

ECH-A NOVEL MULTILEVEL THRESHOLDING TECHNIQUE FOR MINUTIAE BASED FINGERPRINT RECOGNITION

Dattatreya P. Mankame¹, Ravi Subban², P.Pasupathi³, Muthukumar S⁴

¹Associate Professor and Head, Department of Information Science and Engineering, KLE Institute of Technology, Opposite to Airport, Gokul, Hubli- 580 030, Karnataka, India

²Assistant Professor, Department of Computer Science, School of Engineering and Technology, Pondicherry University, R.V.Nagar, Kalapet, Puducherry – 605 014, Pondicherry, India

³Assistant Professor, Centre for Information Technology and Engineering, MS University, Tirunelveli, India

⁴Assistant Professor, Centre for Information Technology and Engineering, MS University, Tirunelveli, India

Abstract

Thresholding is probably the most frequently used technique in segmenting an image. Several computationally efficient thresholding algorithms have been shown to offer threshold value, however whilst many of the methods will give similar guarantees. In this proposed research work, a novel multilevel global thresholding algorithm ECH (pronounced as eh-chch) is introduced. The algorithm uses HAT transformation to maintain uniform intensity in the gray scale images. The optimum threshold value is decided by entropy of the background and foreground regions, where the gray scale samples are clustered as a mixture of two Gaussians for background and foreground objects, using which the peaks and valleys like minutiae points are analyzed. This paper also explores a range of steps implicated in FP recognition. The performances of the algorithm are compared with NEST DB4, DB2 FVC2002, DB2 FVC2004 databases. The performance indices FAR (False Acceptance Ratio) less than 1.3% and FRR (False Rejection Ratio) almost 0% is reported using MATLAB. The experimental results on MATLAB shows prominent result on gray scale finger print images.

Keywords: Thresholding, Entropy, Clustering, Binarization, uneven illumination

1. INTRODUCTON

The main difficulty coupled with discrepancy of gray levels within the object and the background, inadequate contrast. Image thresholding is one of the first image pre-processing image processing techniques used, before image analysis step, to obtain a binary image from its gray scale one. In automated image analysis, finding the optimum value of the threshold is quite vital in view of the fact that it affects the performance of consecutive steps like segmentation, feature extraction and like. The harvest of the thresholding operation is a binary image with gray level of 0 (black) indicating a pixel belonging to a target and gray level of 1 (white) as its background. In spite of the perceptible easiness of the thresholding procedure, it is in reality a complex task. Improper selection of the thresholding value will lead to blotches, streaks, erasures on the converted binary image, which in-turn makes the segmentation and recognition tasks mystifying. The thresholding methods rely on several factors such as the gray level distribution, shading effects etc.

The difficulty with the thresholding algorithms are strict proper matching, time complexity and memory necessities.

All the thresholding algorithms can be classified mainly into two groups. These are:

1. Global thresholding-scheme uses single threshold value for entire image. Here, If 'th' is the threshold, to distinguish two classes (background and foreground) of an original image $f(x,y)$, and any pixel with gray level intensity ' pi ' > this set to 1 (white) else to 0 (black). The thresholded binary image can be given as:

$$f^*(x,y) = \begin{cases} 1, & \text{if } f(x,y) > T \\ 0, & \text{if } f(x,y) \leq T \dots \dots \dots (1) \end{cases}$$

2. Local thresholding-scheme divides the image into sub images and determines the threshold for each sub images. [1][2][3][4]

In the rest of the paper, our discussion is restrained to traditional image thresholding reviews, factors to be considered for thresholding algorithms, proposed new multilevel thresholding technique-ECH, especially for gray scale finger print images, experimental results and discussion.

2. TRADITIONAL IMAGE THRESHOLDING METHODS

There are quite a lot of studies on the performance of the thresholding techniques. Ostu presented an automatic discriminant analysis optimal threshold selection method, based on the histogram of the gray scale image is, not based on the differentiation(i.e. local property such as valley), but based on the integration (i.e. global property such as whorl) of the histogram. It divides the image into two classes (say, C1 for objects and C2 for background) at gray level t .

$$C1=\{1,2,\dots,t\} \text{ and } C2=\{t+1, t+2,\dots, l-1\}$$

Where, l =number of gray levels.

If σ^2W , σ^2B and σ^2T are the variances of within the class, between the classes and total respectively, it calculates the optimum threshold value by minimizing one of the following. Where the variance is measure of region homogeneity(i.e. regions with high homogeneity will have low variance).
 $\lambda = \sigma^2B / \sigma^2W; \eta = \sigma^2B / \sigma^2T; K = \sigma^2T / \sigma^2W; \dots(2)$
 In the above three equations, goodness of the choice for threshold t^* can be estimated as:

$$t^* = \text{ArgMin } \eta \text{ teG} \dots \dots \dots (3)$$

where,

$$\sigma T^2 = \sum_{i=0}^{l-1} (i - \mu T)^2 Pi, \mu T = \sum_{i=0}^{l-1} i Pi,$$

$$\sigma B^2 = \omega_0 \omega_1 (\mu_1 \mu_0)^2, \omega_0 = \sum_{i=0}^t Pi, \omega_0 = 1 - \omega_0,$$

$$\mu_1 = \frac{\mu T - \mu_0}{1 - \omega_0}, \mu_0 = \frac{\mu_0}{\omega_0}, \mu_0 = \sum_{i=0}^t i Pi \dots \dots \dots (4)$$

The drawback of the Ostu's method is that the method breaks down when the two classes are very unequal (i.e. the classes have very different sizes) and does not work well with the images of varying intensity levels.

Zuoyong Li's concept[5] uses difference in standard deviation for calculating threshold value, and finds the optimal threshold by minimizing it. Wang Wen-Cheng's[6] global point dependent technique, which assumes dark objects within light background. By high and mighty that the percentage of the object area is identified, the threshold is computed as the utmost gray level which maps slightest (100p)% of the pixels into the objects in the thresholded image. For example, suppose an object occupies 20% of an image, subsequently the image ought to be thresholded at the utmost gray level that allows at least 20% of the pixels to be mapped into the object. Obviously, this method is not pertinent to images whose object area is not identified. Agus Zaina lArifin [7] proposed a global point dependent gray level histogram scheme; the threshold value corresponds to the valley of the histogram.

This scheme fails for image which has extremely unequal peaks or with broad/ flat valleys.

2.1 Factors Affecting the Selection of the Threshold

The factors that affect the selection of the threshold include:

- Separation between the peaks
- Noise content in the image
- Relative size of the objects and background
- Uniformity of the lighting
- Uniformity of the reflection

Global thresholding methods fail in case of uneven illumination due to gloom or due to the direction of illumination(as in Fig.1)[8]. To surmount this, HAT transformation is applied to the top and bottom regions separately. The fig.1 shows the effect of uneven illumination on thresholding. Taking into account these points, the method recommended in this correspondence may be considered as the good and simple one for threshold selection that can be applied to various practical finger print recognition problems.

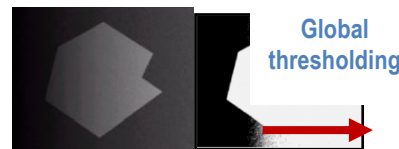


Fig-1: Effect of illumination [8]

3. DESIGN AND IMPLEMENTATION OF FINGERPRINT ALGORITHMS

The fingerprint recognition system contains sensor (like optical or semi-conductor to attain fingerprint), minutiae extractor and minutiae matcher. The detailed design of the overall phases is shown in the Fig. 2.

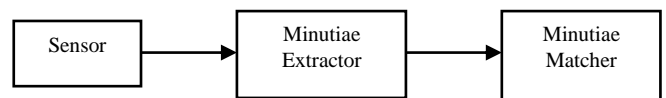


Fig-2:Design of finger print recognition

In view of the fact that, the image captured by the sensor is not perfect, it requires improvement in the quality. This is achieved by means of image pre-processing. The image enhancement increases contrast between the ridges and furrows. It also overcomes from the false broken point of ridges. Our method uses histogram equalization and Fourier transform method for Image enhancement.

The histogram equalization method expands the pixel value distribution of an image so that its perceptual information is increased and it occupies all the ranges from 0 to 255 (as in Fig.3). Thus its visualization effect is enhanced as shown in Fig. 4.

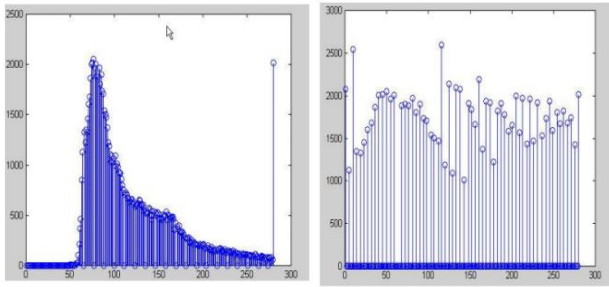


Fig-3:Original Histogram of Fingerprint(left) and Histogram after Histogram Equalization (right)

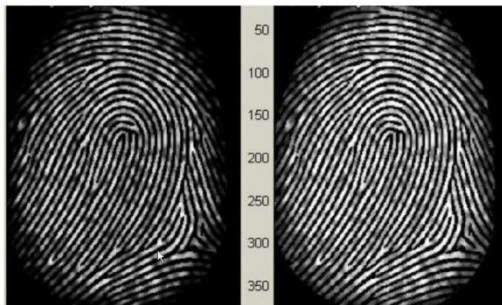


Fig-4:Original Fingerprint (left) and Finger print after Image Enhancement (right)

Fourier transform is applied by separating the image into smaller block of images with a size of 32 by 32 pixels using the formula [9].

$$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) X \exp \left\{ -j2\pi X \left(\frac{ux}{M} + \frac{vy}{N} \right) \right\} \tag{5}$$

Where, u = 0 to 31 and v = 0 to 31.

The enhanced block can be obtained using the equation

$$g(x, y) = F^{-1} \{ F(u, v) X |F(u, v)|^k \} \tag{6}$$

Where, $F^{-1}(F(u, v))$ is obtained by

$$f(x, y) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} F(u, v) X \exp \left\{ j2\pi X \left(\frac{ux}{M} + \frac{vy}{N} \right) \right\} \tag{7}$$

For x= 0 to 31 and y = 0 to 31. K is a constant and can take on value from 0 to 1. In our experiment K is chosen to be 0.45. The higher value of K will improve the appearance of the ridges (as shown in Fig.5) and fills up the small holes. But too high value can cause in phonyfusion of the edges.

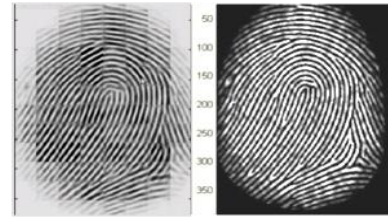


Fig-5:Fingerprint Enhancement using FT (Left) and Original Fingerprint (Right)

The next step is binarization. The binarization is the process of converting grey image into binary image (as in Fig.6), such that it helps in improving the contrast between the ridges and the valleys and in turn extraction of the minutiae. As per MATLAB value 1 is white and value 0 is black pixel. When threshold is applied for an image, all the pixel values are compared with the threshold value and each will be transformed into either 1 or 0 depending on whether the value is greater than or less than threshold respectively. Choice of threshold value is very imperative to retain the data. In our approach, the threshold value is chosen by a novel technique called ECH algorithm. In the proposed method optimum threshold value is decided by entropy of the background and foreground regions, where the gray scale samples are clustered as a mixture of two Gaussians for background and foreground objects, using which the peaks and valleys like minutiae points are analyzed[10].

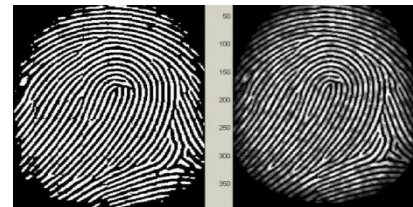


Fig-6:Image after Binarization (left) and Image after enhancement(right)

3.1 ECH Multilevel Thresholding Algorithm:

3.1.1 Algorithm for Uniform Illumination

Input: Loaded the gray scale image from the main program

step1: Open the image to estimate its background

step2: Subtract the image from original image to get image with uniform background.

Step3: Compute the closing of the image minus the image

Output: Uniform illuminated Image

3.1.2 Algorithm for Computing Entropy Threshold Value

Input: Uniform image obtained from previous module
step1: Normalize the histogram so that it sums to 1
step2: Initialize an array for storing entropies
step3: Repeat for white=1 to t and black=(t+1) to 255
 Compute entropy for black and white separately using equations (5 and 6) as given:

$$H_B = \frac{\sum(\text{black})}{(0.001 + \sum(\text{black}))} * \log\left(\frac{\text{black} + 0.001}{(0.001 + \sum(\text{black}))}\right) \dots(8)$$

$$H_W = \frac{\sum(\text{white})}{(0.001 + \sum(\text{white}))} * \log\left(\frac{\text{white} + 0.001}{(0.001 + \sum(\text{white}))}\right) \dots(9)$$

step4: Compute entropies= H_B+H_W
step5: Find entropy threshold= \max value of the entropies
Output: Entropy threshold

3.1.3 Algorithm for Computing Cluster Threshold Value

Input: Entropy threshold obtained from previous module
step1: Let initial threshold be entropy threshold value
step2: Segment the image as background and foreground according to threshold value using [11]
 $G1 = \{f(x, y) : f(x, y) > T\}$
 $G2 = \{f(x, y) : f(x, y) < T\}$
step3: Compute mean of each set i.e. $m1$ and $m2$
step4: Compute new threshold value $T(i+1) = (m1+m2)/2$
step5: Repeat until $|T(i+1) - T| > tol$
 increment i by 1
step6: Find the cluster threshold value
Output: Cluster threshold

3.1.4 Algorithm for Computing Histogram Based Threshold Value

step1: Compute normalized histogram of input image
 $P_i = n(i)/size(m*n) \dots(10)$
step2: Compute cumulative sums $P_1(K)$
 $P1(k) = \sum_{i=0}^k P_i \dots(11)$
 $P2(k) = \sum_{i=k+1}^{L-1} P_i \dots(12)$
 where, Class $C1[0, K]$ and $C2[K+1, L-1]$
step3: Compute cumulative means $m(K)$
 $m1(K) = 1 / p1(K) * \sum_{i=0}^k i * P_i \dots(13)$
 $m2(K) = 1 / p2(K) * \sum_{i=k+1}^{L-1} i * P_i \dots(14)$
step4: Compute Global intensity mean $mG = Pm + Pm$

step5: Compute between class variance
 $(\text{partial})^2 = P1(m1 - mG)^2 + P2(m1 - mG)^2$ for $V=1$ to $L \dots(15)$

step6: Find histogram threshold value

Output: Hist threshold

3.1.5 Algorithm for Converting Grayscale Image to Binary Image

Input: Histogram threshold obtained from the previous module
step1: Using the threshold value convert the grayscale image into binary using `im2bw` function

Output: Binary Image

Segmentation process follows binarization process. The image segmentation task is fulfilled by a three-step approach: block direction estimation, segmentation by direction intensity and Region of Interest (ROI) extraction by morphological operations. The ROI is used to get rid of the image area which does not restrain effective ridges and furrows. Then bound of the residual effective area is traced. In our approach this has been accomplished by means of block direction estimation followed by direction variety check, which is achieved by morphological operation. In block direction estimation, direction of each block (pixel of size $W \times W$) is estimated and the subsequent steps are applied, resulting into image shown in Fig. 7.

Step1: For each block calculate the gradient values along the x-direction(g_x) and y-direction(g_y)

Step2: Find Least Square Approximation for each block using
 $t_g 2\beta = 2\sum\sum(g_x * g_y) / \sum\sum(g_x^2 - g_y^2) \dots(16)$
 for all pixels in each block.

Step3: Find the tangent value of each block direction using
 $t_g 2\theta = 2\sin\theta\cos\theta(\cos^2\theta - \sin^2\theta) \dots(17)$

Step 4: Discard the blocks without significant information of ridges and furrows using
 $E = \{2\sum\sum(g_x * g_y) + \sum\sum(g_x^2 - g_y^2)\} / W * W * \sum\sum(g_x^2 + g_y^2) \dots(18)$

Step5: If certainty level E is less than threshold, then it is treated as background block.

The ROI is extracted using two morphological operations OPEN and CLOSE. The OPEN expands the image and removes the peaks introduced by noise. The CLOSE operation shrinks the images and eliminates the small images(as in Fig.8).

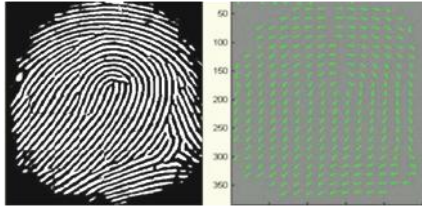


Fig-7:Binarization Fingerprint image (left) and Direction map(right)

The bound area is obtained by subtracting closed area from open area. The algorithm expels uppermost, bottommost, leftmost and rightmost blocks out of bound, so that it contains only bound and inner area.

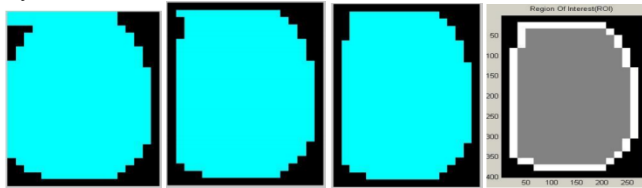


Fig-8:(From left) Original Image, After CLOSE, After OPEN, ROI+bound

3.2 Minutiae Extraction

Minutiae extraction is an important phase in image matching; this is achieved using parallel ridge thinning and minutiae marking. Ridge thinning is a process which eliminates the superfluous pixels of the ridges until the ridge is just one pixel width. The thinning algorithm marks image window 3 X 3 and then removes all marked pixels after scan process. The result is as shown in Fig. 9 [12].



Fig-9:(From left) Original Image, Enhanced Image, Binarized Image, Thinned Image

A Minutiae marking is an easy task. In our approach, concept of Cross Numbering(CN) is used. This method involves the use of skeleton image where the ridge flow pattern eight is connected. The minutiae are extracted by scanning the local neighborhood of each ridge pixel in the image using a 3X3 window. The CN value is then computed, which is defined as half the sum of the differences between pairs of adjacent pixels in the eight neighborhood (as shown in Fig. 10).

If $Cn(P)=1$, it is Ridge End i.e. if central pixel is 1 and has

exactly 3 neighboring 3 one-values.

If $Cn(P)=3$, it is Ridge Bifurcation Point i.e. if central pixel is 1 and has only 1 one-value neighbor.

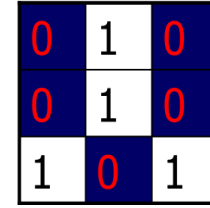


Fig-10:(From Left) Bifurcation, Ridge End (Termination).

3.3 Minutiae Post-Processing:

The final phase of the feature extractor involves removal of false minutiae. The broad-spectrum seven types of the false minutiae are shown in Fig 11.

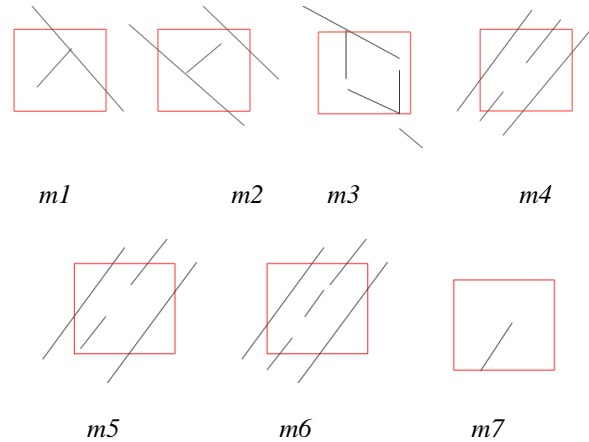


Fig- 11:False Minutiae structure

m1=spike piercing to valley

m2= spike falsely connects 2 ridges

m3= two near bifurcations located in same ridge

m4= two ridge broken points

m5= alike *m4*

m6= extends case *m4*, but ridge found in the middle of the two parts of the broken ridge

m7= short ridge found in the threshold window

Algorithm: To remove false minutiae

step1: If the distance between one bifurcation and one termination is less than *D* and the two minutiae are in the same ridge(*m1*), remove both of them.

step2: If the distance between two bifurcations is less than *D* and they are in the same ridge, remove two bifurcations(*m2, m3*).

step3: If two terminations are within a distance *D* and their directions are coincident with a small angle variation and no any other termination is located between the two

terminations, then they are treated as false minutia and removed ($m4, m5, m6$).

step4: If two terminations are located in a short ridge less than D, remove the two terminations ($m7$).

3.4 Minutia Match:

The final step in the FP recognition is minutia matching. This is achieved by means of alignment-based match algorithm. It is performed using Alignment stage and Match stage.

Algorithm: Alignment stage

step1: Given two fingerprint images to be matched, any one minutia from each image is chosen, and the similarity of the two ridges associated with the two referenced minutia points is calculated.

$$S = \sum_{i=0}^m xiXi / [\sum_{i=0}^m xi^2 Xi^2] ^{0.5} \dots\dots\dots(19)$$

Where, (x_i-x_n) and (X_i-X_n) are series of minutia points for each finger print image. M is the minimal of the n and N value.

If similarity score is more than 0.8 go to step 2.

step2: For each fingerprint, translate and rotate all other minutia with respect to the reference minutia according to the following formula:

$$\begin{bmatrix} xi_new \\ yi_new \\ \theta i_new \end{bmatrix} = TM * \begin{bmatrix} (xi - x) \\ (yi - y) \\ (\theta i_new) \end{bmatrix} \dots\dots\dots(20)$$

where (x,y,θ) are the parameters of the reference minutiae.

$$TM = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \dots\dots\dots(21)$$

The effect of rotation can be shown in Fig. 12.

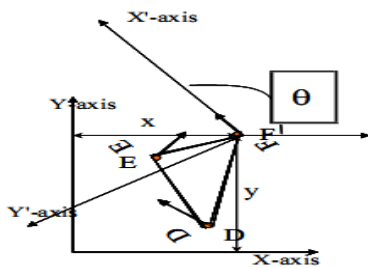


Fig-12: Translation and Rotation

The new coordinate system is originated at minutia F and the new x-axis is coincident with the direction of minutia F. No scaling effect is taken into account by assuming two fingerprints from the same finger have nearly the same size. If it is more than the threshold, each set of minutia is

transformed to a new coordination system whose origin is at the referenced point and whose x-axis is coincident with the direction of the referenced point.

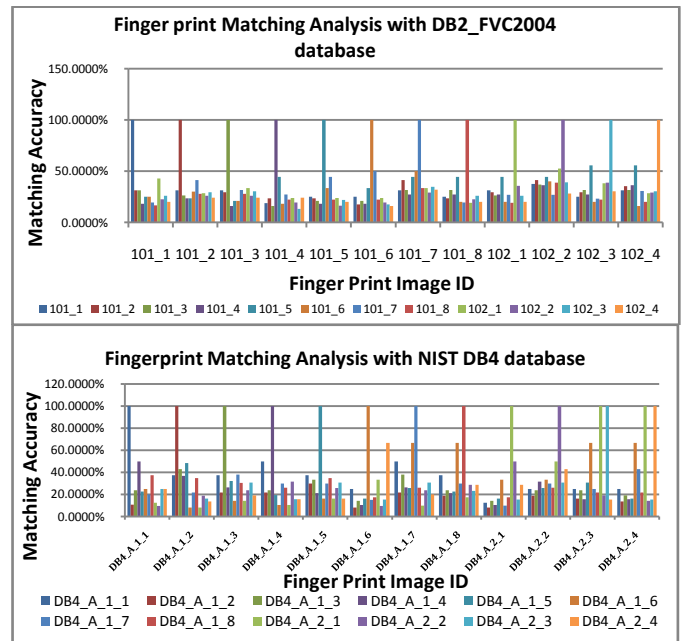
Match stage: Elastic match algorithm is used to count the matched minutia pairs by assuming two minutia having nearly the same position and direction are similar.

4. EXPERIMENTAL RESULTS

The algorithm has been tested with different database like NIST DB4, DB2 FVC2002, DB2 FVC2004 using MATLAB with two finger print image input for matching. The test results reports maximum of 1.3880% FRR with DB4 and 0% with DB2 FVC2002 and DB2 FVC2004 respectively. The FAR reported is 0% with all the databases. The False fingerprint acceptance in most of the cases not even crossed 50% as shown in plots. The Table 1 depicts the image sample used for experimental purpose and details of the data recorded with the matching performance using different databases are shown in Fig. 13.

Table-1: Finger print samples image information with performance indices for different databases

Databas	Image Size	No of	FRR	FAR
NIST	388 X 374	144	0%	1.39
DB2	388X 374	144	0%	0%
DB2	388 X 374	144	0%	0%



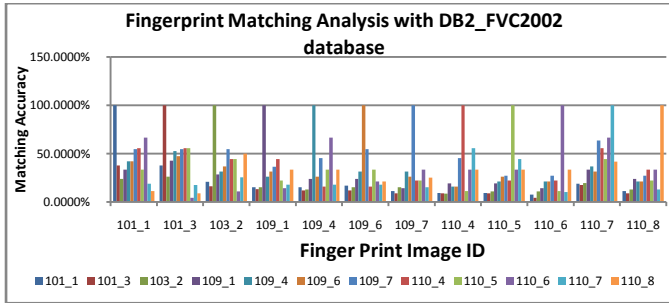


Fig-13: Finger print samples with performance indices for DB2_FVC2004, NIST DB4, DB2 FVC 2002 databases

5. CONCLUSIONS

After reviewing the importance of thresholding in image analysis and difficulty in handling the images captured under uneven illumination, the solution for the problem can be obtained using ECH, a multilevel thresholding technique. The new proposed method overcomes from most of the factors that have listed in section 2.1. This paper presents the minutiae based Finger print recognition method. Our method combines many methods to build a minutia extractor and minutia matcher. Segmentation is achieved using morphological operations, minutia marking with special considering the triple branch counting, minutia unification by decomposing a branch into three terminations, and matching in the unified x-y coordinate system after a two-step transformation are used. The idea of feature extraction using Cross Numbering (CN), which has advantages like increased computational efficiency, improved localization and higher sensitivity as compared with other methods. The overall experimental results were found prominent with different databases.

ACKNOWLEDGMENTS

This work is supported and funded by the University Grant Commission (UGC) of India under major research project (F.No. 41-648-2012 (SR)) to the Department of Computer Science, Pondicherry University, Puducherry, India.

REFERENCES

[1]. Naveed Bin Rais, M. Shehzad Hanif and rmtiaz A. Taj, "Adaptive Thresholding Technique for Document Image Analysis", INMIC 8th International Conference IEEE, pp.61-66, 2004.

[2]. Guoqing Gu Wenwen Han, " Adaptive window based Uneven Lighting Document Segmentation", IEEE Ninth International conference on Document Analysis and Recognition, pp.713-716, 2007.

[3]. P.Kanungo P.K.Nanda A.Ghosh, "Parallel Genetic Algorithm based adaptive thresholding for image segmentation under uneven lighting conditions", IEEE International Conference on System Man and cybernetics, pp. 1904-1911, 2010.

[4]. Mehmet Sezgin, B. Sankur, "Survey over image thresholding techniques and quantitative performance evaluation", Journal of Electronic Imaging Vol. 13(1), pp146-165, 2004.

[5]. Zuoyong Li, Yong Cheng, Chuancai Liu, Cairong Zhao, "Minimum Standard Deviation Difference-Based Thresholding" IEEE International Conference on Measuring Technology and Mechatronics Automation (ICMTMA), pp. 664-667, 2010

[6]. Wang Wen-Cheng and Cui Xiao-Jun, "A Segmentation Method for Uneven Illumination Particle Images", Research Journal of Applied Sciences, Engineering and Technology 5(4): 1284-1289, 2013.

[7]. Agus Zainal Arifin, Akira Asano, "Image segmentation by histogram thresholding using hierarchical cluster analysis", Elsevier Pattern Recognition Letters, Science Direct, 2006.

[8]. Rashmi Saini, Maitreyee Dutta, "Image Segmentation for Uneven Lighting Images using Adaptive Thresholding and Dynamic Window based on Incremental Window Growing Approach", International Journal of Computer Applications (0975 – 8887), Vol. 56, No.13, October 2012.

[9]. Zin Mar Win and Myint Myint Sein, "Fingerprint Recognition System for Low Quality Images", SICE Annual Conference, Japan, Sep.13-18, 2011.

[10]. Mehmet Sezgin, Bulent Sankur, "Survey over image thresholding techniques and quantitative performance evaluation", Journal of Electronic Imaging, 13(1), 146-165, January 2004.

[11]. Aroop Mukherjee, Soumen Kanrar, "Enhancement of Image Resolution by Binarization", International Journal of Computer Applications (0975 – 8887), Volume 10- No.10, November 2010.

[12]. Gualberto Aguilar et al., "Fingerprint Recognition", Second International Conference on Internet Monitoring and Protection (ICIMP), IEEE, 2007.