



Palmprint Recognition Using Hessian Matrix and Two-Component Partition Method

Jyotismita Chaki, VIT University, Vellore, India

 <https://orcid.org/0000-0003-1804-8590>

Nilanjan Dey, JIS University, Kolkata, India

 <https://orcid.org/0000-0001-8437-498X>

ABSTRACT

Palmprint recognition has been comprehensively examined in the past couple of years and various undertakings are done to use it as a biometric methodology for various applications. The point of this study is to construct an effective palmprint recognition technique with low computational multifaceted nature and along these lines to expand the acknowledgment and precision. Since edges are free from distortion, they are very reliable and subsequently used for palm print recognition. The originality of the proposed technique depends on new area of interest (ROI) extraction took after by new principal line extraction and texture matching strategy. The new principal line extraction technique is created by using the Hessian matrix and Eigen value. The texture matching of the ROI is done using new 2-component partition method by segmenting the image into comparative and non-comparative edges. Examinations are finished on a database and exploratory results exhibit that the accuracy of the proposed method is comparable to past methods used for palmprint recognition.

KEYWORDS

Comparative Edges, Edge Detection, Non-Comparative Edges, Palmprint, Principal Line, ROI Extraction, Texture

INTRODUCTION

Checking an individual character with high accuracy is desired in various applications, for instance, some national foundations get the opportunity to control, e-keeping cash, exit and passage and so on. Biometric recognizing verification advancement is a kind of strategies to feasibly approve the identity of a man in perspective of physiological or behavioral characteristics. In connection with ID card or secret scratch card, biometric recognizing verification development is very useful, fruitful and secure with many far-reaching applications.

The aim of this study is to construct an effective palmprint recognition technique from the principal lines. Since palm lines are free from distortion, they are very reliable and thus can be used for palmprint recognition. The originality of the proposed technique includes the development of

DOI: 10.4018/IJDCF.2021010102

This article, published as an Open Access article on February 4, 2021 in the gold Open Access journal, International Journal of Digital Crime and Forensics (IJDCF) (converted to gold Open Access January 1, 2021), is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>) which permits unrestricted use, distribution, and production in any medium, provided the author of the original work and original publication source are properly credited.

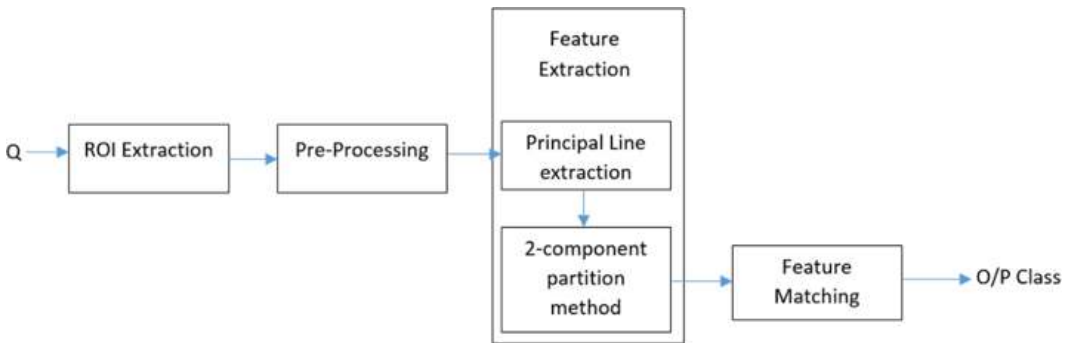
a new method for extracting the region of interest (ROI) followed by new principal line extraction and texture matching technique. The new principal line extraction technique is created by using the Hessian matrix and Eigenvalue. The feature extraction of the ROI is done by using a new 2-component partition method where the principal lines are segmented into comparative and non-comparative edges. The recognition of palmprint image is done by comparing or matching the comparative and non-comparative edges between the training and testing images.

The arrangement of the paper is according to the following: segment 2 describes previous works, segment 3 diagrams the proposed approach, segment 4 gives the details of experimentation and results, segment 5 analyzes the proposed approach with other contemporary methodologies, while segment 6 mentions the general conclusions.

PREVIOUS WORKS

Before decades, distinctive sorts of biometric recognition approaches have been made (Dey et.al. 2014; Rajeswari et.al. 2017), including face, iris, palmprint, finger-knuckle print (Kumar, 2018), hand geometry etc. Palmprint-based biometric systems have been attracting much interest since they can achieve high accuracy. Various visual features, information demonstrating methods and classifiers have been proposed for palmprint recognition. The low-resolution palmprint recognition is finished by Zhang et.al. (2003). Here 2D Gabor phase encoding is utilized for the distinguishing proof of the palmprint images. Zhang et.al. (2006) utilizes a feature level fusion approach by numerous elliptical Gabor filter with various orientation for palmprint recognition. The palmprint recognition by utilizing image sharpness is done by Zhang et.al. (2017). A palmprint image quality appraisal method based on Edge Acutance value is embraced to assess whether the image sharpness is suitable for the recognition or not. Karar et.al. (2012) utilized magnitude of the real and imaginary parts of the complex Zernike moment by using an arrangement of a complex polynomial which produces a total orthogonal premise set characterized on the unit plate for palmprint recognition. The palmprint recognition by using thirteen features of Gray Level Co-occurrence Matrix (GLCM) is done by Zhu et.al. (2011). At last, by using the SVM classifier palmprint images are recognized. Guo et.al. (2010) utilized the multiscale Local Binary Pattern (LBP) histogram for palmprint recognition. Initially, the pattern is isolated into uniform (histogram) and non-uniform (LBP pattern) pixel. The procedure stops when the pixels become uniform. Hong et.al. (2014) utilized enhanced differential box-counting with the multiscale and multidirectional strategy of Gabor and Curvelet transform for the recognition of palmprint images. Fei et.al. (2016) and Shrivastava et.al. (2013) used six orientations of Gabor channel and 6-bit code plane feature for the recognition of palmprint images. Binary orientation co-occurrence vector is used for palmprint recognition by Guo et.al. (2009). A threshold selection in light of binary value distribution was proposed for the BOCV scheme. Hong et.al. (2015) utilized a combination of multispectral Block-based Histogram of Oriented Gradient (BHOG) and Block Dominant Orientation Code (BDOC) for palmprint recognition. George et.al. (2014) used minute statistical features like orientation field, region map, and density map for the classification of palmprint images. Finally based upon the similarity score the palmprint images are distinguished. The double half orientation based strategy is utilized for feature extraction and classification of the palmprint by Fei et.al. (2016). A bank of “half-Gabor” filters are characterized for the half-orientation extraction of a palmprint. Six groups of the double half-Gabor filter with six sorts of orientation are convolved with the palmprint image. Orientation invariance feature extraction is used by Feng et.al. (2015) for the classification of palmprint images. Initially, to make the palm image rotation invariant, the slope of every pixel is ascertained and the orientation is balanced. From that point forward, to have the feature vector, the histogram of oriented gradient and dominant orientation is combined. Oriented multiscale log Gabor channel is utilized for the recognition of the palmprint images by Bounneche et.al. (2016). The feature extraction utilizes a multi-resolution log-Gabor filter where the last element is made out of the winning codes of the least filter’ bank response. The matching procedure utilizes a

Figure 1. Block Diagram of the proposed approach



bitwise Hamming distance and Kullback–Leibler divergence. At last, the decision stage is conveyed by utilizing a combination of the scores produced from various spectral bands to reduce overlapping. Mahesh et.al. (2014) used the combination of local and global features for the characterization of palmprint images. The local features are calculated by using discrete orthonormal Stockwell transform. To get the global feature the size of the discrete orthonormal Stockwell transform is decreased to infinity. To obtain the final matching distance the global and local features are combined. Roy et.al. (2014) reports the comparative study of different types of thresholding techniques that can be used as image segmentation.

PROPOSED APPROACH

The proposed palmprint recognition method incorporates ROI extraction, Pre-processing, Feature extraction and Feature Matching. A block diagram representing major functional blocks and data flow paths of the proposed approach is shown in figure 1.

Q is the Query image. Prior to preprocessing, a region of interest (abbreviated as ROI) is take out from palm images. To extract the ROI, first, the palm image is binarized and then the boundary of the palm image is extracted. After that by means of the Harries corner-finding algorithm, the corners of the binary image are identified. As the orientation is one of the vital parts of ROI extraction, the furthest right 2 corners are used as the reference point to accurate the alignment of the image. After appropriate alignment, the ROI is cropped based on the corners and kept for further processing. The preprocessing stage is used to normalize the translation aspect and the contrast of the ROI. Feature extraction step includes two-steps: (1) based on Hessian matrix and Eigenvalues a novel principal line extraction technique is designed to extract the principal lines and (2) a new 2-component partition technique is developed to extract the comparative and non-comparative edges from the principal lines of the training and testing images. Lastly, the matching of the feature is done depending on the similarity of the comparative and non-comparative edges between the training and testing samples. Depending on the similarity score the output (O/P) class is identified.

Algorithm 1. Proposed_Approach (Q)

Input Query Image: Q (RGB Image), N (Total number of class), NT (Total number of the testing image)

Output Class: CLS (Output Class)

1: I_{ROI} ← ROI (Q)

2: I_B ← Background_Shrunk (I_{ROI})

- 3: $I_C \rightarrow \text{Contrast_Norm} (I_B)$
- 4: $I_P \rightarrow \text{Principal_Line} (I_C)$
- 5: $[I_{R1}, I_{R2}] \rightarrow \text{2-component_partition} (I_{PT}, I_{PS}) // I_{PT}$ (Training Image), I_{PS} (Testing Image) are divided from I_P
- 6: $CLS \rightarrow \text{Feature_Matching} (I_{R1}, I_{R2}, N, NT)$

ROI Extraction

ROI extraction is one of the vital phases in these four phases since it really influences the overall recognition accuracy of the entire framework. It is essential to take the ROI at a similar place from numerous palmprint images to guarantee the strength accuracy of palmprint features to provide the finest recognition rate and fast processing speed. In this paper, a novel ROI extraction approach is suggested. Image thresholding, Edge tracing, Corner finding, orientation correction and finding the ROI location are successively performed to get a square area which holds the palmprint data. In the succeeding context, the details of each stage are presented.

Image Thresholding

The palm image thresholding procedure is to binarize the RGB image to acquire the binary palm image. In this step, the Otsu thresholding (Vala et.al. 2013) is used for binarization. The motivation of using this technique is its computational ease and higher adaptable quality. Additionally, automatically the threshold value can be selected by using this method and there is no necessity to study and visualize the histogram to discover the appropriate threshold. The image comprises two sorts of pixels: background and foreground. This technique computes the perfect border of separating those two classes with the objective that their combined spread (intra-class change) is negligible. The method comprises repeating through all the possible threshold values and determining an amount of spread for the pixel levels each side of the threshold, i.e. the pixels that fall either in background or foreground. The objective is to determine the threshold value where the combination of background and foreground spreads is at its minimum. To sum things up the threshold value is examined that reduces the intra-class change, which is the same as maximizing interclass change, considered as a weighted summation of variances of the two classes. The intra-class variance is given by (Vala et.al. 2013):

$$\sigma_q^2(t) = q_1(t)\sigma_1^2(t) + q_2(t)\sigma_2^2(t) \quad (1)$$

where q_i are the probabilities of the two classes divided by a threshold t and σ_i^2 variances of these classes. The inter-class variance is specified by:

$$\sigma_c^2(t) = \sigma^2 - \sigma_q^2(t) = q_1(t)q_2(t)[\mu_1(t) - \mu_2(t)]^2 \quad (2)$$

The class probability $q_1(t)$ and class mean $\mu_1(t)$ is computed from the histogram as t:

$$q_1(t) = \sum_{i=0}^t p(i)$$

$$\mu_i(t) = \sum_{i=0}^t p(i) x(i) \quad (3)$$

where $x(i)$ is the value at the center of the i^{th} histogram bin. Desired threshold corresponds to the maximum of $\sigma_c^2(t)$.

Algorithm 2. Otsu (I, L_p)

Input: $I(x, y)$ (RGB Image), $L_p(x, y)$ (Low Pass Filter)
Output: $B(x, y)$ (Binary Image)
 1: For each intensity calculate the histogram and probability
 2: $q_i \leftarrow 0$
 3: $\mu_i \leftarrow 0$
 4: **for** $t \rightarrow 1$ to $\max(t)$. // t is the threshold.
 5: Update q_i and μ_i
 6: Compute $\sigma_c^2(t)$
 7: **end for**
 8: $T = \max(\sigma_c^2(t))$
 9: **if** $I(x, y) * L_p(x, y) \geq T$
 10: $B(x, y) \rightarrow 1$
 11: **end if**
 12: **if** $I(x, y) * L_p(x, y) < T$
 13: $B(x, y) \rightarrow 0$
 14: **end if**

After thresholding, the insignificant pixels from the binary image are removed with 3x3 ones arrangement erode component. The binarized image is shown in figure 2(b).

Algorithm 3. Erode (B, S)

Input: B (Binary Image), SC (Structuring Component)
Output: B_E (Erosion Transformed Image)
 1: $B_E \rightarrow B \ominus SC = \{c \mid c + s \in I \forall s \in S\}$

Edge Tracing

After the image thresholding phase, the contour of the hand shape is extracted by using the edge tracing algorithm. The palm edge is acquired by using the Canny edge tracking algorithm (Canny, 1987). The achieved edge image is shown in figure 2(c).

Algorithm 4. Canny (B_E, T_H, T_L)

Input: B_E (Erosion Transformed Image), T_H (Higher Threshold), T_L (Lower Threshold).
Output: E (Edge Image)

$$1: P \rightarrow \frac{1}{159} \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix}$$

$$2: S_x \rightarrow \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}$$

$$3: S_y \rightarrow \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix}$$

4: Apply S_x and S_y in x and y-direction of B_e .

5: Gradient Strength $G \rightarrow \sqrt{S_x^2 + S_y^2}$

6: Gradient Direction $\theta \rightarrow \arctan\left(\frac{S_y}{S_x}\right)$

7: Direction is rounded to one of the four possible directions 0° , 45° , 90° , and 135° .

8: From each position (x,y) step in the two directions perpendicular to edge orientation.

9: Denote the initial pixel (x,y) by A, the two neighboring pixels in the perpendicular directions by B and C.

10: **if** $G(B) > G(A) \ || \ G(C) > G(A)$

11: $G(x,y) \rightarrow 0$

12: **end if**

13: **if** $G > T_H$

14: $E \rightarrow 1$

15: **end if**

16: **if** $G < T_L$

17: $E \rightarrow 0$

18: **end if**

19: **if** $T_H > G > T_L$

20: $E \rightarrow 1$ if and only if the pixel is connected to a pixel that is above the upper threshold.

21: **end if**

Corner Finding

Harris corner finder (Dey et.al. 2012; Dey et.al. 2011) is used to extract corner data from the palmprint edge image. The purpose behind selecting this technique is its low mathematical complication, reliability and invariant to image move, illumination, and orientation. Harris corner finder determined by the autocorrelation of image gradient values. The gradient covariance matrix is specified by:

$$HC_{x,y} = \begin{bmatrix} \left(\frac{\partial I}{\partial x}\right)^2 & \left(\frac{\partial I}{\partial x} \frac{\partial I}{\partial y}\right) \\ \left(\frac{\partial I}{\partial x} \frac{\partial I}{\partial y}\right) & \left(\frac{\partial I}{\partial y}\right)^2 \end{bmatrix} = \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \quad (4)$$

where I_x and I_y specify the image gradient in the x and y directions. This technique considers the lowermost and utmost eigenvalues of the image gradient covariance matrix $HC_{x,y}$ in generating corner finder. A ‘corner’ is acquired when the two eigenvalues are alike in magnitude and large. The corners are presented in figure 2(c).

Algorithm 5. Harries (E)

Input: E (Edge Image)

Output: (P_x, P_y) (x and y coordinate of Corners)

- 1: $HC_{x, y} = \begin{bmatrix} E_x^2 & E_x E_y \\ E_x E_y & E_y^2 \end{bmatrix}$
- 2: $\lambda_1, \lambda_2 \rightarrow$ Eigen Values of HC
- 3: $\det HC \cdot \lambda_1 \lambda_2$
- 4: $trace HC \cdot \lambda_1 + \lambda_2$
- 5: $P \rightarrow \det HC - k(trace HC)^2$
- 6: **if** $P > 0$
- 7: [P_x, P_y] \rightarrow Corner pixel
- 8: **end if**

Orientation Correction

After corner detection, to make the image parallel with the horizontal plane the orientation angle of the palm image is standardized. On every image, the rightmost corners are used as the reference points. By joining the bottommost reference point, B and the topmost reference point, A, the line segment (reference line) AB is created. See figure 2(d). Once the reference line is obtained, the proposed calculation normalizes the orientation of the palmprint image by rotating the image with an angle (φ) so that the reference line turn into parallel with the vertical plane (Bertoline et.al. 2003):

$$\varphi = \cot^{-1} \left(\frac{y_B - y_A}{x_B - x_A} \right) \quad (5)$$

where $(x_B - x_A)$ and $(y_B - y_A)$ are the coordinates of the reference point A and B.

Algorithm 6. Orientation (I)

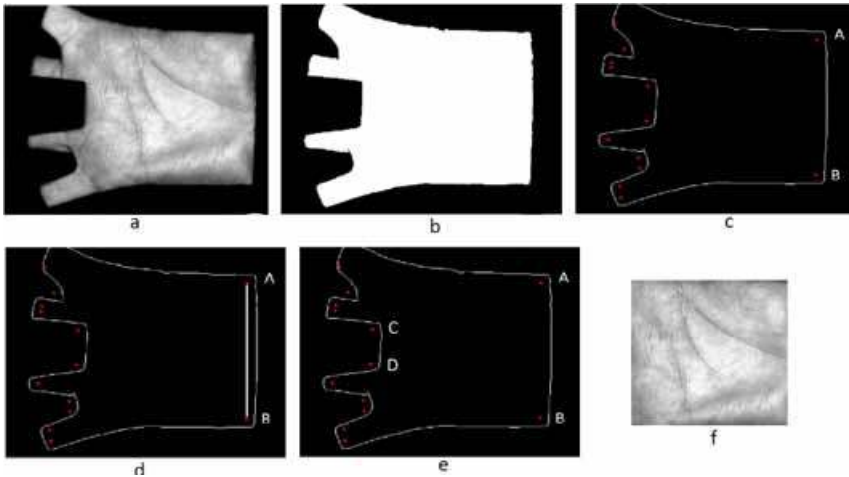
Input: I(x, y) (RGB Image)

Output: I_R (Image after orientation correction)

- 1: $\varphi \leftarrow \cot^{-1} \left(\frac{y_B - y_A}{x_B - x_A} \right)$
- 2: $I_R \leftarrow \text{rotate}(I, \varphi)$ ROI Location

After fixing the orientation angle the ROI is extracted. As the palm images are positioned parallel with the horizontal plane, the co-ordinate of the furthest right 2 corners (A, B) are kept as two corners of the ROI. Passing through from right to left (from A, B), the former two coordinates (C, D) which has at least 20 pixel separation in the vertical direction are kept as additional two corner location of ROI. See figure 2(e). Lastly, the image inside the 4 coordinates (A, B, C, D) are cropped and kept as the ROI. See figure 2(f).

Figure 2. (a): original image, (b): binary image, (c) extracted palm boundary and corners, (d): linesegment drawn between the rightmost corners (A and B) to correct the orientation angle, (e) another two points (C and D) are identified, (f): ROI between A, B, C and D



Algorithm 7. ROI(I_R)

Input: $I_R(x, y)$ (RGB Image)
Output: I_{ROI} (Extracted ROI)

- 1: $P_{r1} \rightarrow P_x(\text{length}(P_x) + 1, 1)$ // P_x is obtained from Harries corner detection function
- 2: $P_{c1} \rightarrow P_y(\text{length}(P_y) + 1, 1)$ // C_y is obtained from Harries corner detection function
- 3: $P_{r2} \rightarrow P_x(\text{length}(P_x), 1)$
- 4: $P_{c2} \rightarrow P_y(\text{length}(P_y), 1)$
- 5: **for** $w \rightarrow 1$ **to** $\text{length}(P_y) - 1$
- 6: $s1(w, 1) \rightarrow \text{abs}(P_y(w, 1) - P_y(w+1, 1))$
- 7: **if** $s1(w, 1) < 10$
- 8: $s2(w, 1) \rightarrow \text{abs}(P_x(w, 1) - P_x(w+1, 1))$
- 9: **end if**
- 10: **end for**


```

11: for  $h \leftarrow \text{length}(s2) - 1$  to 1
12: if  $s2(h, 1) > 20$ 
13:  $P_{r3} \leftarrow P_x(h+1, 1)$ 
14:  $P_{c3} \leftarrow P_y(h+1, 1)$ 
15:  $P_{r4} \leftarrow P_x(h, 1)$ 
16:  $P_{c4} \leftarrow P_y(h, 1)$ 
17: break
18: end if
19: end for
20:  $P_c \leftarrow [P_{c1} P_{c2} P_{c3} P_{c4}]$ 
21:  $Y_{MIN} \leftarrow \text{min}(P_c)$ 
22:  $Y_{MAX} \leftarrow \text{max}(P_c)$ 
23:  $\text{mask} \leftarrow \text{poly2mask}([P_{r4} P_{r2} P_{r1} P_{r3}], [Y_{MAX} Y_{MAX} Y_{MIN} Y_{MIN}], \text{size}(I_R, 1), \text{size}(I_R, 2))$ 
24:  $I_{ROI} \leftarrow I_R * \text{mask}$  // * is the convolution operator
    
```

Algorithm 8. poly2mask (a, b, c, d)

Input: a, b (Vectors to represent polygon vertices); c, d (Size of the binary image)

Output: Mask (Binary region of interest)

- 1: Each unit square is divided into a 5-by-5 grid.
- 2: The vertices positions are altered to be on the intersections of the subpixel grid.
- 3: A path is drawn from each adjusted vertex to the next, following the edges of the subpixel grid.
- 4: The polygon inside pixels are identified to create the Mask using this rule: the pixel is considered as an inside pixel if and only if the pixel's central subpixel is inside the boundaries defined by the path between adjusted vertices.

Pre-Processing

To convert the contrast of the ROI even and to make it translation invariant the pre-processing phase is needed. To standardize the translation issues, the background is cropped until the ROI just fits into its bounding box. After that to standardize the contrast of the ROI image, the lowest intensity of the image is fixed to 0 and the upper intensity is fixed to 255. Values lower than 0 are fixed to 0 and values nearby 255 are fixed to 255.

Algorithm 9. Background_Shrunk (I)

Input: RGB Image (I)

Output: RGB Image (B)

- 1: $[r, c] \leftarrow \text{size}(I)$
- 2: **for** $q \leftarrow 1$ **to** r
- 3: **if** $I(q, :) == 0$
- 4: **continue**
- 5: **else**
- 6: $\text{topmost} \leftarrow q$
- 7: **break**

```
8: end if
9: end for
10: for  $q \leftarrow 1$  to  $c$ 
11: if  $I(:, q) == 0$ 
12: continue
13: else
14:  $leftmost \leftarrow q$ 
15: break
16: end if
17: end for
18: for  $q \leftarrow c$  to  $1$ 
19: if  $I(:, q) == 0$ 
20: continue
21: else
22:  $rightmost \leftarrow q$ 
23: break
24: end if
25: end for
26: for  $q \leftarrow r$  to  $1$ 
27: if  $I(q, :) == 0$ 
28: continue
29: else
30:  $bottommost \leftarrow q$ 
31: break
32: end if
33: end for
34:  $B \leftarrow I$  ( $topmost: bottommost, leftmost: rightmost$ )
```

Algorithm 10. Contrast_Norm (I)

Input: I (RGB Image)
Output: I_c (Contrast normalized image)
1: $p \leftarrow \min(I)$
2: $q \leftarrow \max(I)$
3: $I_c \leftarrow (I - p) * (255 / (q - p))$

Feature Extraction

Principal line extraction, as well as the extraction of features using a 2-component partition method from the principal line, is done in this stage.

Principal Line Extraction

After pre-processing the principal lines are separated from the ROI. To extract the Principal line, first, the Hessian matrix is computed from the ROI. Second order partial derivatives are processed for each image pixel to build the Hessian matrix as shown in equation (6) (Hiriart-Urruty et.al. 1984):

$$H(x, y) = \begin{bmatrix} S_{xx}(x, y) & S_{xy}(x, y) \\ S_{xy}(x, y) & S_{yy}(x, y) \end{bmatrix}$$

$$\begin{aligned}
 S_{xx}(x, y) &= G_{xx}(x, y) * I(x, y) \\
 S_{yy}(x, y) &= G_{yy}(x, y) * I(x, y) \\
 S_{xy}(x, y) &= G_{xy}(x, y) * I(x, y)
 \end{aligned} \tag{6}$$

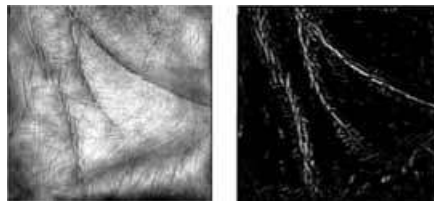
where * is the convolution operator. From the image $I(x, y)$, the 2nd order derivative computed via Gaussian function $G(x, y)$ in x and y-direction with a low value of the smoothing factor σ (Sigma) and represented as S_{xx} , S_{xy} and S_{yy} . The Gaussian function can be represented as:

$$G(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \tag{7}$$

where * is the convolution operator. From the image $I(x, y)$, the 2nd order derivative computed via Gaussian function $G(x, y)$ in x and y-direction with a low value of the smoothing factor σ (Sigma) and represented as S_{xx} , S_{xy} and S_{yy} . The Gaussian function can be represented as:

Then from the Hessian matrix the two Eigenvalues (λ_1 and λ_2) are computed at every pixel. The computed Eigenvalues are real as the Hessian matrix is symmetric. To identify the principal lines, areas have to be found where $\lambda_2 \gg \lambda_1$. Here λ_1 and λ_2 represents the minor and major Eigenvalue. After having the principal lines the image is transformed to binary. See figure 3.

Figure 3. (left): Extracted ROI, (right): Extracted principal lines using the proposed method



Algorithm 11. Principal_Line (I)

Input: I (RGB Image)

Output: I_{PL} (Image Containing Principal Line)

1: $S_{xx}(x, y) \leftarrow G_{xx}(x, y) * I(x, y)$

2: $S_{xy}(x, y) \leftarrow G_{xy}(x, y) * I(x, y)$

3: $S_{yy}(x, y) \leftarrow G_{yy}(x, y) * I(x, y)$

4: $HES \leftarrow \begin{bmatrix} S_{xx}(x, y) & S_{xy}(x, y) \\ S_{xy}(x, y) & S_{yy}(x, y) \end{bmatrix}$

5: $[\lambda_2, \lambda_1] \leftarrow \text{eigen}(HES)$

6: **if** $\lambda_2 \gg \lambda_1$

7: $I_{PL}(x, y) \leftarrow 1$

8: **else**

9: $I_{PL}(x, y) \rightarrow 0$
10: **end if**

2-Component Partition Method

The principal line image is separated into two edge components (comparative and non-comparative) by using the image gradient magnitude. From the training image, the gradient magnitude is calculated using the Sobel operator. The gradient magnitude at coordinate (x,y) of the training image $I_{tr}(x, y)$ is represented by $G_{tr}(x, y)$ and the gradient magnitude of the testing image $I_{ts}(x, y)$ as $G_{ts}(x, y)$. The pixel separation is done according to the following guidelines:

- R1:** If $G_{tr}(x, y) > \text{mean}(G_{ts})$ and $G_{ts}(x, y) > \text{mean}(G_{ts})$, then the pixel is denoted as a comparative edge pixel and 1 is allotted to the resulting feature matrix $I_{C1}(x, y)$, else 0.
- R2:** If $I_{tr}(x, y) > \text{mean}(G_{ts})$ and $I_{ts}(i, j) \leq \text{mean}(G_{ts})$ or $I_{ts}(x, y) > \text{mean}(G_{ts})$ and $I_{tr}(x, y) \leq \text{mean}(G_{ts})$ then the pixel is denoted as a non-comparative edge pixel and 1 is allotted to the resulting feature matrix $I_{C2}(x, y)$, else 0.

Algorithm 12. 2-component_partition (I_{tr}, I_{ts})

Input: I_{tr} (Training Image), I_{ts} (Testing Image)

Output: I_{C1} (Image containing comparative edges), I_{C2} (Image containing non-comparative edges)

1: $U_{tsx} \rightarrow \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} * I_{ts}$ // x- direction gradient calculation of

testing image using Sobel operator

2: $U_{tsy} \rightarrow \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & +1 \end{bmatrix} * I_{ts}$ // y- direction gradient calculation of

testing image using Sobel operator

3: $G_{ts}(x, y) \rightarrow \sqrt{U_{tsx}^2 + U_{tsy}^2}$ // gradient magnitude of the testing image

4: $U_{trx} \rightarrow \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} * I_{tr}$ // x- direction gradient calculation of

training image using Sobel operator

5: $U_{try} \rightarrow \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & +1 \end{bmatrix} * I_{tr}$ // y -direction gradient calculation of

training image using Sobel operator

6: $G_{tr}(x, y) \rightarrow \sqrt{U_{trx}^2 + U_{try}^2}$

7: **if** $G_{tr}(x, y) > \text{mean}(G_{ts})$ **&&** $G_{ts}(x, y) > \text{mean}(G_{ts})$

8: $I_{C1}(x, y) \rightarrow 1$

```

9: else
10:  $I_{C1}(x, y) \rightarrow 0$ 
11: end if
12: if ( $I_{tr}(x, y) > \text{mean}(G_{ts})$ ) && ( $I_{ts}(x, y) \leq \text{mean}(G_{ts})$ ) || ( $I_{ts}(x, y) > \text{mean}(G_{ts})$ ) &&
 $I_{tr}(x, y) \leq \text{mean}(G_{ts})$ 
13:  $I_{C2}(x, y) \rightarrow 1$ 
14: else
15:  $I_{C2}(x, y) \rightarrow 0$ 
16: end if
    
```

Feature Matching

After calculating the feature matrix I_{C1} and I_{C2} , the average of both the matrices is computed and kept in the matrix $I_{SC1}(x, y)$ and $I_{SC2}(x, y)$. Then in $I_{SC1}(x, y)$, if the value calculated from the same training and testing class image is highest as compared to the value computed from different training and testing class images, then 1 is allotted to the 1st cell of the final feature matrix I_F . Similarly, in $I_{SC2}(x, y)$, if the value calculated from the same training and testing classes is lowest or highest as compared to the value computed from different training and testing class images, then 1 is allocated to the 1st cell of the final feature matrix I_F . Here the highest value is also considered as if there is a shift in the testing image, then most of the edges will be converted to non-comparative. If both of the cells of I_F contains 1, then the test image is considered as the right classification.

Algorithm 13. Feature_Matching (I_{C1} , I_{C2} , N , NT)

```

Input:  $I_{C1}$  (Image containing comparative edges),  $I_{C2}$  (Image
containing non-comparative edges),  $N$  (Total number of class),  $NT$ 
(Total number of the testing image)
Output: CLS (Class Name)
1:  $I_{SC1} \rightarrow \text{mean}(\text{mean}(I_{C1}))$ 
2:  $I_{SC2} \rightarrow \text{mean}(\text{mean}(I_{C2}))$ 
3: for  $k \rightarrow 1$  to  $NT$ 
4: for  $j \rightarrow 1$  to  $N$ 
5: if  $I_{SC1}(k, 1) == \text{max}(I_{SC1}(k, j))$ 
6:  $I_F(k, 1) \rightarrow 1$ 
7: end if
8: end for
9: end for
10: for  $k \rightarrow 1$  to  $NT$ 
11: for  $j \rightarrow 1$  to  $N$ 
12: if ( $I_{SC2}(k, 1) == \text{max}(I_{SC2}(k, j))$ ) || ( $I_{SC2}(k, 1) == \text{min}(I_{SC2}(k, j))$ )
13:  $I_F(k, 2) \rightarrow 1$ 
14: end if
15: end for
16: end for
17: for  $k \rightarrow 1$  to  $NT$ 
18: if  $\text{sum}(I_F(k, :)) == 2$ 
    
```

```
19: Accepted class (CLS)  
20: end if  
21: end for
```

EXPERIMENTATIONS AND RESULTS

Dataset

Experimentation is done using 800 high-resolution palm images divided into 40 classes, collected from Hong Kong Polytechnic University (PolyU) palmprint database. Out of 20 images of each class, 10 are considered for training and 10 for testing. Figure 4 demonstrates one image of each class in the data set.

ROI Extraction

To ensure the performance of the proposed palmprint ROI extraction algorithm some experimentation is done as presented in figure 5 as well as figure 6. Figure 5 designates the ROI extraction outcome using two unlike left palmprint images and figure 6 shows the ROI extraction outcome on two different right palmprint images.

Pre-Processing

After ROI extraction the pre-processing step contains the improvement of contrast and standardization of translation factor. Figure 7 represents the outcome of the ROI after pre-processing.

Feature Extraction

This stage consists of the principal line extraction from ROI by applying the proposed method. Then 2-component partition method is used to extract features from principal lines.

Principal Line Extraction

The proposed method is used to extract the principal lines from the palm ROI. Furthermore, a comparison is done between the canny filter and proposed a method to detect the principal lines from the palm ROI. The canny filter is the most commonly used gradient approach in light of the Sobel

Figure 4. Samples of 40 classes of the dataset

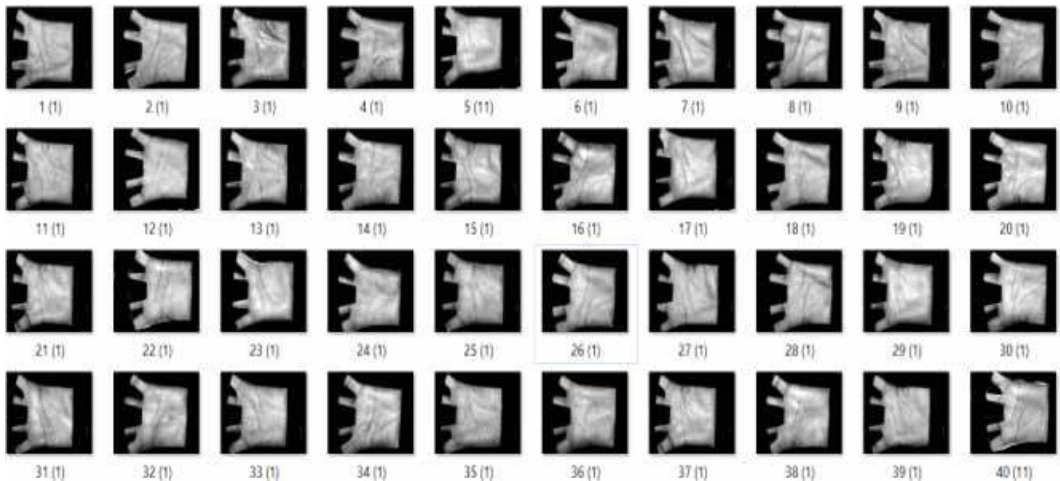


Figure 5. The ROI extraction result (a2, b2) by applying the proposed algorithm to two different left palmprint (a1, b1)

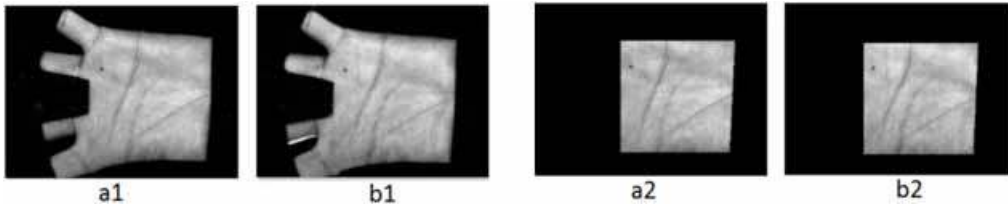


Figure 6. The ROI extraction result (a2, b2) by applying the proposed algorithm to two different right palmprint (a1, b1)

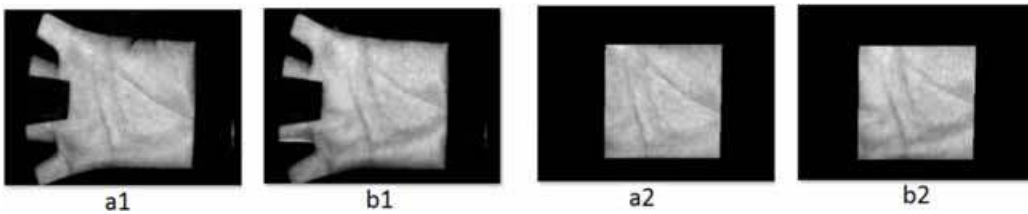
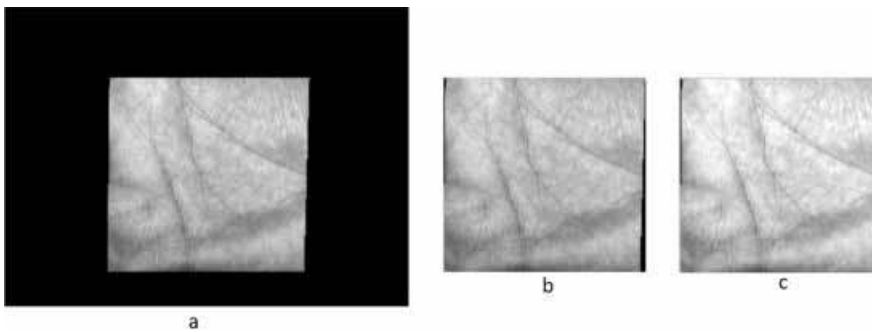
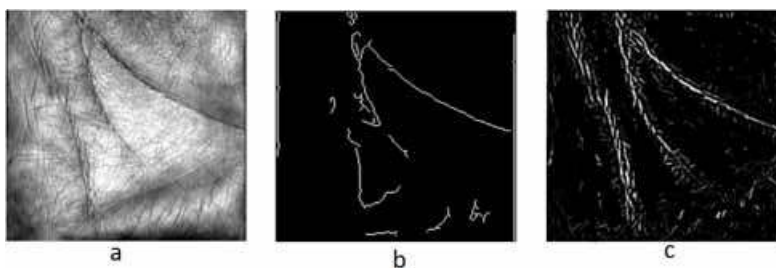


Figure 7. (a): Original ROI, (b): After standardizing the translation factor, (c): After improving the contrast



filter. This filter uses a hysteresis thresholding that helps to determine two perfect thresholds for thorough extraction of principal lines from the ROI. However, the limit points from the first decided might never again be perfect and can cause false identification. Figure 8 displays the comparison of principal lines using a Canny filter and the proposed method.

Figure 8. (a): Extracted ROI, (b): Extracted principal lines using a Canny filter, (c): Extracted principal lines using the proposed method



Values of σ (Gaussian smoothing factor) varying from 1 to 5 are used to extract the lines by using the proposed technique. Figure 9 designates the output using $\sigma = 1, 2$ and 3.

From figure 9 it is clear that at $\sigma = 1$, the data of the principal lines are much better-quality as compared to other σ values. So, in this experimentation, the value of σ is considered as 1.

2-Component Partition Method

The extracted principal line image is divided into 2-components by using this technique: comparative and non-comparative by comparing the training and testing images. Figure 10 and 11 shows the output of this step (I_{C1} and I_{C2}).

Feature Matching and Classification

Table 1 indicates the results using this proposed method. Overall precision is 90.25%. Figure 12 shows the accuracy plot (summation of two cells of I_{SM}) of a test sample of Class 1.

ANALYSIS

The present work is contrasted with other contemporary palmprint recognition methods to the current dataset to watch their exhibitions. In this paper, the methods used for comparison are the Discrete Cosine Transform (DCT), Zernike Moment, Gray Level Co-occurrence Matrix (GLCM), Local Binary Pattern (LBP) and Hamming distance. The time required for the execution of the proposed method is 310 seconds with Intel Core i5 processor and 2GB RAM.

Figure 9. Extracted Principal lines using Proposed method (a): $\sigma = 1$, (b): $\sigma = 2$, (c): $\sigma = 3$

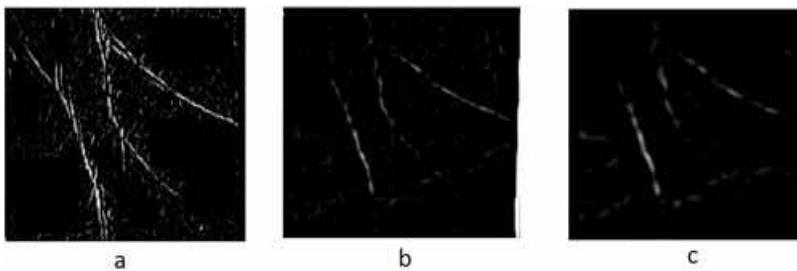


Figure 10. S1: Extracted Testing image Principal line of Class 1; T1, T2, T3: Extracted Training image Principal Line of Class 1, 2 and 3; T1_S1, T2_S1, T3_S1: I_{R1} output computed between the training and testing images

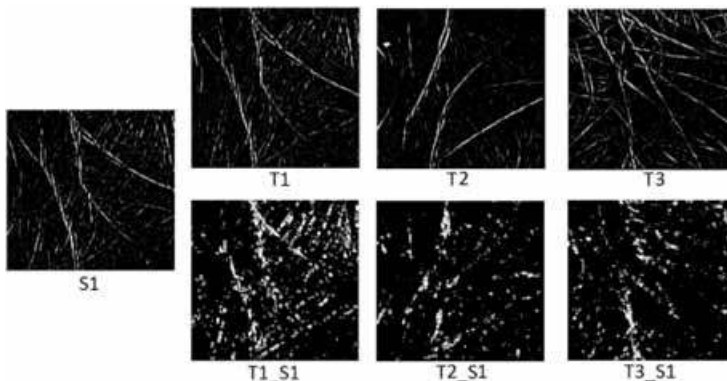


Figure 11. I_{R2} output computed between the training and testing images. T1_S1 (MS): when there is a minimum shift between the training and testing of the same class; T1_S1(S): when there is a maximum shift between the training and testing of same class

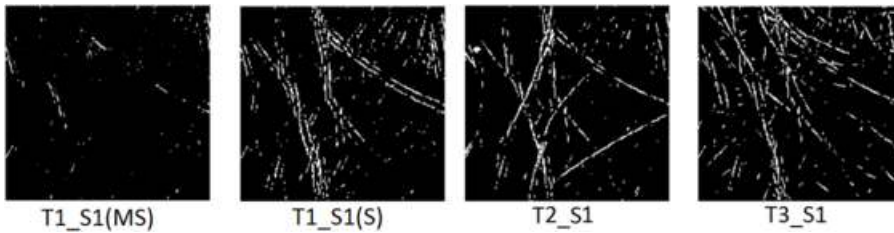
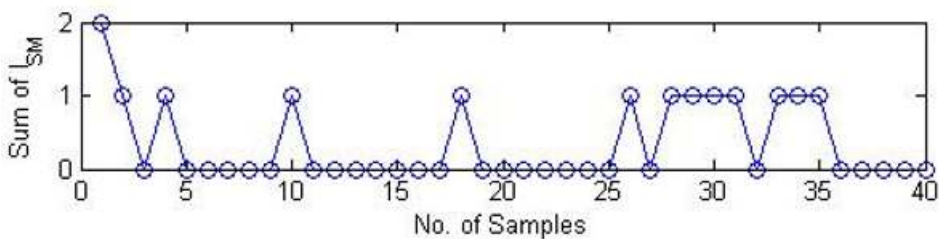


Table 1. Percent Recognition Rates of palmprint testing images

Class	Accuracy	Class	Accuracy	Class	Accuracy	Class	Accuracy
1	90	11	80	21	90	31	100
2	100	12	90	22	90	32	90
3	80	13	90	23	90	33	100
4	80	14	90	24	100	34	80
5	90	15	90	25	90	35	70
6	100	16	100	26	90	36	90
7	90	17	90	27	100	37	90
8	90	18	100	28	100	38	90
9	90	19	100	29	90	39	60
10	90	20	90	30	90	40	90

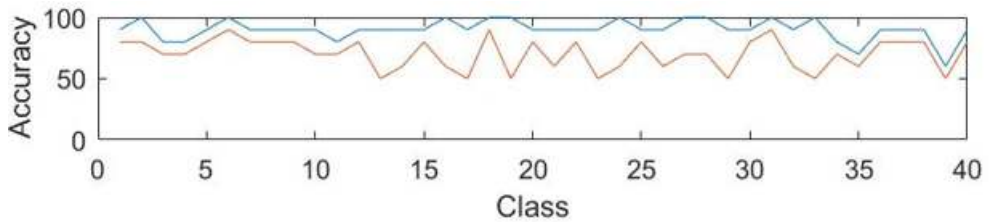
Figure 12. Accuracy plot of a sample of Class 1



Guo et.al. (2009) uses DCT for the recognition of palmprint images. While the input from preprocessed 8×8 pieces is the whole number, the output values are typically real-valued. So a quantization step is required to settle on a few choices about the qualities in each DCT block and create output that is integer-valued. Thus high computational time is required. Obtained overall accuracy is 70.25% using the method used by Guo et.al. The time required for the method used by Guo et.al. is 350 seconds. Figure 13 analyzes class exactness of the approach used by Guo et.al. (red) with the proposed approach (blue).

Zernike moment is used by Karar et.al. (2012) for the recognition of palmprint images. The major drawback with Zernike moment: they have to standardize an image to accomplish scale invariance. This presents a few mistakes since it includes the re-examining and re-evaluating of digital images,

Figure 13. Accuracy using the approach used by Guo et.al. (red) and proposed approach (blue)



and prompts the incorrectness of the classifier. The drawback is that it is numerically complicated and computationally heavy. Obtained overall accuracy is 81% using the method used by Karar et.al. The time required for the method used by Karar et.al. is 300 seconds. Figure 14 analyzes class exactness of the approach used by Karar et.al. (red) with the proposed approach (blue).

Zhu et.al. (2011) used GLCM features for the recognition of palmprint images. The major drawback of the use of GLCM to calculate texture features for image segmentation is that it is very intensive computationally. Although, most of the calculations are done over unnecessary zero probabilities. Obtained overall accuracy is 79% using the method used by Jun et.al. The time required for the method used by Zhu et.al. is 360 seconds. Figure 15 analyzes class exactness of the approach used by Jun et.al. (red) with the proposed approach (blue).

Figure 14. Accuracy using the approach used by Karar et.al. (red) and proposed approach (blue)

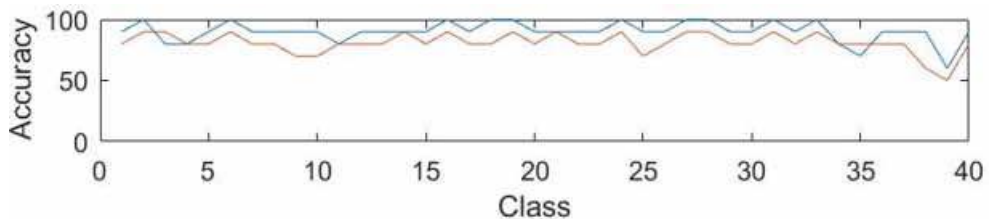
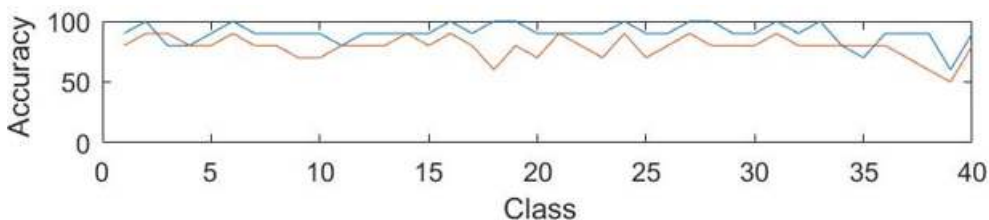


Figure 15. Accuracy using the approach used by Jun et.al. (red) and proposed approach (blue)



Local Binary Pattern is used by Guo et.al. (2010) for the recognition of palmprint images. The local binary pattern (LBP) operator have three disadvantages: (1) They create rather long histograms, which back off the recognition speed particularly on vast scale database; (2) Under some specific situation, they miss the nearby structure as they don't consider the impact of the middle pixel; (3) The binary data delivered by them are delicate to noise. Obtained overall accuracy is 67% using the

method used by Guo et.al. The time required for the method used by Guo et.al. is 340 seconds. Figure 16 analyzes class exactness of the approach used by Guo et.al. (red) with the proposed approach (blue).

Hamming distance is used by Kumar et.al. (2014) for the recognition of palmprint images. The main drawback of this method is, it is very sensitive to the applied threshold value. Also, this method is very restrictive when making comparisons. Obtained overall accuracy is 42.5% using the method used by Kumar et.al. The time required for the method used by Kumar et.al. is 370 seconds. Figure 17 analyzes class exactness of the approach used by Kumar et.al. (red) with the proposed approach (blue).

Figure 16. Accuracy using the approach used by Guo et.al. (red) and proposed approach (blue)

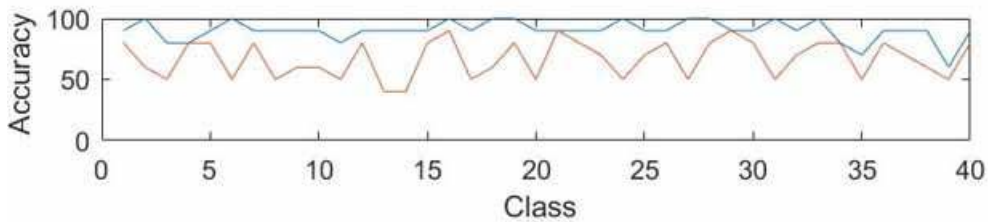
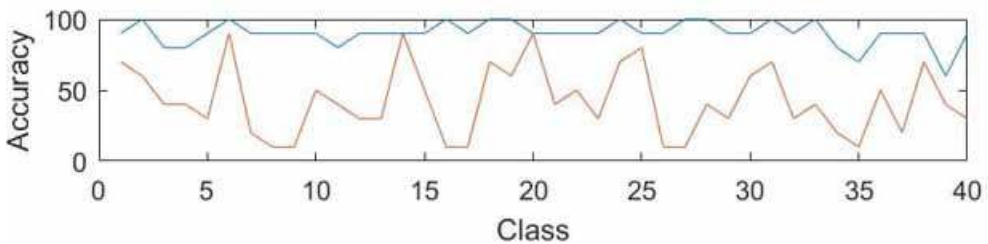


Figure 17. Accuracy using the approach used by Kumar et.al. (red) and proposed approach (blue)



CONCLUSION

This article proposes a new technique for recognition of palmprint images. This palmprint recognition technique incorporates ROI extraction, pre-processing, feature extraction and feature matching. ROI extraction is considered as one of the imperative stages in these four phases since it incredibly impacts the general recognition accuracy and handling rate of the entire framework. It is necessary to take ROI at a similar position from different palmprint images to guarantee the quality of the recognition of palmprint images. In this paper, a new ROI extraction method is suggested based on corner identification. Corner identification is done by Harries corner detection algorithm. Feature extraction method includes two-steps: Principal line extraction which is based on the Eigenvalues computed from the Hessian matrix and 2-component partition method which is used to segment the principal lines of any test image in similar and non-similar edges. At last the feature matching is carried out relying upon the score of similar and non-similar edges. Future scope includes the detection of palmprint image with low resolution and different lighting conditions.

REFERENCES

- Bounneche, M. D., Boubchir, L., Bouridane, A., Nekhoul, B., & Ali-Chérif, A. (2016). Multi-spectral palmprint recognition based on oriented multiscale log-Gabor filters. *Neurocomputing*, 205, 274–286. doi:10.1016/j.neucom.2016.05.005
- Bertoline, G. R., Wiebe, E. N., Miller, C. L., & Nasman, L. O. (2003). *Technical graphics communication*. McGraw-Hill.
- Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-8*(6), 679–698. doi:10.1109/TPAMI.1986.4767851 PMID:21869365
- Dey, M., Dey, N., Mahata, S. K., Chakraborty, S., Acharjee, S., & Das, A. (2014, January). Electrocardiogram feature based inter-human biometric authentication system. In *Electronic Systems, Signal Processing and Computing Technologies (ICESC), 2014 International Conference on* (pp. 300-304). IEEE. doi:10.1109/ICESC.2014.57
- Dey, N., Das, S., & Rakshit, P. (2011). A novel approach of obtaining features using wavelet based image fusion and Harris corner detection. *Int J Mod Eng Res*, 1(2), 396–399.
- Dey, N., Roy, A. B., & Das, A. (2012, August). Detection and measurement of bimalleolar fractures using Harris corner. In *Proceedings of the International Conference on Advances in Computing, Communications and Informatics* (pp. 45-51). ACM. doi:10.1145/2345396.2345405
- Dey, N., Nandi, P., Barman, N., Das, D., & Chakraborty, S. (2012). *A comparative study between Moravec and Harris corner detection of noisy images using adaptive wavelet thresholding technique*. arXiv preprint arXiv:1209.1558
- Fei, L., Zhang, B., Xu, Y., & Yan, L. (2016). Palmprint recognition using neighboring direction indicator. *IEEE Transactions on Human-Machine Systems*, 46(6), 787–798. doi:10.1109/THMS.2016.2586474
- Fei, L., Xu, Y., & Zhang, D. (2016). Half-orientation extraction of palmprint features. *Pattern Recognition Letters*, 69, 35–41. doi:10.1016/j.patrec.2015.10.003
- Feng, J., Wang, H., Li, Y., & Liu, F. (2015, November). Palmprint feature extraction method based on rotation-invariance. In *Chinese Conference on Biometric Recognition* (pp. 215-223). Springer. doi:10.1007/978-3-319-25417-3_26
- George, A., Karthick, G., & Harikumar, R. (2014, March). An efficient system for palm print recognition using ridges. In *Intelligent Computing Applications (ICICA), 2014 International Conference on* (pp. 249-253). IEEE. doi:10.1109/ICICA.2014.60
- Guo, Z., Zhang, L., Zhang, D., & Mou, X. (2010, September). Hierarchical multiscale LBP for face and palmprint recognition. In *Image Processing (ICIP), 2010 17th IEEE International Conference on* (pp. 4521-4524). IEEE. doi:10.1109/ICIP.2010.5653119
- Guo, Z., Zhang, D., Zhang, L., & Zuo, W. (2009). Palmprint verification using binary orientation co-occurrence vector. *Pattern Recognition Letters*, 30(13), 1219–1227. doi:10.1016/j.patrec.2009.05.010
- Hiriart-Urruty, J. B., Strodhot, J. J., & Nguyen, V. H. (1984). Generalized Hessian matrix and second-order optimality conditions for problems with C 1, 1 data. *Applied Mathematics & Optimization*, 11(1), 43–56. doi:10.1007/BF01442169
- Hong, D., Pan, Z., & Wu, X. (2014). Improved differential box counting with multi-scale and multi-direction: A new palmprint recognition method. *Optik (Stuttgart)*, 125(15), 4154–4160. doi:10.1016/j.ijleo.2014.01.093
- Hong, D., Liu, W., Su, J., Pan, Z., & Wang, G. (2015). A novel hierarchical approach for multispectral palmprint recognition. *Neurocomputing*, 151, 511–521. doi:10.1016/j.neucom.2014.09.013
- Karar, S., & Parekh, R. (2012). Palm print recognition using zernike moments. *International Journal of Computers and Applications*, 55(16), 15–19. doi:10.5120/8839-3069
- Kong, A., Zhang, D., & Kamel, M. (2006). Palmprint identification using feature-level fusion. *Pattern Recognition*, 39(3), 478–487. doi:10.1016/j.patcog.2005.08.014

- Kumar, N. B., & Premalatha, K. (2014). Palmprint Authentication System based on local and global feature fusion using DOST. *Journal of Applied Mathematics*, 2014(918376), 1–12. doi:10.1155/2014/918376
- Kumar, R. (2018). *A Robust Biometrics System Using Finger Knuckle Print*. In Handbook of Research on Network Forensics and Analysis Techniques (pp. 416–446). IGI Global.
- Rajeswari, P., Raju, S. V., Ashour, A. S., & Dey, N. (2017). Multi-fingerprint unimodel-based biometric authentication supporting cloud computing. In *Intelligent techniques in signal processing for multimedia security* (pp. 469–485). Springer. doi:10.1007/978-3-319-44790-2_21
- Roy, P., Dutta, S., Dey, N., Dey, G., Chakraborty, S., & Ray, R. (2014, July). Adaptive thresholding: a comparative study. In *Control, Instrumentation, Communication and Computational Technologies (ICCICCT), 2014 International Conference on* (pp. 1182–1186). IEEE. doi:10.1109/ICCICCT.2014.6993140
- Rudzki, M. (2009, October). Vessel detection method based on eigenvalues of the hessian matrix and its applicability to airway tree segmentation. In *Proceedings of the 11th International PhD Workshop OWD* (pp. 100–105). Academic Press.
- Sato, Y., Nakajima, S., Atsumi, H., Koller, T., Gerig, G., Yoshida, S., & Kikinis, R. (1997). 3D multi-scale line filter for segmentation and visualization of curvilinear structures in medical images. In *CVRMed-MRCAS'97* (pp. 213–222). Springer. doi:10.1007/BFb0029240
- Shrivastava, G., Pandey, A., & Sharma, K. (2013). Steganography and its technique: Technical overview. In *Proceedings of the Third International Conference on Trends in Information, Telecommunication and Computing* (pp. 615–620). Springer. doi:10.1007/978-1-4614-3363-7_74
- Vala, M. H. J., & Baxi, A. (2013). A review on Otsu image segmentation algorithm. *International Journal of Advanced Research in Computer Engineering & Technology*, 2(2), 387.
- Zhang, D., Kong, W. K., You, J., & Wong, M. (2003). Online palmprint identification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(9), 1041–1050. doi:10.1109/TPAMI.2003.1227981
- Zhang, K., Huang, D., & Zhang, D. (2017). An optimized palmprint recognition approach based on image sharpness. *Pattern Recognition Letters*, 85, 65–71. doi:10.1016/j.patrec.2016.11.014
- Zhu, X., Liu, D., Zhaoshan, Z., Zhang, Q., & Wenhua, L. (2011). Research of the near palmprint classification based on gray level co-occurrence matrix and SVM. *Journal of Computers*, 6(7), 1535–1541.

Jyotismita Chaki is an Asst. Professor in the School of Information Technology and Engineering VIT University, India. She has done her PhD (Engg) from Jadavpur University, Kolkata, India. Her research interests include: Computer Vision and Image Processing, Pattern Recognition, Medical Imaging, Soft computing, Data mining, Machine learning.

Nilanjan Dey is an Assistant Professor in Department of Information Technology at Techno India College of Technology (under Techno India Group), Kolkata, India. He is also a Visiting Professor at Duy Tan University, Vietnam, He was an honorary Visiting Scientist at Global Biomedical Technologies Inc., CA, USA (2012-2015), Research Scientist of Laboratory of Applied Mathematical Modeling in Human Physiology, Territorial Organization of- Scientific and Engineering Unions, Bulgaria. Associate Researcher of Laboratoire RIADI, University of Manouba, Tunisia. He is an associated Member of University of Reading, London, UK and Scientific Member of - Politécnica of Porto. In addition, he was awarded as one among the top 10 most published academics in the field of Computer Science in India during the period of consideration 2015-17 during 'Faculty Research Awards' organized by Careers 360 at New Delhi, India on March 20, 2018. Before he joined Techno India, he was Assistant Professor of JIS College of Engineering and Bengal College of Engineering and Technology. He has authored/edited more than 30 books with Elsevier, Wiley, CRC Press and Springer, and published more than 270 papers. His h-index is 26 with more than 3000 citations. He is the Editor-in-Chief of Int. J. of Ambient Computing and Intelligence (IJACI, IGI Global,UK, Scopus), Int. J. of Rough Sets and Data Analysis (IGI Global,US, DBLP,ACM dl), Co- Editor-in-Chief of Int. J. of Synthetic Emotions (IGI Global,US, DBLP,ACM dl) and International Journal of Natural Computing Research (IGI Global,US, DBLP,ACM dl). He is the Series Co-Editor of Advances in Ubiquitous Sensing Applications for Healthcare (AUSAH), Elsevier, Series Editor of Computational Intelligence in Engineering Problem Solving and Intelligent Signal processing and data analysis, CRC Press (FOCUS/Brief Series) and Advances in Geospatial Technologies (AGT) Book Series, (IGI Global), US, serves as an editorial board member of several international journals, including International Journal of Image Mining (IJIM), Inderscience, Associated Editor of IEEE Access (SCI-Indexed), and International Journal of Information Technology, Springer. His main research interests include Medical Imaging, Machine learning, Computer Aided Diagnosis as well as Data Mining. He has been on program committees of over 50 international conferences, a workshop organizer of 5 workshops, and acted as a program co-chair and/or advisory chair of more than 10 international conferences. He has given more than 40 invited lectures in 10 countries, including many invited plenary/keynote talks at the international conferences such as ITITS2017 (China), TIMEC2017 (Egypt) and SOFA2018 (Romania), BioCom2018 (UK), etc.