	2_2_1	3_2_1	4_2_1	5_2_1
	TN_2	1_2_2	2_2_2	3_2_2	4_2_2	5_2_2
	TN_3	1_2_3	2_2_3	3_2_3	4_2_3	5_2_3
	TN_4	1_2_4	2_2_4	3_2_4	4_2_4	5_2_4
Niblack local thresholding	TN_1	1_3_1	2_3_1	3_3_1	4_3_1	5_3_1
	TN_2	1_3_2	2_3_2	3_3_2	4_3_2	5_3_2
	TN_3	1_3_3	2_3_3	3_3_3	4_3_3	5_3_3
	TN_4	1_3_4	2_3_4	3_3_4	4_3_4	5_3_4
Bernsen's local image thresholding	TN_1	1_4_1	2_4_1	3_4_1	4_4_1	5_4_1
	TN_2	1_4_2	2_4_2	3_4_2	4_4_2	5_4_2
	TN_3	1_4_3	2_4_3	3_4_3	4_4_3	5_4_3
	TN_4	1_4_4	2_4_4	3_4_4	4_4_4	5_4_4

the highest score belongs to some other subjects' templates. The system may reject a partial fingerprint due to no similarity with any stored templates. Suppose the system is enrolled with $\mathcal{P}$ partial fingerprints, and $\mathcal{C}$, $\mathcal{F}$, and $\mathcal{R}$ denotes the count of CDI, FA, and rejected partial fingerprints, respectively. Detect and identification rate (DIR), δ, FA rate (FAR), F, and rejection rate (RR), Υ, are computed as: $\delta = \frac{\mathcal{C}}{\mathcal{P}} \times 100$, $F = \frac{\mathcal{F}}{\mathcal{P}} \times 100$, and $\Upsilon = \frac{\mathcal{R}}{\mathcal{P}} \times 100$.

An identification system producing lowest MasterPrints at higher DIR and lower FAR would become ideal for practical use in the FBS. The identification test results on each dataset involve computing the DIR, FAR, RR, and the percentage of MasterPrints generated without setting a predefined threshold. A cumulative matching characteristic (CMC) curve shows the rank-k performance of an identification system, depicting the identification of the correct subject at different ranks.[52] The results from the identification test make up the data for the CMC curve. Suppose we have k subjects enrolled with a system. Ideally, the rank-k identification rate should be 100%. The best approach is expected to reach 100% performance at the earliest. Hence, the CMC plots presented here show the DIR performance till rank-10.

For each combination that produced MasterPrints in the identification test, a zero MasterPrint detection test was conducted. In this test, the system threshold was raised gradually until no MasterPrints are observed. Suppose, τ is the threshold at which no MasterPrints were observed. The DIR, δ_0, at τ is calculated using the formula for δ. A good approach should show marginal variation between δ and δ_0. Subsequently, δ_0 is divided into three intermediate thresholds to compute DIR and FAR at each of these thresholds. The DIR and FAR at δ, δ_0, and the three intermediate thresholds provides the data to plot the watchlist receiver operating characteristic (ROC) curve for each dataset. The curve occupying top-left region in the watchlist ROC plot is considered robust as it shows slight variation in DIR and significant reduction in FAR as the system threshold is increased to accept highly similar partial fingerprints.
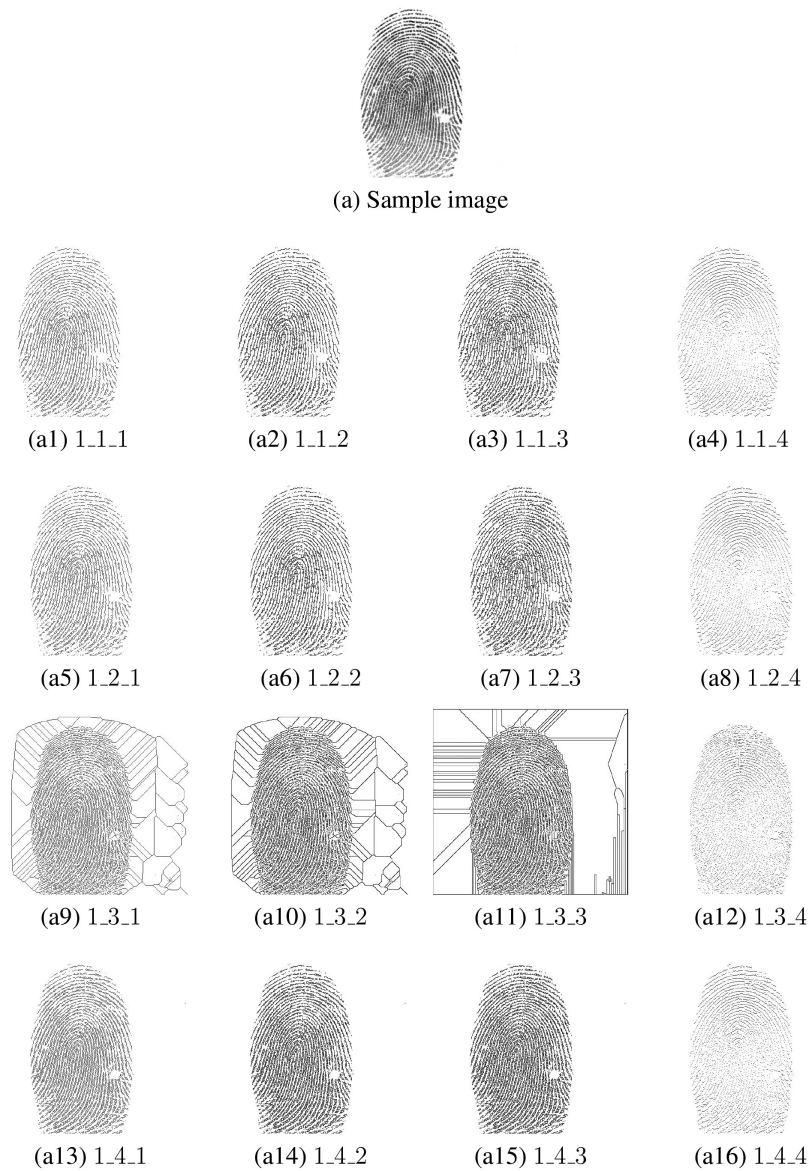
(a) Sample image

(a1) 1_1_1   (a2) 1_1_2   (a3) 1_1_3   (a4) 1_1_4

(a5) 1_2_1   (a6) 1_2_2   (a7) 1_2_3   (a8) 1_2_4

(a9) 1_3_1   (a10) 1_3_2   (a11) 1_3_3   (a12) 1_3_4

(a13) 1_4_1   (a14) 1_4_2   (a15) 1_4_3   (a16) 1_4_4

Fig. 14 A sample fingerprint image (a) from the CrossMatch Sample DB dataset and corresponding thinned images, (a1–a16), generated from various combinations of thresholding and thinning approaches.

6.1 Identification and Zero MasterPrint Detection Test Results

The partial fingerprint identification method with different combinations during the pre-processing stage was evaluated for σ, F, σ_0, Υ, and the percentage of MasterPrints generated. The results of each combination of thresholding and thinning approach for the identification and zero MasterPrint detection test on CrossMatch Sample DB dataset is presented in Table 3. The highest DIR observed here was 92.65% by 1_1_3 while producing nearly 10% MasterPrints. The combination 1_2_3 generated more than 21% MasterPrints. The results of each combination of thresholding and thinning approach for the identification and zero MasterPrint detection test on FVC2002 DB1_A dataset is given in Table 4. Only two combinations, namely, 2_4_2 and 2_4_3 could achieve more than 90% DIR. However, 2_4_2 generated the maximum percentage of MasterPrints during the experimentation. The results of each combination of thresholding and thinning approach for the identification and zero MasterPrint detection test on FVC2002 DB2_A dataset is given in Table 5. The combination 3_4_1 delivered more than 93% DIR but produced above 16% MasterPrints. The lowest percentage of MasterPrints during the experiments was

Table 3 Results on CrossMatch Sample DB dataset. σ_0 denotes DIR in zero MasterPrint generation test and MP represents the percentage of MasterPrints generated in identification test.

Binding approach	σ (%)	F (%)	σ_0 (%)	MP (%)	Υ (%)
1_1_1	84.8	6.6	62.21	4.8	8.6
1_1_2	77.9	7.7	61.8	7.94	14.4
1_1_3	92.65	3.8	75.65	9.65	3.55
1_1_4	90.12	3.86	78.4	9.12	6.02
1_2_1	80.39	8.26	62.32	8.39	11.35
1_2_2	81.15	5.34	60.84	15.15	13.51
1_2_3	91.52	5.7	73.29	21.12	2.78
1_2_4	91.89	2.45	75.62	12.89	5.66
1_3_1	79.95	7.95	60.4	9.95	12.1
1_3_2	70.29	9.41	55.21	6.29	20.3
1_3_3	84.31	6.34	62.87	8.31	9.35
1_3_4	80.88	4.46	65.48	8.89	14.66
1_4_1	87.25	4.55	60.98	17.25	8.2
1_4_2	68.8	7.12	55.43	18.87	24.08
1_4_3	89.69	2.61	72.64	9.69	7.7
1_4_4	92.16	3.06	74.7	9.16	4.78

Table 4 Results on FVC2002 DB1_A dataset. σ_0 denotes DIR in zero MasterPrint generation test and MP represents the percentage of MasterPrints generated in identification test.

Binding approach	σ (%)	F (%)	σ_0 (%)	MP (%)	Υ (%)
2_1_1	85.2	4.18	68.7	8.2	10.62
2_1_2	82.06	6.84	69.57	12.06	11.1
2_1_3	79.35	2.99	60.58	7.35	17.66
2_1_4	85.88	5.48	62.74	15.88	8.64
2_2_1	69.61	2.29	55.49	17.61	28.1
2_2_2	79.85	5.27	62.74	19.85	14.88
2_2_3	75.88	3.48	60.8	15.88	20.64
2_2_4	77.11	8.87	65.71	17.11	14.02
2_3_1	68.05	3.27	55.96	6.05	28.68
2_3_2	89.71	4.54	75.21	9.71	5.75
2_3_3	78.69	8.93	65.84	5.69	12.38
2_3_4	89.12	7.13	62.09	17.12	3.75
2_4_1	82.75	6.85	70.75	12.75	10.4
2_4_2	91.13	6.14	73.25	21.13	2.73
2_4_3	90.31	4.54	78.09	12.37	5.15
2_4_4	79.84	4.16	64.86	7.84	16

Table 5 Results on FVC2002 DB2_A dataset. σ_0 denotes DIR in zero MasterPrint generation test and MP represents the percentage of MasterPrints generated in identification test.

Binding approach	σ (%)	F (%)	σ_0 (%)	MP (%)	Υ (%)
3_1_1	81.26	4.06	75.21	9.26	14.68
3_1_2	81.8	7.41	78.95	10.31	10.79
3_1_3	85.29	7.21	74	8.29	7.5
3_1_4	85.05	7.28	62.54	8.05	7.67
3_2_1	88.53	2.73	60.27	11.53	8.74
3_2_2	67.33	5.22	51.06	12.5	27.45
3_2_3	85.05	7.29	71.98	15.05	7.66
3_2_4	84.31	6.53	62.7	14.31	9.16
3_3_1	92.16	3.2	80.29	12.16	4.64
3_3_2	91.79	5.54	76.32	17.79	2.67
3_3_3	70.51	7.18	62.98	19.51	22.31
3_3_4	79.26	8.95	59.21	9.26	11.79
3_4_1	93.32	3.94	80.9	16.32	2.75
3_4_2	79.51	7.19	60.77	19.51	13.3
3_4_3	91.67	4.81	74	11.67	3.52
3_4_4	89.3	6.03	65.74	9.3	4.67

around 8% by the combination 3_1_4. The results of each combination of thresholding and thinning approach for the identification and zero MasterPrint detection test on NIST sd302b dataset is given in Table 6. Here, three combinations, namely, 4_1_1, 4_2_1, and 4_4_2 achieved above 90% DIR. However, 4_4_2 generated above 18% MasterPrints and so was ineffective toward mitigating the MasterPrint vulnerability. The results of each combination of thresholding and thinning approach for the identification and zero MasterPrint detection test on NIST sd302d dataset is given in Table 7. The highest DIR observed here was 92.38% by 5_1_2 while producing 16.81% MasterPrints. The combination 5_4_4 generated more than 23% MasterPrints.

The entries from Tables 3–7 demonstrated that the DIR ranges between 62%–93%. But the average DIR in the Joshi et al.[7] approach was 93.8%. Thus, the DIR performance of the Joshi et al.[7] method has been reduced by more than 11% on average due to varying combinations of thresholding and thinning methods. Moreover, the DIR in the zero MasterPrint detection test was also lowered by nearly 10% during the investigation. However, the average FAR has decreased by only 0.6% compared with Joshi et al.[7] work. The average RR also reduced by 7.16%. The percentage of MasterPrint generated during the investigation ranges between 3.82%–23.45%, whereas for the Joshi et al.[7] method, the range lies within 0.1%–2.03%. The analysis on the percentage of MasterPrints generated showed that the investigation witnessed a 14 times increase in the MasterPrints, whereas the average percentage of MasterPrints generated for the experiments was more than 11% compared to the original paper. These statistics thus demonstrated that the accuracy and MasterPrint mitigation performance of the Joshi et al.[7] method had notably reduced when utilizing different pre-processing schemes. The entries for σ, F, σ_0, Υ, and the percentage of MasterPrints generated during the investigations ranges between 62.01% and 93.32%, 1.83% and 9.41%, 50.2% and 80.9%, 1.43% and 36.16%, and 3.82% and 23.45%, respectively. Thus, the results from Tables 3–7 for identification and zero MasterPrint detection test appear diversely distributed for each of the five parameters under consideration. Therefore, no binding approach have shown remarkable variations invariably on several datasets. Hence, these performance measures do not form a concrete base to attribute any pair, specific thresholding or thinning method as preferable over others for fingerprint recognition.

Table 6 Results on NIST SD sd302b dataset. σ_0 denotes DIR in zero MasterPrint generation test and MP represents the percentage of MasterPrints generated in identification test.

Binding approach	σ (%)	F (%)	σ_0 (%)	MP (%)	Υ (%)
4_1_1	92.85	3.37	74.65	4.85	3.78
4_1_2	85.75	3.65	60.9	9.75	10.6
4_1_3	65.2	4.07	52.49	15.2	30.73
4_1_4	62.01	2.94	49.65	6.01	35.05
4_2_1	91.1	5.11	79.58	9.1	3.79
4_2_2	81.75	6.61	67.25	13.75	11.64
4_2_3	62.01	1.83	50.2	12.01	36.16
4_2_4	62.25	3.08	52.36	12.25	34.67
4_3_1	74.57	7.41	56.8	11.57	18.02
4_3_2	74.02	8.53	65.21	22.02	17.45
4_3_3	84.8	6.88	72.95	8.8	8.32
4_3_4	88.97	6.33	76.32	18.97	4.7
4_4_1	81.52	8.97	74.68	15.53	9.51
4_4_2	90.28	6.29	78.98	18.28	3.43
4_4_3	78.19	6.83	62.4	17.19	14.98
4_4_4	83.33	6.26	60.35	13.39	10.41

Table 7 Results on NIST SD302d dataset. σ_0 denotes DIR in zero MasterPrint generation test and MP represents the percentage of MasterPrints generated in identification test.

| Binding approach	σ (%)	F (%)	σ_0 (%)	MP (%)	Υ (%)
5_1_1	82.17	6.07	70.25	12.11	11.76
5_1_2	92.38	6.19	79.28	16.81	1.43
5_1_3	73.82	2.12	62.47	3.82	24.06
5_1_4	67.91	2.31	51.64	7.96	29.78
5_2_1	82.35	7.3	69.4	8.35	10.35
5_2_2	83.06	8.21	67.24	9.06	8.73
5_2_3	67.01	2.32	50.85	7.01	30.67
5_2_4	88.58	6.54	69.4	8.97	4.88
5_3_1	85.78	7.2	60.36	15.78	7.02
5_3_2	87.09	5.45	65.32	17.05	7.46
5_3_3	83.63	7.02	59.21	13.63	9.35
5_3_4	71.96	3.55	57.39	11.96	24.49
5_4_1	72.55	8.37	60.9	17.55	19.08
5_4_2	91.1	5.22	78.35	15.12	3.68
5_4_3	84.17	5.67	68.21	14.85	10.16
5_4_4	73.43	8.09	64.85	23.45	18.48
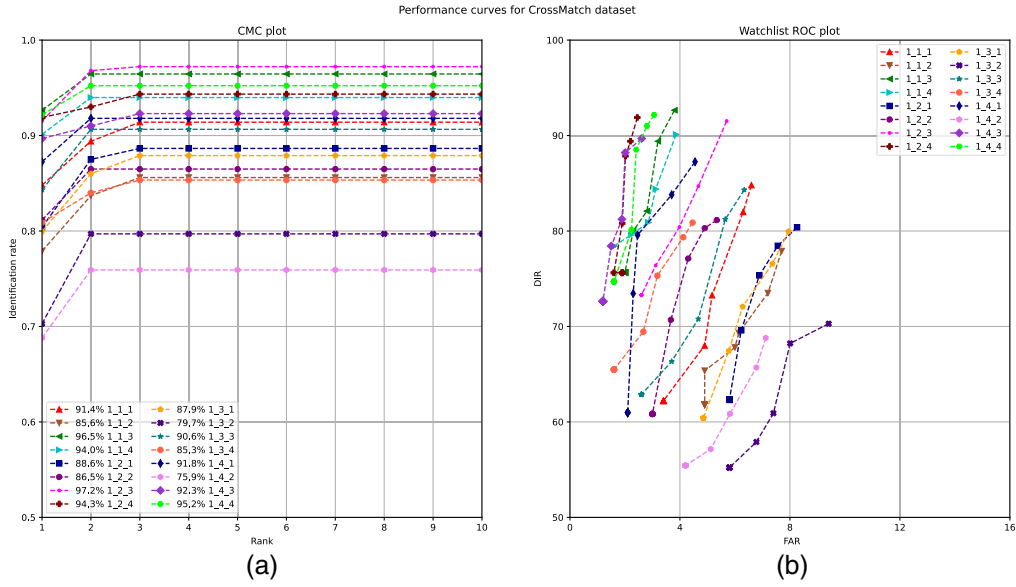
Fig. 15 CMC plot (a) and Watchlist ROC plot (b) for CrossMatch Sample DB dataset.

6.2 *CMC and Watchlist ROC Curve Performance*

The CMC and Watchlist ROC curves for each pre-processing combination on an individual dataset are depicted on the left and right portion of Figs. 15–19, respectively. The highest and average rank-10 DIR achieved during the investigation was 98.6% and 87.35%, respectively. The plots also demonstrated that the DIR did not improve beyond rank-3. However, Joshi et al.[7] method achieved 100% DIR on each dataset till rank-2. In the case of Watchlist ROC plots, ideally, the curves are expected to deviate marginally for DIR and show significant variations on FAR. But the plots from Figs. 15–19 demonstrated that the identification rate reduced considerably compared to FAR. Moreover, the average DIR in zero MasterPrint generation test, σ_0, has reduced from 76.14% in the Joshi et al.[7] work to 65.96% during the investigations. Thus, delivered a degraded accuracy of more than 10%. The plots thus showed that the average identification accuracy was reduced by more than 12% when diverse combinations of pre-processing methods were employed in Joshi et al.[7] work. However, it is still infeasible to label any single thresholding
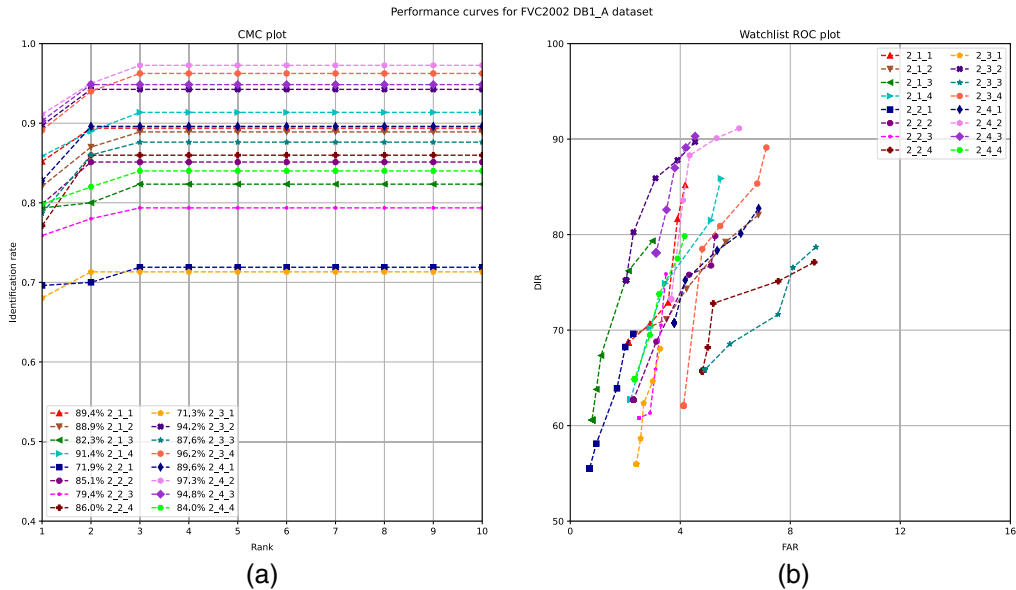

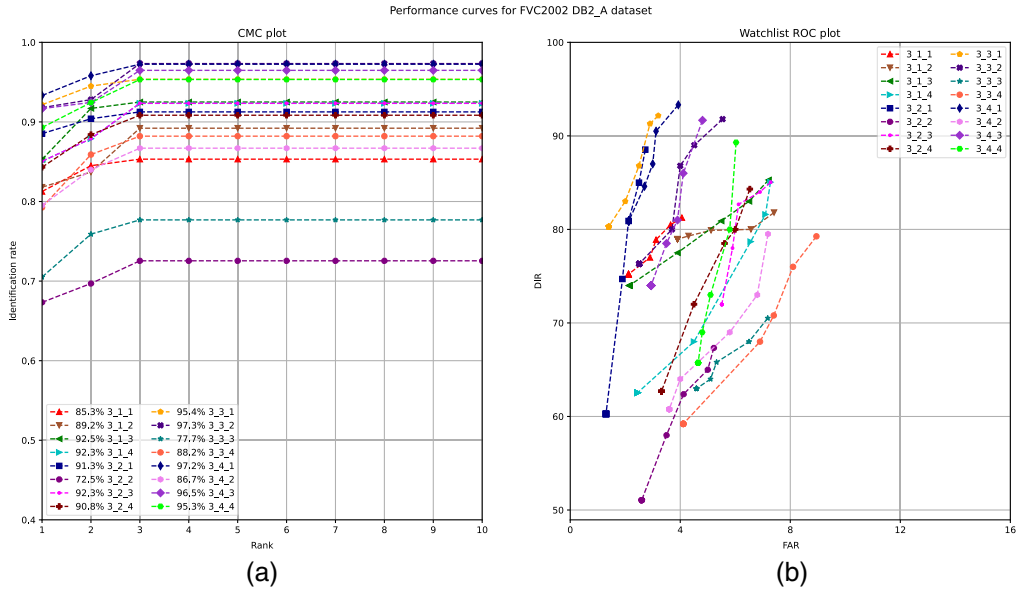**Fig. 16** CMC plot (a) and Watchlist ROC plot (b) for FVC2002 DB1_A dataset.

Fig. 17 CMC plot (a) and Watchlist ROC plot (b) for FVC2002 DB2_A dataset.
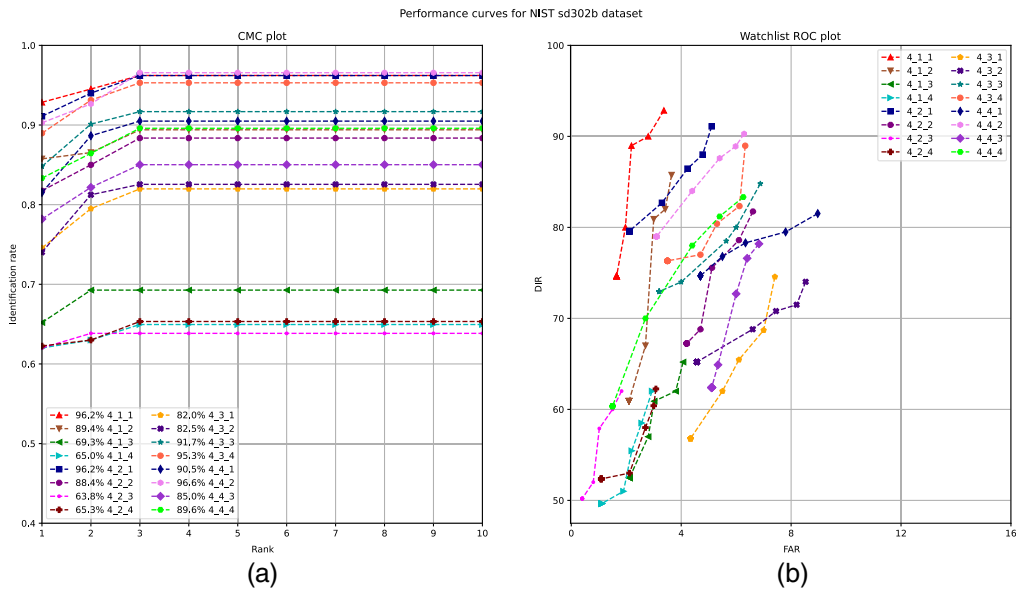

Fig. 18 CMC plot (a) and Watchlist ROC plot (b) for NIST sd302b dataset.

or thinning method, or their specific combination as more suitable for fingerprint recognition due to their inconsistent performance in CMC and Watchlist ROC plots.

7 Discussion

The investigation illustrated that the partial fingerprint identification and MasterPrint mitigation method presented by Joshi et al.[7] delivered low performance on crucial parameters when a variety of thresholding and thinning methods were employed in place of the adaptive thresholding approach[8] and Zhang–Suen thinning algorithm[9] employed in Joshi et al.[7] work. The combination of Bernsen's local image thresholding and K3M thinning algorithm produced above 90% DIR on three datasets, namely, DB1_A, NIST sd302b, and sd302d. But the approach generated more than 15% MasterPrints during the same experiment. Another binding of Bernsen's local image
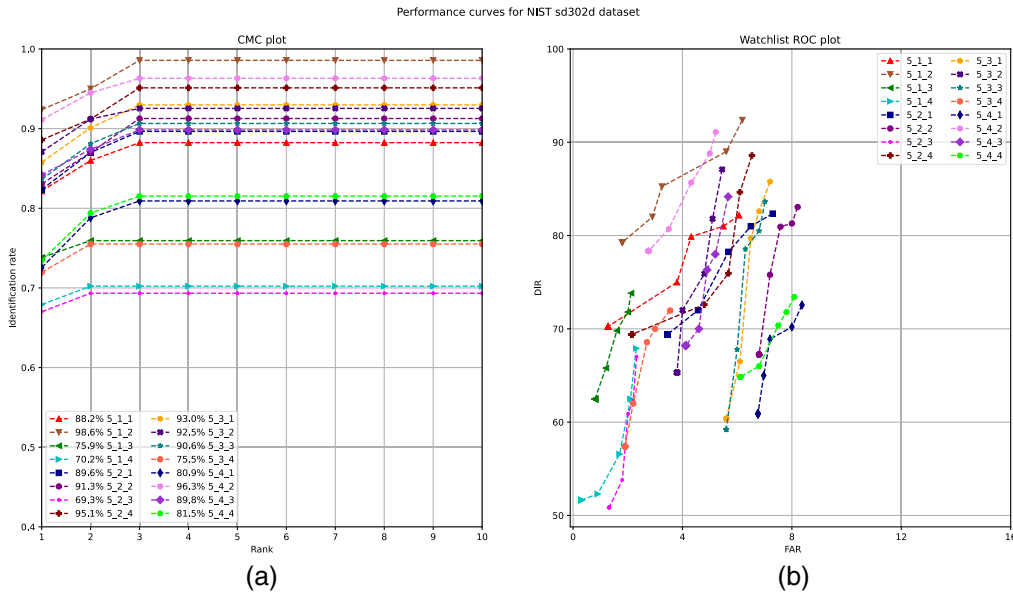
Fig. 19 CMC plot (a) and Watchlist ROC plot (b) for NIST sd302d dataset.

thresholding and Hilditch thinning algorithm delivered greater than 90% DIR on FVC datasets while generating more than 11% MasterPrints. Thus, the DIR and the percentage of MasterPrint generated by various pre-processing combinations on five benchmark datasets showed that no pairing has consistently performed better over the other methods. Moreover, the entries from Tables 3–7 also confirmed that it is infeasible to suggest specific thresholding or thinning method as appropriate for partial fingerprint identification when experimented with given datasets. Precise thresholding and thinning of fingerprint ridges are the driving factors for accurate minutiae detection. The literature on image processing includes several approaches for thresholding and thinning grayscale images. All the thinning and thresholding approaches from the literature are incompatible for high-security applications like anonymous user identification through a fingerprint. Conclusively, the results confirmed that high-security applications and user identification systems employing biometric traits of an individual are greatly influenced by the choices made in the pre-processing stage.

Jabeen and Khan[53] proposed a hybrid algorithm for false minutiae and boundary elimination. The algorithm removes false minutiae from a thinned fingerprint image arising due to bridges, spikes, and ridge breaks. Xiao and Raafat[54] presented a false minutiae detection and elimination method. The authors represented false minutiae using structural and statistical approaches. Kim et al.[50] method for false minutiae removal employed in the experimentation, and other similar approaches utilise thinned images for false minutiae detection and elimination. Hence, an inappropriate binarized and thinned fingerprint image is the agent behind the improper functioning of any false minutiae removal scheme.

Although there is no lower bound on the count of minutiae matched to decide if the fingerprints are matched, some legal procedures accept at least an 8 to 17 minutiae match for evidence.[55] An average minutiae density on a 500 dpi fingerprint is estimated to be 0.246 minutiae/mm^2.[56] Such statistics should be employed to confirm that accurate minutiae detection and matching was carried out by a novel method proposed in future involving thresholding and thinning methods in the pre-processing phase. A robust novel fingerprint identification method should be experimented with fingerprint datasets employing diverse sensor types. A fingerprint identification system can generate MasterPrints if the fingerprint pre-processing approaches are not precisely tested on several datasets using different sensors.

The first method addressing the MasterPrint vulnerability for partial fingerprint identification was published by Joshi et al.[7] A thorough search of the relevant literature on thresholding and thinning of fingerprint images yielded no prior work that experimented with diverse combinations in the pre-processing stage to study the impact on the identification accuracy, and the

percentage of MasterPrint generated using a partial fingerprint identification method. The work in this paper is a maiden attempt to demonstrate that every image pre-processing approach might not be suitable for a high-security person identification using partial fingerprints. Thus the work initiates the requirement of image pre-processing methods extensively tested on fingerprint datasets from diverse sensors.

As future scope of this work, we will study the feasibility of robust partial fingerprint identification and MasterPrint mitigation method for poor quality latent fingerprints and utilize multiple pre-processing schemes to prove its practicability. As experimental results showed that pre-processing methods highly affect an identification system's accuracy, the initial goal will be to experiment with different pre-processing and false minutiae removal approaches on poor quality partial and latent fingerprint datasets acquired using dissimilar sensor types.

8 Conclusion

The MasterPrint vulnerability makes a partial fingerprint identification system susceptible to presentation attacks. This paper presented an investigation using sixteen combinations of thresholding and thinning methods on a partial fingerprint identification system to study their impact on identification accuracy and the percentage of MasterPrints generated. The results demonstrated that the existing partial fingerprint identification and MasterPrint mitigation approach performed unsatisfactorily when diverse combinations of thresholding and thinning methods were employed during the pre-processing stage. The investigation showed that existing thresholding and thinning methods are incompatible for minutiae-based fingerprint identification systems. The results also confirmed that average minutiae density is a preferable measure compared to the average minutiae count for feature extraction from the fingerprint image. Furthermore, the investigations demonstrated that the system performance can be barely improved by using false minutiae removal methods if the underlying thresholding and thinning approaches does not produce thinned images that retains the ridge patterns and ridge connectivity from the original grayscale fingerprint images. Thus, the work in this paper instigates the requirement of rigorously tested pre-processing methods suitable for fingerprint images.

References

1. A. Ross, K. Nandakumar, and A. K. Jain, *Handbook of Multibiometrics*, The Kluwer International Series on Biometrics, Vol. **6**, Springer (2006).
2. A. Farina, Z. M. Kovács-Vajna, and A. Leone, "Fingerprint minutiae extraction from skeletonized binary images," *Pattern Recognit.* **32**(5), 877–889 (1999).
3. N. K. Ratha, S. Chen, and A. K. Jain, "Adaptive flow orientation-based feature extraction in fingerprint images," *Pattern Recognit.* **28**(11), 1657–1672 (1995).
4. T. Sakai, M. Nagao, and H. Matsushima, "Extraction of invariant picture sub-structures by computer," *Comput. Graph. Image Process.* **1**(1), 81–96 (1972).
5. J. Feng, "Combining minutiae descriptors for fingerprint matching," *Pattern Recognit.* **41**(1), 342–352 (2008).
6. Z. Guo and R. W. Hall, "Parallel thinning with two-subiteration algorithms," *Commun. ACM* **32**(3), 359–373 (1989).
7. M. Joshi, B. Mazumdar, and S. Dey, "Mitigating MasterPrint vulnerability by employing minutiae geometry," *J. Electron. Imaging* **31**(1), 013026 (2022).
8. D. Bradley and G. Roth, "Adaptive thresholding using the integral image," *J. Graph. Tools* **12**, 13–21 (2007).
9. T. Y. Zhang and C. Y. Suen, "A fast parallel algorithm for thinning digital patterns," *Commun. ACM* **27**, 236–239 (1984).
10. A. Roy, N. D. Memon, and A. Ross, "Masterprint: exploring the vulnerability of partial fingerprint-based authentication systems," *IEEE Trans. Inf. Forensics Security* **12**(9), 2013–2025 (2017).
11. P. Bontrager et al., "DeepMasterPrints: generating MasterPrints for dictionary attacks via latent variable evolution," in *IEEE 9th Int. Conf. Biometrics Theor., Appl. and Syst. (BTAS)*, IEEE (2018).

12. C. I. Watson, "NIST special database 9, mated fingerprint card pairs," (1993).
13. X. Jia et al., "A cross-device matching fingerprint database from multi-type sensors," in *21st Int. Conf. Pattern Recognit. (ICPR)*, pp. 3001–3004 (2012).
14. K. Arora and D. Garg, "A quantitative survey of various fingerprint enhancement techniques," *Int. J. Comput. Appl.* **28**, 24–29 (2011).
15. A. A. Abbood, G. Sulong, and S. U. Peters, "A review of fingerprint image pre-processing," *Jurnal Teknologi* **69**(2), 79–84 (2014).
16. X. Wang and L. Chen, "An effective histogram modification scheme for image contrast enhancement," *Signal Process. Image Commun.* **58**, 187–198 (2017).
17. Y. Qi et al., "A comprehensive overview of image enhancement techniques," *Arch. Comput. Methods Eng.* **29**, 583–607 (2021).
18. N. Longkumer, M. Kumar, and R. Saxena, "Contrast enhancement techniques using histogram equalization: a survey," *Int. J. Curr. Eng. Technol.* **4**, 1561–1565 (2014).
19. W. A. Mustafa and M. M. M. A. Kader, "A review of histogram equalization techniques in image enhancement application," *J. Phys. Conf. Ser.* **1019**, 012026 (2018).
20. D. Maltoni et al., *Handbook of Fingerprint Recognition*, 2nd ed., Springer (2009).
21. A. Shetter, S. Prajwalasimha, and H. Swapna, "Finger print image enhancement using thresholding and binarization techniques," in *Second Int. Conf. Inventive Commun. and Comput. Technol. (ICICCT)*, IEEE (2018).
22. D. Maio and D. Maltoni, "Direct gray-scale minutiae detection in fingerprints," *IEEE Trans. Pattern Anal. Mach. Intell.* **19**(1), 27–40 (1997).
23. S. A. Sudiro, M. Paindavoine, and T. M. Kusuma, "Simple fingerprint minutiae extraction algorithm using crossing number on valley structure," in *IEEE Workshop on Autom. Identification Adv. Technol.*, IEEE (2007).
24. M. Sonka, V. Hlaváč, and R. Boyle, *Image Processing, Analysis and Machine Vision*, 3rd ed., Thomson (2008).
25. P. Stathis, E. Kavallieratou, and N. Papamarkos, "An evaluation technique for binarization algorithms," *J. Univers. Comput. Sci.* **14**(18), 3011–3030 (2008).
26. S. H. Shaikh, K. Saeed, and N. Chaki, "Performance benchmarking of different binarization techniques for fingerprint-based biometric authentication," in *Proc. 8th Int. Conf. Comput. Recognit. Syst. CORES 2013*, Milkow, Poland, 27-29 May 2013, R. Burduk et al., Eds., Advances in Intelligent Systems and Computing, Vol. 226, 237–246, Springer (2013).
27. N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst. Man Cybern.* **9**(1), 62–66 (1979).
28. W. Niblack, *An Introduction to Digital Image Processing*, Prentice Hall International (1988).
29. J. Bernsen, "Dynamic thresholding of grey-level images," in *ICPR'86: Proc. Int. Conf. Pattern Recognit.*, Paris, France, pp. 1251–1255 (1986).
30. K. Buch et al., "Quantitative variations in texture analysis features dependent on mri scanning parameters: a phantom model," *J. Appl. Clin. Med. Phys.* **19**(6), 253–264 (2018).
31. N. B. Rais, M. S. Hanif, and I. A. Taj, "Adaptive thresholding technique for document image analysis," in *Proc. INMIC 8th Int. Multitopic Conf.*, pp. 61–66 (2004).
32. K. Khurshid et al., "Comparison of niblack inspired binarization methods for ancient documents," *Proc. SPIE* **7247**, 72470U (2009).
33. L. P. Saxena, "Niblack's binarization method and its modifications to real-time applications: a review," *Artif. Intell. Rev.* **51**, 673–705 (2017).
34. N. Senthilkumaran and S. Vaithegi, "Image segmentation by using thresholding techniques for medical images," *Comput. Sci. Eng. Int. J.* **6**, 1–13 (2016).
35. C. Eyupoglu, "Implementation of bernsen's locally adaptive binarization method for gray scale images," *Online J. Sci. Technol.* **7**, 68–72 (2017).
36. Q. Li, X. Bai, and W. Liu, "Skeletonization of gray-scale image from incomplete boundaries," in *15th IEEE Int. Conf. Image Process.*, IEEE (2008).
37. M. Nazarkevych et al., "Evaluation of the effectiveness of different image skeletonization methods in biometric security systems," *Int. J. Sens. Wireless Commun. Control* **11**, 542–552 (2021).
38. C. J. Hilditch, "Linear skeletons from square cupboards," in *Machine Intelligence 4*, B. Meltzer and D. Michie, Eds., p. 403, Edinburgh University Press (1969).

39. P. K. Saha, G. Borgefors, and G. S. di Baja, "A survey on skeletonization algorithms and their applications," *Pattern Recognit. Lett.* **76**, 3–12 (2016).

40. F. W. M. Stentiford and R. G. Mortimer, "Some new heuristics for thinning binary hand-printed characters for OCR," *IEEE Trans. Syst. Man Cybern.* **SMC-13**(1), 81–84 (1983).

41. K. Saeed et al., "K3M: a universal algorithm for image skeletonization and a review of thinning techniques," *Int. J. Appl. Math. Comput. Sci.* **20**(2), 317–335 (2010).

42. M. Tabedzki, K. Saeed, and A. Szczepanski, "A modified K3M thinning algorithm," *Int. J. Appl. Math. Comput. Sci.* **26**(2), 439–450 (2016).

43. K. Saeed, M. Rybnik, and M. Tabedzki, "Implementation and advanced results on the non-interrupted skeletonization algorithm," *Proc. SPIE* **2124**, 601–609 (2001).

44. K. Saeed, "Text and image processing: non-interrupted skeletonization," in *Proc. 1st Int. IEEE Conf. Circuits, Syst., Commun. and Comput.—IEEE-CSCC'01*, Society Press, pp. 350–354 (2001).

45. J. Yu and Y. Li, "Improving hilditch thinning algorithms for text image," in *Proc. Int. Conf. E-Learning, E-Business, Enterprise Inf. Syst., and E-Government, EEEE '09*, IEEE Computer Society, USA, pp. 76–79 (2009).

46. P. Patil, S. Suralkar, and F. Sheikh, "Rotation invariant thinning algorithm to detect ridge bifurcations for fingerprint identification," in *17th IEEE Int. Conf. Tools with Artif. Intell. (ICTAI05)*, IEEE (2005).

47. S. Yokoi, J. Toriwaki, and T. Fukumura, "Topological properties in digitized binary pictures," *Syst. Comput. Controls* **4**, 32–39 (1973).

48. P. Subashini, "Optimal thinning algorithm for detection of FCD in MRI images," *Int. J. Sci. Eng. Res.* **2**(9), 1–7 (2011).

49. G. Limei, Z. Yingbin, and H. Duan, "A fingerprint minutiae extraction method in quantum thinned binary image," *Int. J. Theor. Phys.* **60**, 1883–1894 (2021).

50. S. Kim, D. Lee, and J. Kim, "Algorithm for detection and elimination of false minutiae in fingerprint images," *Lect. Notes Comput. Sci.* **2091**, 235–240 (2001).

51. D. Blackburn et al., "Biometric testing and statistics," Tech. Rep., National Science and Technology Council (NSTC), Washington, DC (2006).

52. B. DeCann and A. Ross, "Relating ROC and CMC curves via the biometric menagerie," in *IEEE Sixth Int. Conf. Biometrics: Theor., Appl. and Syst., BTAS 2013*, Arlington, VA, USA, September 29- October 2, IEEE, pp. 1–8 (2013).

53. S. Jabeen and S. A. Khan, "A hybrid false minutiae removal algorithm with boundary elimination," in *IEEE Int. Conf. Syst. of Syst. Eng.*, IEEE (2008).

54. Q. Xiao and H. Raafat, "Fingerprint image postprocessing: a combined statistical and structural approach," *Pattern Recognit.* **24**, 985–992 (1991).

55. S. Z. Li and A. K. Jain, Eds., *Encyclopedia of Biometrics*, 2nd ed., Springer US (2015).

56. S. Pankanti, S. Prabhakar, and A. K. Jain, "On the individuality of fingerprints," *IEEE Trans. Pattern Anal. Mach. Intell.* **24**(8), 1010–1025 (2002).

Mahesh Joshi is currently a research scholar in the Department of Computer Science and Engineering at Indian Institute of Technology Indore, India. He received his MTech degree in Computer Science and Engineering from the Visvesvaraya National Institute of Technology, Nagpur, India. His current research interests include fingerprint biometrics and pattern recognition. He has six publications from refereed journals, book chapters, and international conference to his credit.

Bodhisatwa Mazumdar is currently working as an assistant professor in the Department of Computer Science and Engineering at Indian Institute of Technology Indore, India. He received his BTech degree from the University of Kalyani and MS degree in electronics and electrical communication engineering from IIT Kharagpur, India. He received his PhD in computer science and Engineering from IIT Kharagpur. His research area includes power based side-channel analysis of cryptographic primitives such as S-boxes, and security vulnerability analysis of emerging technologies in VLSI Design. He received the best student paper award in the 25th IEEE International Conference on VLSI Design 2012, Hyderabad, India. He was a post-doctoral

associate in the Design for Excellence Laboratory at New York University Abu Dhabi, Abu Dhabi. He has published over 50 research articles (including papers in international journals, conferences).

Somnath Dey is currently working as an associate professor in the Department of Computer Science and Engineering at Indian Institute of Technology Indore, India. He received his BTech degree in information technology from the University of Kalyani in 2004. He completed his MS (by research) and PhD degrees in information technology from the School of Information Technology, Indian Institute of Technology Kharagpur, in 2008 and 2013, respectively. His research interest includes biometric security, biometric template protection, biometric crypto system. He has published over 40 research articles (including papers in international journals, conferences, and book chapters).