

Research Article

A Digital Camera-Based Rotation-Invariant Fingerprint Verification Method

Sajid Khan,¹ Dong-Ho Lee ,² Asif Khan,³ Ahmad Waqas,⁴ Abdul Rehman Gilal,⁴ and Zahid Hussain Khand⁴

¹Center of Excellence for Robotics, Artificial Intelligence and Blockchain, Sukkur IBA University, Sukkur, Pakistan

²School of Electrical Engineering, Hanyang University ERICA Campus, Ansan, Republic of Korea

³Department of Electrical Engineering, Sukkur IBA University, Sukkur, Pakistan

⁴Department of Computer Science, Sukkur IBA University, Sukkur, Pakistan

Correspondence should be addressed to Dong-Ho Lee; dhlee77@hanyang.ac.kr

Received 3 January 2020; Accepted 29 April 2020; Published 15 May 2020

Academic Editor: Asad M. Khatak

Copyright © 2020 Sajid Khan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Fingerprint registration and verification is an active area of research in the field of image processing. Usually, fingerprints are obtained from sensors; however, there is recent interest in using images of fingers obtained from digital cameras instead of scanners. An unaddressed issue in the processing of fingerprints extracted from digital images is the angle of the finger during image capture. To match a fingerprint with 100% accuracy, the angles of the matching features should be similar. This paper proposes a rotation and scale-invariant decision-making method for the intelligent registration and recognition of fingerprints. A digital image of a finger is taken as the input and compared with a reference image for derotation. Derotation is performed by applying binary segmentation on both images, followed by the application of speeded up robust feature (SURF) extraction and then feature matching. Potential inliers are extracted from matched features by applying the M-estimator. Matched inlier points are used to form a homography matrix, the difference in the rotation angles of the finger in both the input and reference images is calculated, and finally, derotation is performed. Input fingerprint features are extracted and compared or stored based on the decision support system required for the situation.

1. Introduction

Image processing has many applications [1–3], including biometric registration and verification. Biometric registration and verification are performed to restrict unauthorized people from entering a specific building or wing, to keep track of users of service by registering their fingerprints in a database, to take automatic attendance of employees, and to store data along with biometric features in a database for issuing passports or national identity cards.

Biometric verification and registration using image processing techniques such as face detection, fingerprint detection, and retina or iris detection have been widely researched [4–8]. Face detection-based techniques [9, 10] store facial features to be compared during the registration and verification process. Face verification and registration

provides accuracy at the cost of high complexity and extensive allocation of system memory. Accuracy demands that issues such as scale difference, posture, illumination, and the opening and closing of the eyes and mouth be addressed, further increasing complexity.

Iris-based recognition and verification [7, 11, 12], on the other hand, is done by applying segmentation in the region of the iris to extract its radius. This is followed by feature extraction. These techniques are used along with fingerprint verification at airports and other high-risk buildings to provide accurate identification. Iris-based recognition and verification provide good accuracy, but the required equipment is costly.

Fingerprint verification and registration is a widely researched area [13–23] in which work has focused on algorithms and techniques to store and compare fingerprint

features obtained using a scanner. Fingerprint-based biometric verification and registration are widely used in banks [19, 20] and national databases for maintaining passport and national identity card information [24].

By using different approaches, such as combining fingerprint registration with odor analysis [21] and applying a ridgelet transform [22], researchers have improved existing algorithms. Complexity can be reduced by comparing a small number of features instead of entire images. However, these methods require the use of sensors (fingerprint readers) that are not always available. This can make performing the registration and verification process difficult.

Solutions that avoid the need of sensors have been proposed [24–27]. These techniques focus on processing images captured using ordinary digital cameras. Hiew et al. [24–26] applied skin color-based segmentation followed by morphological processing to extract regions of interest. Short-time Fourier transform analysis (STFT) is then applied, followed by ridge orientation calculation and core point detection. The detection and matching of features such as bifurcations and ridges are not discussed. However, detection and matching can be performed using the same fingerprint techniques used in the algorithms applied to input obtained from sensors.

Sankaran et al. [27] proposed a method similar to that discussed in this paper that performs skin color segmentation followed by the region of interest (ROI) extraction. Image enhancement is applied followed by ScatNet-based feature extraction. Finally, feature-based fingerprint registration/matching is performed.

When features are extracted from a digital image of a fingerprint, scanners are not needed. However, orientation alignment is an issue that is inadequately addressed in the algorithms proposed by Hiew et al. and Sankaran et al. The fingerprint in the input image must be vertically aligned when conventional methods are used. When rotation of the finger is present, a fingerprint mismatch will occur even if features of that person are present in the database.

Therefore, this paper proposes a rotation and scale-invariant technique that takes an image captured using an ordinary digital camera as the input and applies skin color-based segmentation to the input image and a reference image to extract regions of interest from both images. SURF [28] algorithm is applied to the segmented binary images, and the detected features from both images are matched. Inliers are extracted from the matched features using the M-estimator [29] for accurate calculation of the rotation angle using a homography matrix [30]. The input image is derotated to obtain an image with a vertically aligned finger. The vertically aligned input image is then subjected to image enhancement techniques, such as unsharp masking and contextual blockwise Fourier domain filtering. Otsu thresholding is applied for binarization. Then, nonmaximal suppression is used to obtain ridges having a thickness of one pixel. Features such as bifurcations and ridge endings are extracted from the thin ridges to be stored in a database for registration or compared with the features of registered fingerprints for verification. Simulation results show that the proposed method provides an accuracy of 100 percent if the suggested parameter range is applied.

This paper is organized as follows: Section 2 discusses the proposed methodology, Section 3 contains the results and discussion, Section 4 presents the conclusions and the direction of future work.

2. Methodology

The objective of this work is to propose a cost-effective rotation and scale-invariant alternative to the processing of fingerprint images obtained using scanners. A flowchart of the algorithm is shown in Figure 1. The major steps of the proposed algorithm are the derotation of the input image, image enhancement, binarization, thinning, minutiae extraction, and matching. The steps shown in the flowchart are explained in detail in this section.

2.1. Image Acquisition. In this study, image acquisition was performed using several smartphones: Samsung Galaxy S2, iPhone 3s, Galaxy S6, and iPhone 8. Images of fingers were captured with the focus on their ridges.

2.2. Image Derotation. A flowchart depicting the derotation process is shown in Figure 2. An image containing a finger aligned along the y -axis is selected as a reference image, as seen in Figure 3(a). Any input finger image requiring registration or verification is considered an input image and is subjected to the derotation process so that the fingerprint registration/matching process can be done smoothly. Figure 3(b) shows an example of an input image that requires rotational alignment for accurate feature matching.

2.2.1. Skin Color-Based Segmentation. Both the input and reference images are subjected to adaptive skin color-based image segmentation [31]. The images shown in Figures 3(c) and 3(d) show the results of this segmentation on the reference and input images. The input image shown in Figure 3(b) is particularly complex because the color of the background is similar to the color of the skin. Therefore, noise is detected as part of the finger. The white region in Figure 3(d) is the region of interest (ROI) and will be used in upcoming sections.

2.2.2. Input Image Scale Adjustment. Both input and reference images usually are of different dimensions. Therefore, the fingers in both images might have a huge scale difference. In that case, scale variance might produce problems while detecting and comparing features. In order to provide a scale-invariant derotation, scales of both input and reference fingers should nearly be the same. If the number of pixels belongs to the finger in the segmented reference image, it is denoted by N_{ref} and if the number of pixels belongs to ROI, it is denoted by N_{input} . Scaling factor is calculated using (1) as

$$S_{\text{ref}}^{\text{input}} = \frac{N_{\text{ref}}}{N_{\text{input}}}. \quad (1)$$

If $S_{\text{ref}}^{\text{input}} < 0.9$, both the intensity and segmented input images are scaled down by a factor of $S_{\text{ref}}^{\text{input}}$, whereas if

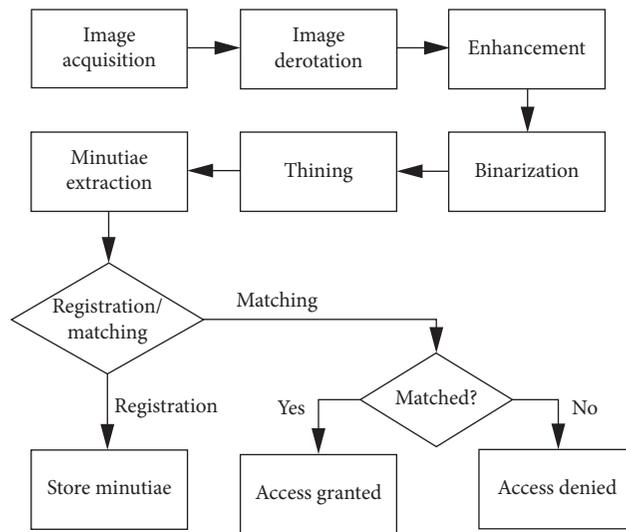


FIGURE 1: Flowchart of the proposed algorithm.

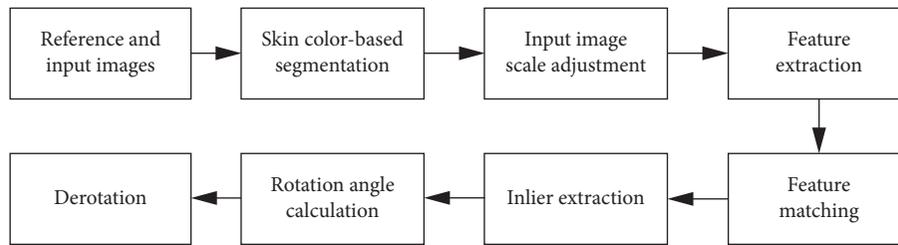


FIGURE 2: Flowchart of the derotation process.

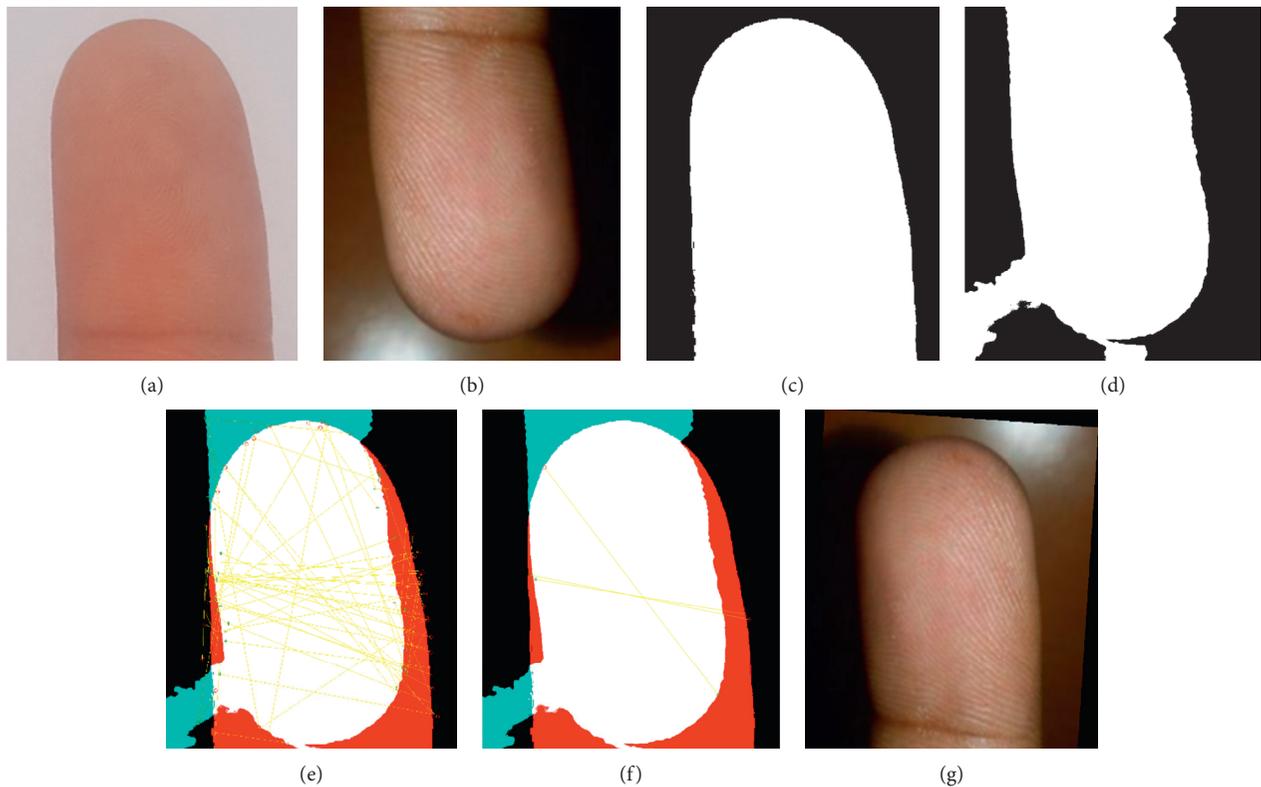


FIGURE 3: Stages of rotation alignment: (a) reference intensity image, (b) input intensity image, (c) segmented reference image, (d) segmented input image, (e) matched features including outliers, (f) matched inliers, and (g) derotated input image.

$S_{\text{ref}}^{\text{input}} > 1.1$, the images are interpolated by a factor of $S_{\text{ref}}^{\text{input}}$ using simple bilinear interpolation.

2.2.3. Feature Extraction and Matching. Intensity images are converted into binary images for time-efficient feature matching. Since finger shapes are all similar, features detected along their boundaries can provide better estimations of geometrical differences. For feature detection, the SURF [28] method is applied to the segmented binary images to obtain boundary features for both images. SURF is selected for its robustness and reasonable match rate for two same images with different angle of rotation [29]. After obtaining features, nearest neighbor-based feature matching by computing the pairwise distance between features is performed. Then, feature matching provides matched features that contain truly matched inliers as well as erroneously matched outliers. Figure 3(e) shows all the matched features including both inliers and outliers.

2.2.4. Inlier Extraction and Derotation of Image. To extract inliers from the matched points, an M-estimator [29] algorithm is applied. Figure 3(f) shows the inliers extracted from all matched features using the M-estimator. It can be seen that extracted inliers represent true matches that can be used to estimate the angle of rotation between accurately matched inlier points.

Once several potential inliers are extracted, the angle of rotation between the matched inliers must be determined. A homography matrix [30] is calculated for this purpose. For the two sets of matched points, $\mathbf{D} = (x_i, y_i)$ for the reference image and $\mathbf{D}' = (x'_i, y'_i)$ for the input image. Then, the relation between them is as given in the following equation:

$$\begin{aligned} \mathbf{D}' &= \mathbf{H}\mathbf{D}, \\ \text{or } \mathbf{H} &= \mathbf{D}'\mathbf{D}^{-1}, \end{aligned} \quad (2)$$

where

$$\begin{aligned} \mathbf{D} &= \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}, \\ \mathbf{D}' &= \begin{bmatrix} x'_i \\ y'_i \\ 1 \end{bmatrix}, \end{aligned} \quad (3)$$

The resultant matrix in (3) is of the form given in (4) as

$$\mathbf{H} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} = \begin{bmatrix} \varepsilon \cos(\theta) & -\sin(\theta) & t_x \\ \varepsilon \sin(\theta) & \cos \theta & t_y \\ 0 & 0 & 1 \end{bmatrix}. \quad (4)$$

Therefore, the angle of rotation can be calculated using h_{12} or h_{22} using (5) as

$$\begin{aligned} \theta &= \sin^{-1}(-h_{12}), \\ \theta &= \cos^{-1}(-h_{22}), \end{aligned} \quad (5)$$

where θ can be used to derotate the input image to obtain an image with a vertically aligned finger. Figure 3(g) shows the derotation of the input image after the calculation of $\theta = -176.29^\circ$. Figure 4 shows the alignment method applied

to various input images showing fingers at different angles of rotation. It can be seen that the proposed algorithm predicted the angles of rotation accurately and that the fingers in the derotated images are aligned vertically irrespective of their initial alignment.

2.3. Image Enhancement. Edge-based image enhancement is applied to the grayscale image to improve the visibility of ridges on the finger. Edge-based enhancement is performed in two stages. First, unsharp masking [32] is applied to the image. This algorithm subtracts an image from a blurred version of itself and enhances those regions which provide a good response after subtraction. To calculate the blurred image, a Gaussian filter is convolved with the original grayscale image using the following equation:

$$I_{\text{gray}} * G_\sigma, \quad (6)$$

where

$$G_\sigma = \frac{1}{\sqrt{\pi}\sigma} e^{-x^2/2\sigma^2}. \quad (7)$$

For the second stage of image enhancement, a contextual blockwise Fourier domain filter [33] is applied to the grayscale image by dividing it into a 16×16 overlapping block. These blocks are then filtered in the Fourier domain using a frequency and orientation-selective filter with parameters based on the estimated local ridge orientation and frequency. The outcome of the Fourier domain-based enhancement is another image denoted as I_{enhanced} . I_{enhanced} is multiplied with a skin-segmented binary image so that only the region of interest (ROI) is processed further.

Figure 5 shows the effects of this method on the derotated input image. The image in Figure 5(a) shows ridges that are not visible. The results of the unsharp masking can be seen in Figure 5(b) showing improved visibility of the ridges. Figure 5(c) shows the image once it has been converted to grayscale. Figure 5(d) shows that application of Fourier-based image enhancement further sharpens unclear edges and other details. Figure 5(e) shows the extracted ROI.

2.4. Image Binarization and Edge Thinning. Features such as ridge endings and bifurcations can be accurately extracted when ridges having a thickness of one pixel are detected. To detect thin ridges, all ridges are first detected by applying Otsu thresholding [34] to the enhanced region of interest. This will result in a binary image with ridge pixels represented in white and the remaining pixels in black. However, this will result in the detection of thick ridges. To limit the results to thin ridges, nonmaximal suppression [35] is applied. This method sets all the ridge pixels in a certain direction to zero if they do not have a maximum gradient value and retains ridge pixels with the maximal value. This results in conversion of the thick ridges to ridges with a thickness of only one pixel. Figure 6(a) shows Otsu thresholding being applied to the enhanced region of interest. The detected ridges are thick and thus cannot be used for feature detection. Application of nonmaximal

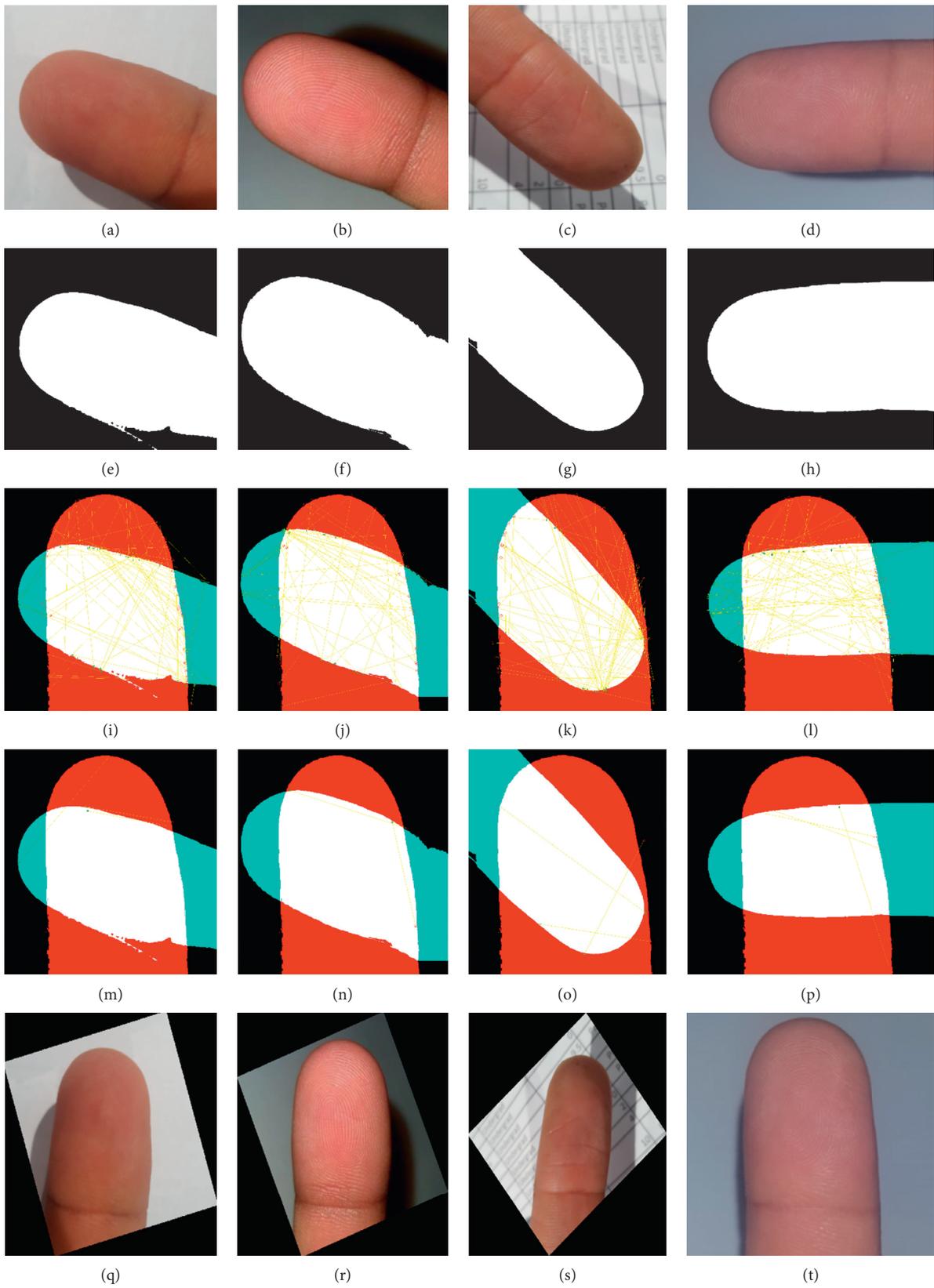


FIGURE 4: Stages of rotation alignment method applied to four images: (a–d) input intensity image, (e–h) segmented input image, (i–l) matched features including outliers, (m–p) matched inliers, and (q–t) derotated input image.

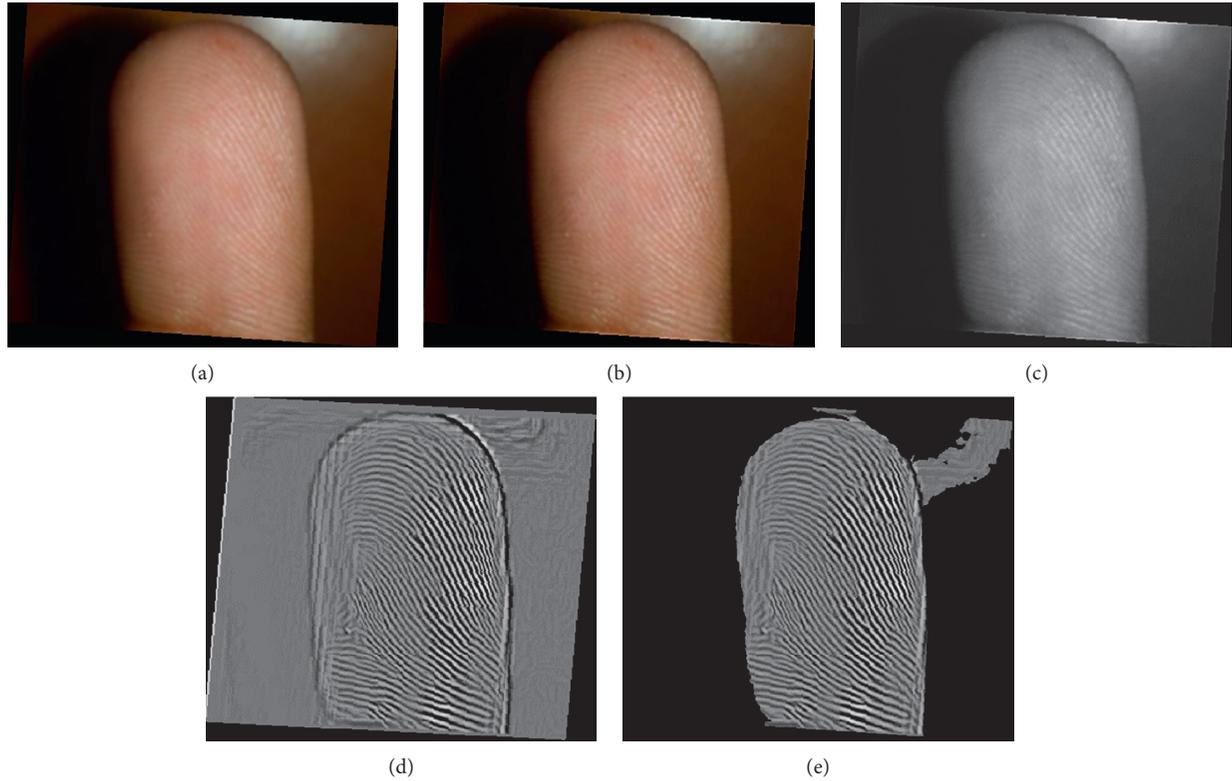


FIGURE 5: Application of preprocessing on the fingerprint image: (a) original input image, (b) unsharp masking, (c) grayscale conversion, (d) blockwise Fourier-based enhancement, and (e) extracted region of interest.

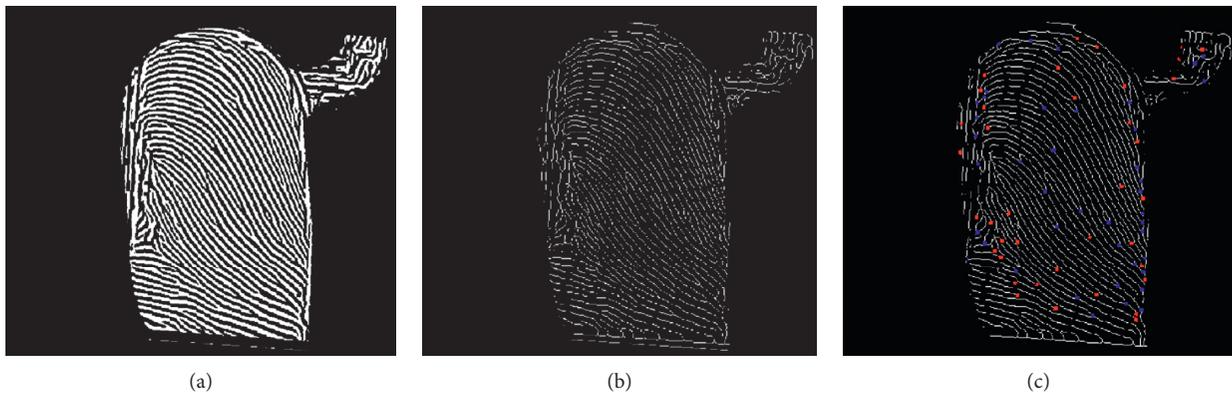


FIGURE 6: Applying edge thinning operations to an enhanced image: (a) binary segmentation, (b) nonmaximal suppression, and (c) minutiae extraction.

suppression results in more desirable thin ridges, as seen in Figure 6(b).

2.5. Minutiae Extraction. Minutiae are features that are used for biometric verification. Instead of storing fingerprint images in the database, minutiae are stored and compared. The most common method used for minutiae extraction is the crossing number (CN) [36].

For extracting minutiae, the 3×3 window shown in Figure 7 is moved across each pixel to calculate its crossing number using the following equation:

$$CN(x, y) = \frac{1}{2} \sum_{i=1}^8 |p_i - p_{i-1}|, \quad (8)$$

where p_i denotes the pixels around the center pixel at location (x, y) . Using CN, the ridge pixel can then be classified as a ridge ending, bifurcation, or isolated point using Table 1. Figure 6(c) shows bifurcations and ridge endings represented by blue and red squares, respectively.

For each bifurcation and ridge ending, the Euclidean distance $D_{x,y}$ between its spatial coordinate (x, y) and $(x_{ROI}^{\min}, y_{ROI}^{\min})$, the normalized gradient value $G^{x,y}$, and the

P_4	P_3	P_2
P_5	(x,y)	P_1
P_6	P_7	P_8

FIGURE 7: 3 × 3 window used in minutiae extraction.

TABLE 1: Ridge pixel classifications using the crossing number.

CN	Decision
0	Isolated point
1	Ridge ending
2	Continuing ridge
3	Bifurcation
4	Crossing

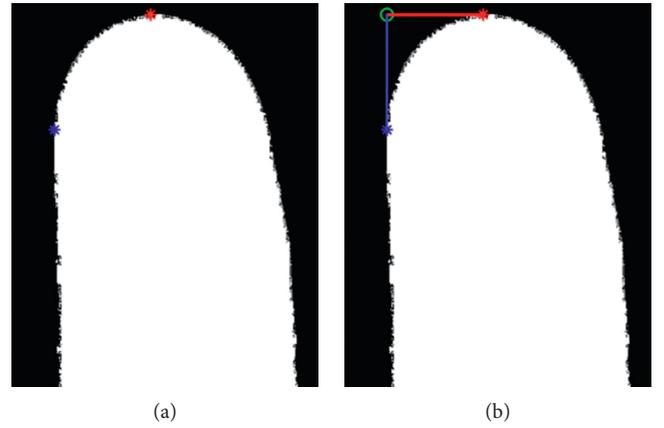
gradient orientation $\alpha^{x,y}$ are stored in $\mathbf{S}_{\text{BF}}^{\text{in}}$ and $\mathbf{S}_{\text{RE}}^{\text{in}}$, respectively, for storage or comparison. Hence, every bifurcation and ridge ending point is stored in $\mathbf{S}_{\text{BF}}^{\text{in}}$ or $\mathbf{S}_{\text{RE}}^{\text{in}}$ in the form of 3D point $(D_{\text{BF}}^{x,y}, G_{\text{BF}}^{x,y}, \alpha_{\text{BF}}^{x,y})$ and $(D_{\text{RE}}^{x,y}, G_{\text{RE}}^{x,y}, \alpha_{\text{RE}}^{x,y})$, respectively. The values donated by $x_{\text{ROI}}^{\text{min}}$ and $y_{\text{ROI}}^{\text{min}}$ are x -axis and y -axis points taken from two ROI points corresponding to the minimum x -axis and y -axis values. Figure 8 shows an example of calculation of $(x_{\text{ROI}}^{\text{min}}, y_{\text{ROI}}^{\text{min}})$ that is represented by a green circle. This was determined using the minimum x -axis ROI point represented by a blue cross and the minimum y -axis ROI point represented by a red cross.

2.6. Fingerprint Registration and Matching. Extracted minutiae are either stored or compared with minutiae already existing in the database as required by the use case. Suppose that the ridge endings extracted from the input image are represented by the set $\mathbf{S}_{\text{BF}}^{\text{in}}$ and that bifurcations are represented by $\mathbf{S}_{\text{RE}}^{\text{in}}$ as follows:

$$\mathbf{S}_{\text{RE}}^{\text{in}} = \left\{ (D_{\text{RE}1}^{\text{in}}, G_{\text{RE}1}^{\text{in}}, \alpha_{\text{RE}1}^{\text{in}}), (D_{\text{RE}2}^{\text{in}}, G_{\text{RE}2}^{\text{in}}, \alpha_{\text{RE}2}^{\text{in}}), \dots, (D_{\text{REN}_{\text{RE}}^{\text{in}}}^{\text{in}}, G_{\text{REN}_{\text{RE}}^{\text{in}}}^{\text{in}}, \alpha_{\text{REN}_{\text{RE}}^{\text{in}}}^{\text{in}}) \right\}, \quad (9)$$

$$\mathbf{S}_{\text{BF}}^{\text{in}} = \left\{ (D_{\text{BF}1}^{\text{in}}, G_{\text{BF}1}^{\text{in}}, \alpha_{\text{BF}1}^{\text{in}}), (D_{\text{BF}2}^{\text{in}}, G_{\text{BF}2}^{\text{in}}, \alpha_{\text{BF}2}^{\text{in}}), \dots, (D_{\text{BFN}_{\text{BF}}^{\text{in}}}^{\text{in}}, G_{\text{BFN}_{\text{BF}}^{\text{in}}}^{\text{in}}, \alpha_{\text{BFN}_{\text{BF}}^{\text{in}}}^{\text{in}}) \right\}, \quad (10)$$

where $\mathbf{S}_{\text{RE}}^{\text{in}}$ and $\mathbf{S}_{\text{BF}}^{\text{in}}$ are sorted in an ascending order according to the calculated Euclidean distance such that $D_{\text{RE}}^{\text{in}(k)} \leq D_{\text{RE}}^{\text{in}(k+1)}$ and $D_{\text{BF}}^{\text{in}(l)} \leq D_{\text{BF}}^{\text{in}(l+1)}$ for all $k \in \{1, 2, \dots, N_{\text{RE}}^{\text{in}}\}$ and $l \in \{1, 2, \dots, N_{\text{BF}}^{\text{in}}\}$. These sets must be compared with the stored features of a number of fingerprints N_{DB} stored in the database, as follows:

FIGURE 8: Determination of point $(x_{\text{ROI}}^{\text{min}}, y_{\text{ROI}}^{\text{min}})$: (a) minimum x -axis and y -axis ROI points and (b) $(x_{\text{ROI}}^{\text{min}}, y_{\text{ROI}}^{\text{min}})$ determination using two minimum axis points.

$$\mathbf{S}_{\text{RE}}^{\text{DB}} = \left\{ \mathbf{S}_{\text{RE}}^{\text{DB}1}, \mathbf{S}_{\text{RE}}^{\text{DB}2}, \dots, \mathbf{S}_{\text{RE}}^{\text{DBN}_{\text{RE}}^{\text{DB}}} \right\}, \quad (11)$$

$$\mathbf{S}_{\text{BF}}^{\text{DB}} = \left\{ \mathbf{S}_{\text{BF}}^{\text{DB}1}, \mathbf{S}_{\text{BF}}^{\text{DB}2}, \dots, \mathbf{S}_{\text{BF}}^{\text{DBN}_{\text{BF}}^{\text{DB}}} \right\}, \quad (12)$$

where

$$\mathbf{S}_{\text{RE}}^{\text{DB}i} = \left\{ (D_{\text{RE}1}^{\text{DB}i}, G_{\text{RE}1}^{\text{DB}i}, \alpha_{\text{RE}1}^{\text{DB}i}), (D_{\text{RE}2}^{\text{DB}i}, G_{\text{RE}2}^{\text{DB}i}, \alpha_{\text{RE}2}^{\text{DB}i}), \dots, (D_{\text{REN}_{\text{RE}}^{\text{DB}i}}^{\text{DB}i}, G_{\text{REN}_{\text{RE}}^{\text{DB}i}}^{\text{DB}i}, \alpha_{\text{REN}_{\text{RE}}^{\text{DB}i}}^{\text{DB}i}) \right\}, \quad (13)$$

$$\mathbf{S}_{\text{BF}}^{\text{DB}i} = \left\{ (D_{\text{BF}1}^{\text{DB}i}, G_{\text{BF}1}^{\text{DB}i}, \alpha_{\text{BF}1}^{\text{DB}i}), (D_{\text{BF}2}^{\text{DB}i}, G_{\text{BF}2}^{\text{DB}i}, \alpha_{\text{BF}2}^{\text{DB}i}), \dots, (D_{\text{BFN}_{\text{BF}}^{\text{DB}i}}^{\text{DB}i}, G_{\text{BFN}_{\text{BF}}^{\text{DB}i}}^{\text{DB}i}, \alpha_{\text{BFN}_{\text{BF}}^{\text{DB}i}}^{\text{DB}i}) \right\}. \quad (14)$$

$\mathbf{S}_{\text{RE}}^{\text{DB}i}$ and $\mathbf{S}_{\text{BF}}^{\text{DB}i}$ are also sorted in an ascending order according to the Euclidean distances $D_{\text{RE}}^{\text{DB}i m}$ and $D_{\text{BF}}^{\text{DB}i n}$ where $m \in \{1, 2, \dots, N_{\text{RE}}^{\text{DB}i}\}$ and $n \in \{1, 2, \dots, N_{\text{BF}}^{\text{DB}i}\}$. Suppose that $\mathbf{S}_{\text{RE}}^{\text{in}}$ and $\mathbf{S}_{\text{BF}}^{\text{in}}$ are to be compared with the stored features of the i th fingerprint having feature sets $\mathbf{S}_{\text{RE}}^{\text{DB}i}$ and $\mathbf{S}_{\text{BF}}^{\text{DB}i}$. First, $N_{\text{min}}^{\text{RE}}, N_{\text{min}}^{\text{BF}}, N_{\text{max}}^{\text{RE}}$, and $N_{\text{max}}^{\text{BF}}$ are to be determined using the following equations:

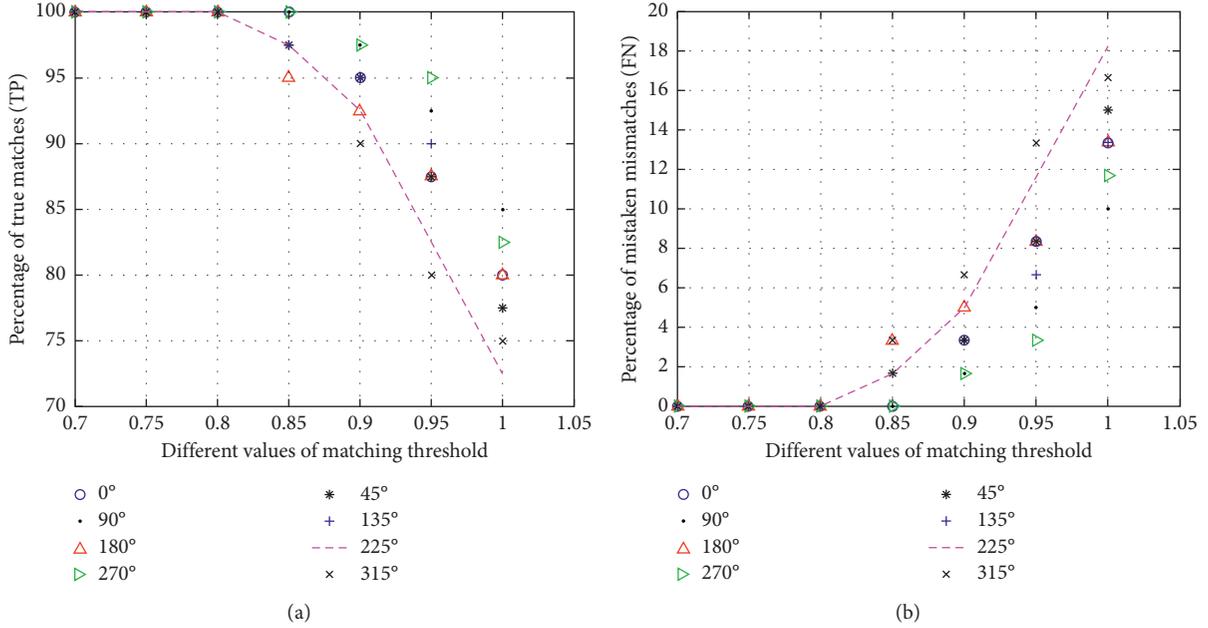


FIGURE 9: Percentage of (a) TP and (b) FN for different values of t_{match} .

$$N_{\min}^{\text{RE}} = \min(N_{\text{RE}}^{\text{in}}, N_{\text{RE}}^{\text{DBi}}), \quad (15)$$

$$N_{\min}^{\text{BF}} = \min(N_{\text{BF}}^{\text{in}}, N_{\text{BF}}^{\text{DBi}}), \quad (16)$$

$$N_{\max}^{\text{RE}} = \max(N_{\text{RE}}^{\text{in}}, N_{\text{RE}}^{\text{DBi}}), \quad (17)$$

$$N_{\max}^{\text{BF}} = \max(N_{\text{BF}}^{\text{in}}, N_{\text{BF}}^{\text{DBi}}), \quad (18)$$

where $N_{\text{RE}}^{\text{in}}$ and $N_{\text{RE}}^{\text{DBi}}$ are the number of ridge endings extracted from the input image and the i th image features stored in the database and $N_{\text{BF}}^{\text{in}}$ and $N_{\text{BF}}^{\text{DBi}}$ are the number of extracted bifurcations. The input fingerprint features are compared with the i th fingerprint features if the condition in (19) is justified:

$$\begin{aligned} N_{\min}^{\text{RE}} &\geq 0.8N_{\max}^{\text{RE}}, \\ N_{\min}^{\text{BF}} &\geq 0.8N_{\max}^{\text{BF}}. \end{aligned} \quad (19)$$

For matching, all 3D bifurcation and ridge ending points of the input fingerprint and the i th fingerprint from the database and input fingerprint are stored in $\mathbf{M}_{\text{RE}}^{\text{ref}}$, $\mathbf{M}_{\text{BF}}^{\text{ref}}$, $\mathbf{M}_{\text{RE}}^{\text{in}}$, and $\mathbf{M}_{\text{BF}}^{\text{in}}$ respectively. $\mathbf{M}_{\text{RE}}^{\text{ref}}$ is compared with $\mathbf{M}_{\text{RE}}^{\text{in}}$, and $\mathbf{M}_{\text{BF}}^{\text{ref}}$ is compared with $\mathbf{M}_{\text{BF}}^{\text{in}}$. The steps for matching are

- If $\mathbf{M}_{\text{BF}}^{\text{in}}$ has fewer bifurcations, store $\mathbf{M}_{\text{BF}}^{\text{in}}$ in $\mathbf{M}_{\text{BF}}^{\text{min}}$ and $\mathbf{M}_{\text{BF}}^{\text{ref}}$ in $\mathbf{M}_{\text{BF}}^{\text{max}}$ and vice versa. Initialize bifurcation matching score score_{BF} to zero.
- Compare each bifurcation 3D point in $\mathbf{M}_{\text{BF}}^{\text{min}}$ at the p th position with all bifurcation points at locations $p+t$ in $\mathbf{M}_{\text{BF}}^{\text{max}}$ where $t \in \{-5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5\}$.
- If $|G_{p+t}^{\text{max}} - G_p^{\text{min}}| < T_1$ and $|\alpha_{p+t}^{\text{max}} - \alpha_p^{\text{min}}| < T_2$ is true for any value of t , then increment score_{BF} and move to the next bifurcation point for matching. Here, G_p^{min}

and α_p^{min} are the magnitude and orientation of the normalized gradient of bifurcation of $M_{\text{BF}}^{\text{min}}$ at position p , and G_{p+t}^{max} and $\alpha_{p+t}^{\text{max}}$ are the magnitude and orientation of $M_{\text{BF}}^{\text{max}}$ at any of the eleven locations in Step b.

- Repeat Steps a–c, comparing ridge endings and updating score_{RE} to track the number of ridges matched.
- If $(\text{score}_{\text{BF}} + \text{score}_{\text{RE}}) / (N_{\min}^{\text{BF}} + N_{\min}^{\text{RE}}) > t_{\text{match}}$, two fingerprints are said to be matched.

3. Results and Discussion

MATLAB 2018a was used in this study. A value of $\sigma = 2.5$ was used, and a Gaussian window of size $(2\sigma + 1) \times (2\sigma + 1)$, $T_1 = .06$, and $T_2 = 10^\circ$ was used throughout the work. There was no dataset available for evaluation purposes, so features from the fingerprints of 40 people were stored in a database. Features obtained from digital images of the fingers of 100 people, including those of registered people, were compared with the features already stored in the database in order to evaluate the performance of the proposed algorithm. Every person provided eight images of their finger at approximately 90° , 135° , 180° , 225° , 270° , 315° , 0° , and 45° .

The performance of the proposed method was evaluated by analyzing the performance of the derotation and matching methods. It was not possible to calculate the exact errors in the calculation of the rotation angles, so error calculation was done manually for ten random inputs for each set of rotation angles.

According to this manual evaluation, the errors fell in the range from ~ -2.5 to $\sim 2.5^\circ$ for all cases. The error range for the angles 0° , 180° , and 270° was from $\sim -1.5^\circ$ to

1.5°. For a rotation angle of 90°, the error was approximately 0°.

True positives, true negatives, false positives, and false negatives were calculated for different values of the matching threshold t_{match} . True positives, true negatives, false positives, and false negatives are given in (20)–(23) as

$$\text{TP} = \frac{n_{\text{correctMatches}} \times 100}{n_{\text{ActualMatches}}}, \quad (20)$$

$$\text{TN} = \frac{n_{\text{correctMissMatches}} \times 100}{n_{\text{ActualMismatches}}}, \quad (21)$$

$$\text{FP} = \frac{n_{\text{mistakenMatches}} \times 100}{n_{\text{total}}}, \quad (22)$$

$$\text{FN} = \frac{n_{\text{mistakenMisMatches}} \times 100}{n_{\text{ActualMismatches}}}. \quad (23)$$

Figure 9 shows a comparison of the percentages of true matches (TP) and mistaken mismatches (FN). It can be seen from Figure 9 that a value of $0.7 \leq t_{\text{match}} \leq 0.8$ successfully verified all 40 of the registered fingerprints irrespective of their angle of rotation. However, for values of $t_{\text{match}} > 0.8$, mistaken mismatches occurred in the range from $\sim(2\% - 18\%)$. True mismatches (TN) occurred in 100% of the cases, and mistaken matches (FP) occurred in 0% of all cases.

4. Conclusions and Future Work

This paper proposes a fingerprint detection and verification method that uses a digital image of a finger as an input instead of a sensor. The input fingerprint image can be of any resolution and depends on the quality of the digital camera. Often, the finger in the image is not vertically aligned. Therefore, the input image is derotated by applying skin color-based binary segmentation, followed by downscaling or upscaling of the input image with a scale calculated using the segmented input image and a reference image. The angle of rotation of the input finger is then calculated by applying SURF-based feature extraction on both binary images. The features detected in both images are matched, and outliers are removed from the matched points using the M-estimator. The angle between the matched inliers is calculated using a homography matrix, and the input image is derotated by the calculated angle. After derotation, enhancement techniques including unsharp masking and Fourier-based blockwise enhancement are applied to increase the visibility of the ridges in the input image. Image enhancement is followed by the application of Otsu thresholding to segment the ridge regions, and then nonmaximal suppression is applied for edge thinning. A crossing number technique is used to extract ridge endings and bifurcations. This is followed by the proposed Euclidean distance-based matching to detect a fingerprint match or mismatch based on a matching score. The proposed

method is a novel method, and simulation results show that if the matching threshold is kept within a suggested range, accuracy of 100% can be achieved.

Scar removal-based fingerprint detection using digital images and edge-preserving image interpolation techniques could extend the proposed method and is being considered for future work.

Abbreviations

STFT:	Short-time Fourier transform analysis
ROI:	Region of interest
SURF:	Speeded up robust features
M-estimator:	Maximum-likelihood estimator
CN:	Crossing number
TP:	True positives
TN:	True negatives
FP:	False positives
FN:	False negatives.

Data Availability

The data used to support the findings of this study can be obtained by contacting the corresponding author through e-mail: dhlee77@hanyang.ac.kr.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

Sajid Khan performed all the simulations and worked on the solution of most of the problems under the supervision of Dong-Ho Lee. Asif Khan provided helped in finalization of the content of manuscript and writing of the manuscript. Ahmad Waqas and Abdul Rehman Gilal collected the fingerprints for experiments and validation. Zahid Hussain Khand helped in the understanding of mathematical models and equations that are used in this research.

References

- [1] S. Khan and D. H. Lee, "An adaptive dynamically weighted median filter for impulse noise removal," *EURASIP Journal on Advances in Signal Processing*, vol. 2017, no. 1, pp. 1–14, 2017.
- [2] S. Khan, D.-H. Lee, M. A. Khan, A. R. Gilal, and G. Mujtaba, "Efficient edge-based image interpolation method using neighboring slope information," *IEEE Access*, vol. 7, pp. 133539–133548, 2019.
- [3] S. Khan and D. Lee, "Efficient deinterlacing method using simple edge slope tracing," *Optical Engineering*, vol. 54, no. 10, pp. 103108–103111, 2015.
- [4] M. J. Burge and K. W. Bowyer, *Handbook of Iris Recognition*, Springer Publishing Company, Berlin, Germany, 2013.
- [5] D. S. Moon, K. Y. Moon, Y. O. O. Jang-Hee et al., "Fingerprint verification method and apparatus with high security," U.S. Patent and Trademark Office, Washington, DC, USA, U.S. Patent No. 8,699,799, 2014.
- [6] H. Guesmi, H. Trichili, A. M. Alimi, and B. Solaiman, "Fingerprint verification system based on curvelet transform

- and possibility theory,” *Multimedia Tools and Applications*, vol. 74, no. 9, pp. 3253–3272, 2015.
- [7] K. B. Raja, R. Raghavendra, V. K. Vemuri, and C. Busch, “Smartphone based visible iris recognition using deep sparse filtering,” *Pattern Recognition Letters*, vol. 57, pp. 33–42, 2015.
 - [8] R. Jafri and H. R. Arabnia, “A survey of face recognition techniques,” *Journal of Information Processing Systems*, vol. 5, no. 2, pp. 41–68, 2009.
 - [9] O. M. Parkhi, A. Vedaldi, and A. Zisserman, “Deep face recognition,” *BMVC*, vol. 1, no. 3, pp. 1–12, 2015.
 - [10] Y. Taigman, M. Yang, M. A. Ranzato, and L. Wolf, “Deepface: closing the gap to human-level performance in face verification,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1701–1708, Columbus, OH, USA, June 2014.
 - [11] H. Rai and A. Yadav, “Iris recognition using combined support vector machine and Hamming distance approach,” *Expert Systems with Applications*, vol. 41, no. 2, pp. 588–593, 2014.
 - [12] A. I. Ismail, H. S. Ali, and F. A. Farag, “Efficient enhancement and matching for iris recognition using SURF,” in *Proceedings of the Information Technology: Towards New Smart World*, pp. 1–5, Riyadh, Saudi Arabia, February 2015.
 - [13] Z. M. Kovacs-Vajna, “A fingerprint verification system based on triangular matching and dynamic time warping,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1266–1276, 2000.
 - [14] X. Luo, J. Tian, and Y. Wu, “A minutiae matching algorithm in fingerprint verification,” *Pattern Recognition*, vol. 4, no. 15, pp. 833–836, 2000.
 - [15] R. Cappelli, D. Maio, D. Maltoni, J. L. Wayman, and A. K. Jain, “Performance evaluation of fingerprint verification systems,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 1, pp. 3–18, 2006.
 - [16] Y. He, J. Tian, X. Luo, and T. Zhang, “Image enhancement and minutiae matching in fingerprint verification,” *Pattern Recognition Letters*, vol. 24, no. 9–10, pp. 1349–1360, 2003.
 - [17] S. Khan, A. Waqas, M. A. Khan, and A. W. Ahmad, “A camera-based fingerprint registration and verification method,” *International Journal of Computer Science and Network Security*, vol. 18, no. 11, pp. 26–31, 2018.
 - [18] K. Delac and M. Grgic, “A survey of biometric recognition methods,” in *Proceedings of the Elmar-2004. 46th International Symposium on Electronics in Marine*, pp. 184–193, Zadar, Croatia, June 2004.
 - [19] P. Mahajan, M. Malekar, A. More, A. Wairagade, and B. Mahalakshmi, “Secured internet banking using fingerprint authentication,” *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 4, no. 3, pp. 6199–6208, 2016.
 - [20] R. Priya, V. Tamilselvi, and G. P. Rameshkumar, “A novel algorithm for secure internet banking with finger print recognition,” in *Proceedings of the International Conference on Embedded Systems (ICES)*, pp. 104–109, Coimbatore, India, July 2014.
 - [21] D. Baldisserra, A. Franco, D. Maio, and D. Maltoni, “Fake fingerprint detection by odor analysis,” in *Proceedings of the International Conference on Biometrics*, pp. 265–272, Hong Kong, China, January 2006.
 - [22] S. B. Nikam and S. Agarwal, “Ridgelet-based fake fingerprint detection,” *Neurocomputing*, vol. 72, no. 10–12, pp. 2491–2506, 2009.
 - [23] H. Choi, R. Kang, K. Choi, A. T. B. Jin, and J. H. Kim, “Fake-fingerprint detection using multiple static features,” *Optical Engineering*, vol. 48, no. 4, Article ID 047202, 2009.
 - [24] B. Y. Hiew, A. B. J. Teoh, and O. S. Yin, “A secure digital camera based fingerprint verification system,” *Journal of Visual Communication and Image Representation*, vol. 21, no. 3, pp. 219–231, 2010.
 - [25] B. Y. Hiew, A. B. Teoh, and Y. H. Pang, “Digital camera based fingerprint recognition,” in *Proceedings of the IEEE International Conference on Telecommunications and Malaysia International Conference on Communications*, pp. 676–681, Penang, Malaysia, May 2007.
 - [26] B. Y. Hiew, A. B. Teoh, and D. C. Ngo, “Automatic digital camera based fingerprint image preprocessing,” in *Proceedings of the International Conference on Computer Graphics, Imaging and Visualisation*, pp. 182–189, Sydney, Australia, July 2006.
 - [27] A. Sankaran, A. Malhotra, A. Mittal, M. Vatsa, and R. Singh, “On smartphone camera based fingerphoto authentication,” in *Proceedings of the International Conference on Biometrics Theory, Applications and Systems (BTAS)*, pp. 1–7, Arlington, VA, USA, September 2015.
 - [28] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, “Speeded-up robust features (SURF),” *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.
 - [29] D. E. Tyler, “A distribution-free $\$M\$$ -Estimator of multivariate scatter,” *The Annals of Statistics*, vol. 15, no. 1, pp. 234–251, 1987.
 - [30] Z. Zhang, “Flexible camera calibration by viewing a plane from unknown orientations,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 666–673, Kerkyra, Greece, September 1999.
 - [31] S. L. Phung, D. Chai, and A. Bouzerdoud, “Adaptive skin segmentation in color images,” in *Proceedings of the ICASSP*, pp. 353–356, Hong Kong, China, April 2003.
 - [32] G. Ramponi, N. K. Strobel, S. K. Mitra, and T. H. Yu, “Nonlinear unsharp masking methods for image contrast enhancement,” *Journal of Electronic Imaging*, vol. 5, no. 3, pp. 353–367, 1996.
 - [33] S. Chikkerur, C. Wu, and V. Govindaraju, “A systematic approach for feature extraction in fingerprint images,” in *Biometric Authentication*, pp. 344–350, Springer, Berlin, Germany, 2004.
 - [34] N. Otsu, “A threshold selection method from gray-level histograms,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.
 - [35] J. Canny, “A computational approach to edge detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679–698, 1986.
 - [36] L. Wieclaw, “A minutiae-based matching algorithms in fingerprint recognition systems,” *Journal of Medical Informatics & Technologies*, vol. 13, pp. 65–72, 2009.