International Journal of Advanced Computer Research
ISSN (Print): 2249-7277 ISSN (Online): 2277-7970
Volume-5 Issue-21 December-2015

Efficient Fingerprint Matching Based Upon Minutiae Extraction

Chiranjeeb Roy Chowdhury* **and Banani Saha**
Department of Computer Science and Engineering, University of Calcutta, Kolkata, India

Received: 21-August-2015; Revised: 08-October-2015; Accepted: 18-October-2015

Abstract

Fingerprints are one of the oldest and most widely used biometric security measures. Rapid advances in Computer Science and digital Image Processing have made it possible to design various Automatic Fingerprint Identification Systems (AFIS) which can compare certain features of an input fingerprint image with a series of template images stored in a database and find a match. This paper deals with the extraction of certain specific features from a fingerprint, called minutiae. Since low quality images tend to generate multiple false minutiae, a method has been included to detect and remove such false minutiae. Fingerprint matching is performed by matching the number and type of the minutiae. The false minutiae removal process helps to reduce the computational complexity and improve the accuracy of the match.

Keywords

Fingerprint matching, Minutiae, Bifurcation Point, Ridge-ending, False minutiae, Alignment.

1. Introduction

Biometric recognition refers to the use of distinctive physiological and behavioural characteristics like fingerprints, face or hand geometry, retina scans, etc. for automatically recognizing a person. Since it has been proven that fingerprints usually never repeat and also persist throughout the lifespan of an individual, they are one of the most widely used biometric characteristics [1]. Fingerprint matching can be further broken up into either identification or verification. Identification involves accepting the fingerprint of an individual and matching it with a database of fingerprints.

If this individual already has a recorded template fingerprint in the system, then the algorithm will throw up a match and it will be able to know the identity of the individual. Hence this is like a one-to-many match. Verification, on the contrary, requires the individual to declare his identity. Thus, the algorithm will match the input fingerprint with only the template corresponding to that individual. If it is a successful match, then the identity of the individual gets verified. This is like a one-to-one matching process.

Before a fingerprint image can be matched, it must undergo certain pre-processing steps to increase its quality. In case the fingerprint image obtained is partial or of low quality, then corrective measures must be taken. Hong et. al. [2] have an evergreen paper dealing with fingerprint enhancements, orientation using ridge alignments and also recovery of poor images. On a similar note, Jea and Govindaraju [3] have devised an algorithm for matching partial or low quality prints utilizing a neural network. They are basing the initial match on standard minutiae points, but overcome the drawbacks of some matching algorithms by using secondary features and flow network based brute force matching.

After pre-processing, certain characteristic features must be extracted from the fingerprint image. Usually, features like ridge endings and bifurcations, called minutiae, are extracted. Farina et. al. [4] have proposed methods for extraction of different types of minutiae.

Finally, before two fingerprints can be matched based upon their extracted features, they must be registered. Registration ensures that even if the two fingerprints are skewed, they get properly aligned and the features can be matched accordingly. Liu et. al. [5] have proposed a registration method based on entropy by maximizing the mutual information obtained from the images. The proposed work follows the lines laid

*Author for correspondence

out by many other related works in this field by utilizing minutiae like ridge endings and bifurcations to match fingerprint. But unlike most other work, this paper also deals with a method to remove false minutiae. This helps to reduce the computational complexity while matching with the template.

The organization of the paper is as follows. Section 2 deals with the preliminaries. Some related works in this field have been mentioned and described in Section 3. Section 4 deals with the proposed work and provides an analysis of it. Its results are mentioned in Section 5. Section 6 contains the conclusion and scope for future work.

2. Preliminaries

A fingerprint is simply an impression of the epidermal layer of skin on the fingertip. The most prominent feature that can be seen is an interleaved pattern of ridges and valleys. Usually, in fingerprint images, ridges are dark and valleys are white. These ridges and valleys mostly run parallel to each other, though at times ridges bifurcate or abruptly terminate. These ridges and valleys exhibit various patterns when the fingerprint is examined on different scales.

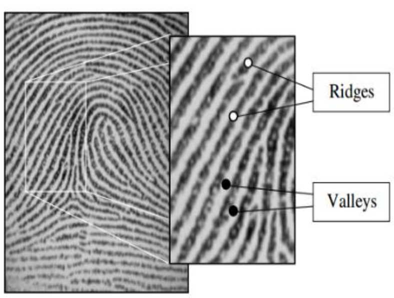

Figure 1: Ridge Valley Structure

Fingerprint matching can be performed based upon three main sets of features [1]:

i. Global Features – Singularity points like the core of a fingerprint (loop, whorl, delta, etc.), external fingerprint shape, coarse ridge line shape, etc.
ii. Local Features – Minutiae features like ridge termination, bifurcation, lake, spur, crossover, etc.
iii. Inter-ridge Features – High resolution images make it possible to distinguish sweat pores whose number, position or shape are usually distinct.

Most automated fingerprint matching techniques use the position and type of minutiae to help recognize a fingerprint correctly. Sweat pores usually cannot be distinguished due to the low quality of fingerprint images, smudged prints, etc. Global features may be used as a supporting parameter.

At a global level, ridges and valleys sometimes assume distinctive shapes in some regions, and these areas are known as Singularities. Usually three types of singularity points exist; loop, delta and whorl. Sometimes fingerprints are aligned using other features, like a landmark or center point known as the core. Unfortunately identifying the core is not possible for all sorts of fingerprints and thus registration may not be possible.

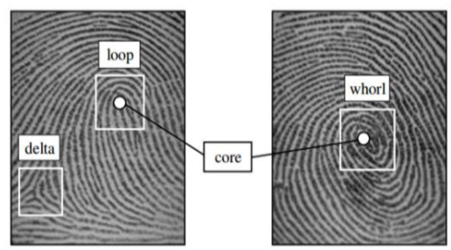

Figure 2: Singularity Points

Most algorithms deal with minutiae, which can be uncovered at the local level. Minutiae refer to the various ways in which the ridges can be discontinuous and thus exhibit various patterns. These are illustrated below:

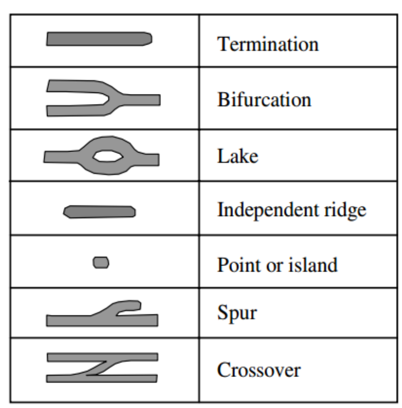

Figure 3: Types of Minutiae

As mentioned before, sweat pores (as shown below) can only be identified in images of approximately 1000dpi or more. Even though such pore information is quite distinctive and would have made a wonderful parameter to help recognize fingerprints, most automatic fingerprint matching algorithms cannot take an advantage of this because detecting pores reliably requires perfect, high quality fingerprints which are quite hard to come by.

Figure 4: Sweat Pores

3. Related Work

Dadlani et. al. [6] have submitted an interesting project which contains a fingerprint recognition mechanism utilizing minutiae. They have proposed an interesting technique of aligning fingerprints by taking triplets of minutiae from the initial template and storing the distance & angle between them. From the input image, they propose to consider triplets of minutiae and attempt to find a close match with any one of the previously stored triplets. Using the deviation thus obtained, they can easily align the two images.

Ohtsuka et. al. [7] has developed a method of fingerprint matching by Core and Delta detection. This can also be used for fingerprint registration. Ammar [8] and later Emary[9] have proposed fingerprint matching using Genetic Algorithms. Bhargava et. al. [10] have proposed a method whereby they identify ridge endings and bifurcation points of a template fingerprint by using a 3X3 window and store their amount and locations. The concept of Crossing Number (CN) has been used by Hong [2] and Thai[11], which has been further evolved in a recent work by Babatunde et. al. [12], and similar concepts have been used by Bansal et. al. [13].

The work proposed in this paper also identifies and removes false minutiae. A recent work [14] investigates the impact made on the matching process by the presence of such false minutiae. Popovic et. al. [15] have proposed a method to detect and eliminate false minutiae based upon multi scale directional information.

4. Proposed Work

This section outlines the process in which minutiae are extracted from a fingerprint image. The image must be initially pre-processed to increase its quality and then thinned such that the ridges are 1 pixel wide. Only then can the minutiae set be extracted. Finally, false minutiae must be removed from the above set, since many spurious minutiae can get introduced due to poor image quality or the thinning process.

The proposed work that is described in this paper has been illustrated in the flow diagram below followed by a detailed description of each stage. A pseudocode representation of the minutiae extraction and matching processes are presented in the next subsection. An analysis of it is given in subsection 4.2.

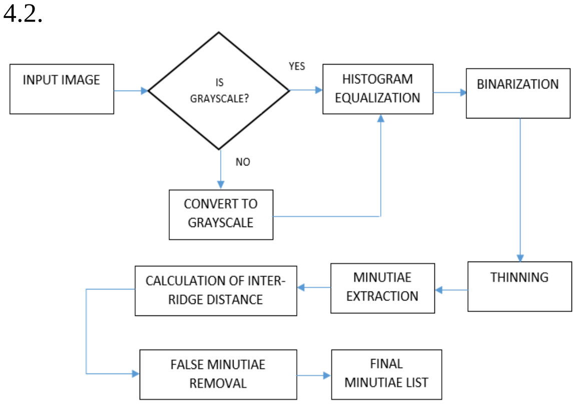

Figure 5: Flow Diagram

The algorithm assumes that the input image might be a colour image which shall be converted into a grayscale image. This makes it easier to enhance and then subsequently binarize the image. Since the input images can often be low contrast images, the algorithm enhances the image by performing Histogram Equalization which helps to spread the intensity levels of the image over all possible values thereby increasing the effective contrast.

This enhanced image has to be binarized by thresholding. Usually global thresholding cannot be performed since all parts of the image may not react properly to that global threshold. The proposed work uses Otsu's method for threshold calculation. It assumes that the image contains two classes of pixels (black and white). Then it calculates the optimum threshold value separating the two classes so that their inter-class variance is minimized. Using this threshold value the image is converted into a binary image. Pixels with values less than the threshold are mapped to zero while those with values above the threshold are mapped to one [16].

The binary image thus obtained now has to be thinned. This is essentially a morphological operation that iteratively removes pixels from objects thereby making them thin. When the process is performed for an infinite number of times, objects without holes in them ultimately shrink down to a connected stroke. Thinning helps to reduce the amount of information that the algorithm has to deal with and helps it to easily identify the various minutiae, since all ridges now reduce to white lines of one pixel width, and the background of the image is pure black.

The necessary minutiae can now be extracted from the given thinned, binary image. The minutiae that are dealt with are ridge bifurcations and terminations. Here, a 3X3 window is used to detect the two types of minutiae, as shown below.

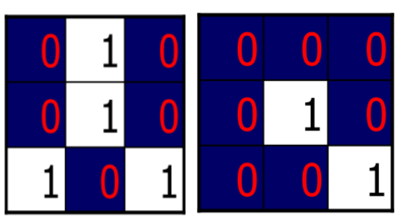

Figure 6: 3x3 Windows for Minutiae

These 3X3 windows are moved over all the pixels in the image. If the center pixel happens to be 1, it then performs a summation of all the pixel values around the center, that is, its 8 neighbors. If the summation turns out to be 3 then the minutiae is a ridge bifurcation. This is because, a continuous ridge suddenly breaks into two parts at the center point and thus the center point has three neighbors set to 1. On the other hand, if the summation is 1, it denotes a ridge termination. Since there are no other ridges around it, the center point has only one neighbor set to 1. Once a minutiae has been successfully identified, its X and Y coordinates are extracted and stored in a list. The window size is taken as 3X3 because only the exact positions of the ridge endings or bifurcations are of interest. Windows of any larger size may include multiple minutiae inside it and proper results will not be obtained. The computational complexity of each window will also get increased.

The average inter-ridge width D is now calculated. It is simply the average distance between two neighboring ridges and can be calculated as follows. A row of the thinned image is scanned and all the pixels that have a value of 1 is summed up. The row length is then divided by the above summation to get the required inter-ridge width. In order to make the calculations more accurate, the above calculation is performed on all rows of the image, and the values are averaged to get the final value of D.

Obviously, any noise in the input image or errors that have crept in due to the histogram equalization or thinning processes will cause many spurious minutiae to be generated and detected. Spurious minutiae may also be generated due to inaccuracies in the sensor device or due to unequal pressures of the finger. The most common problems are the appearance of extra bifurcations or endings or the joining of ridges. These spurious minutiae will now be removed as much as possible.

Several types of false minutiae have been identified as shown below.

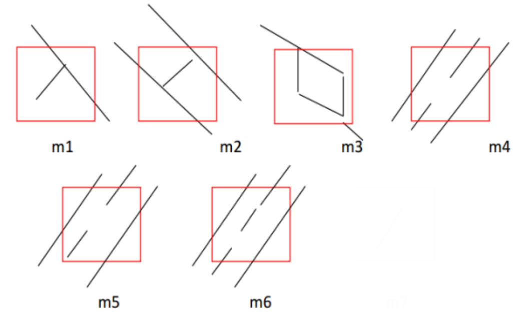

Figure 7: False Minutiae

From the figures above it is apparent that, if a bifurcation and a ridge ending pair have a distance less than the average inter-ridge width D they must be a spurious minutiae. A similar concept applies for a pair of bifurcations or ridge endings. These cases have been handled as follows.

Case (i) – If the distance between a bifurcation and a termination is less than D then it represents case ‘m1’.

Case (ii) – If the distance between two bifurcations is less than D it represents cases ‘m2’ and ‘m3’.

Case (iii) – If two ridge terminations are located at a distance lesser than D then it could be any of the cases ‘m4’, ‘m5’ or ‘m6’.

In all the above cases, both the minutiae are removed. Thus the final list of minutiae for the given image are obtained. The main algorithm takes as input two images, the input image and the template images one-by-one. On each of these images, the above procedure is applied to get the reduced list of minutiae after false positive removal. It then proceeds to calculate a percentage denoting the similarity of the two images.

But before the two fingerprints can be matched, a basic alignment step is performed. The leftmost minutiae of each of the images are located and the differences in their X and Y coordinates are calculated. This difference is used as a translation offset, to translate the template image’s leftmost minutiae on to that of the input image. Thus both the images get aligned to a certain extent and the match procedure is then performed only upon those minutiae that are in the valid range.

The percentage calculation is performed as follows. Since the minutiae sets obtained can be of different lengths, the smaller set is first located and the iteration is performed over that set. One minutiae is picked up from each of the two sets and the distance between them is calculated. If the types of the chosen minutiae is a match and their distances do not vary by a huge amount then these minutiae are considered to be a match. An approximate value of 30 pixels was used as the tolerance limit for this minutiae deviation. This allows slightly askew fingerprints to undergo a match without specialized image registration methods. Finally, the percentage is calculated and if it is high enough then the fingerprints are reported as a match.

4.1 Algorithm

The steps followed by the main fingerprint matching process and percentage calculation, is shown in the pseudocode procedure below.

Procedure Fingerprint Match –
Input: Input image and a set of Template images
Output: Matched Fingerprint
Step 1–Call Procedure MinutiaeExtract(I) and store the result in m1. /*Extract minutiae set of input image I in m1*/
Step 2–Call Procedure MinutiaeExtract(T) and store the result in m2. /*Extract minutiae set of template image T in m2*/
Step 3–Align the minutiae according to the leftmost minutiae of each image.
Step 4–Find the amount of matching minutiae by iterating over the smaller of the two minutiae set.
Step 5–Print matching percentage
Step 6–Return template image with highest matching percentage.

The steps for minutiae extraction from a fingerprint image and subsequent elimination of false minutiae, as described in the previous section, are illustrated in the pseudocode procedure given below.

Procedure Minutiae Extract –
Input: Fingerprint Image
Output: Extracted Minutiae Set

Step 1–If the image is color then convert it into grayscale
/*Pre-processing begins*/
Step 2–Perform histogram equalization on the image.
Step 3–Calculate a suitable threshold for binarization.
Step 4–Binarize the image using the above threshold.
Step 5–Repeatedly *thin* the image until the ridges are one pixel thick.
/*Minutiae Extraction begins*/
Step 6–Slide a 3X3 window across a pixel of the image.
Step 7–Calculate the summation of the 8 neighbors.
Step 8–If the center pixel of the window is 1 then go to step 9, else goto step 10.
Step 9–If the sum = 1 then it is a termination, else if the sum = 3 then it is a bifurcation.
Step 10–Repeat Steps 6 to 9 for each pixel of the image.
/*Post-processing begins*/
Step 11–Calculate the inter-ridge width D for each row.
Step 12–Find the average over all such D.
/*False Minutiae Removal*/
Step 13–If the distance of a ridge from neighboring ridges is less than D then discard the minutiae.

Step 14–Repeat the above process for bifurcation pairs and ridge-bifurcation pairs.
Step 15–Return the final minutiae set.

4.2 Analysis

The method for minutiae extraction from a fingerprint image takes approximately $O(n^2)$ time, where n is the number of pixels in the image. This is because we are moving the 3X3 window over all the pixels. The removal of false minutiae involves identifying pairs of bifurcations or ridge endings and hence requires $O(m^2)$ time, where m is the number of minutiae. Removal of false minutiae helps to increase the accuracy and the speed of the matching process. Also the extra time spent in false minutiae removal is low, since m << n. It has been shown in the next section that the minutiae count reduces by over 80% after removing false minutiae, thereby justifying the utility of this step.

5. Results

In this section the results of the proposed work is discussed and the steps that an input image goes through are illustrated. The images used here have been acquired from the FVC2002 fingerprint image database. This was the Second International Competition for Fingerprint Verification Algorithms. The implementation of the algorithm has been done in MATLAB. The input and template images are given below. Ten template images and one input image has been chosen from the above database.

Figure 8: Template and Input Images

The next image shows the binarized input image after enhancement operations like Histogram Equalization have been applied on it.

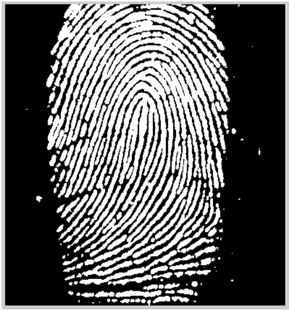

Figure 9: Binarized Input Image

After repeatedly thinning the input image till the ridges are just one pixel wide the following image is obtained. For clarity, only a part of the image is shown.

Figure 10: Thinned Input Image

The image obtained after minutiae extraction on the above thinned input image is given below.

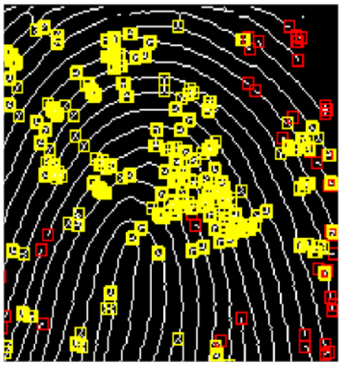

Figure 11: Detection of Minutiae

Once false minutiae are removed the next image is obtained. It is quite apparent though, that some false minutiae have remained near the fringes of the image.

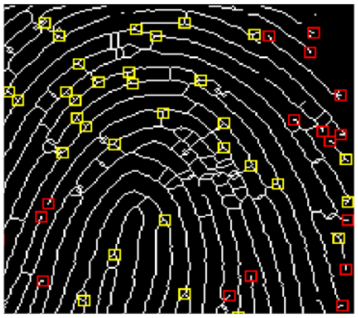

Figure 12: Removing False Minutiae

The table below shows the matching percentage of the input with the various chosen template images. The template with the highest percentage is reported as the final match. In this case, template 2 shows the highest match percentage, since it is of the same finger as the input image.

Table 1: Match Percentage

Image	Match Percentage
TEMPLATE 1	67.12
TEMPLATE 2	83.03
TEMPLATE 3	53.03
TEMPLATE 4	14.58
TEMPLATE 5	2.58
TEMPLATE 6	49.11
TEMPLATE 7	5.6
TEMPLATE 8	8.9
TEMPLATE 9	55.77
TEMPLATE 10	15.73

Table 2 shows a count of the number of minutiae that are extracted from each of the template images and the input image, before and after removal of false minutiae. The percentage reductions of the minutiae are also shown.

Figure 13 chart depicting the same information as the table above. It is quite clear from the chart, that after removal of false minutiae the number of spurious minutiae decreases hugely and helps to speed up the matching process. Thus, images that have a large number of false minutiae are actually low quality images, which lead to the creation of spurious minutiae.

Table 2: Minutiae Count

Image	Minutiae Count		% Reduced
	Before Removal	After Removal	
Template 10	2192	357	83.71
Template 9	974	52	94.66
Template 8	2765	394	85.75
Template 7	3231	342	89.41
Template 6	2053	170	91.71
Template 5	1890	211	88.83
Template 4	2339	222	90.50
Template 3	1518	103	93.21
Template 2	472	87	81.56
Template 1	1281	80	93.75
Input	860	161	81.27

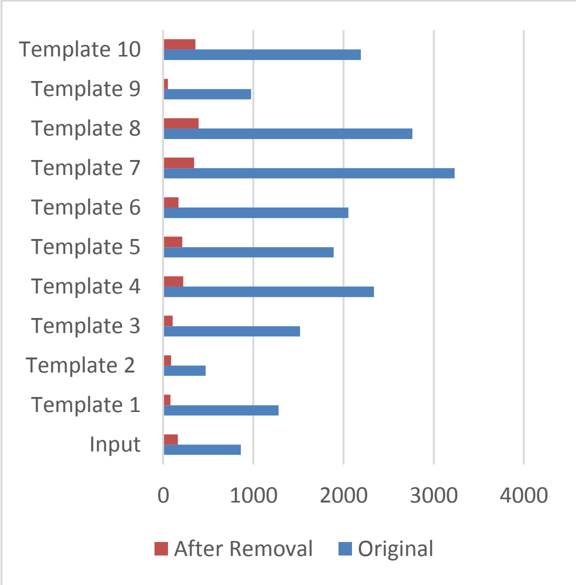

Figure 13: Minutiae Count

6. Conclusion and Future Work

The method proposed in this paper takes a fingerprint image, pre-processes it using histogram equalization to enhance its quality and then proceeds to extract ridge end and bifurcation minutiae from it. These minutiae are then reduced by removing certain false positives. Finally, fingerprint matching is performed between two images by matching corresponding minutiae pairs of the images.

Proper alignment and registration of the two fingerprint images has to be performed, since it increases the accuracy of the matching procedure. Apart from the simple translation performed here, rotation and other processes must be incorporated for maximum alignment. Also, extraction of more types of minutiae can be performed. But this increases the computational complexity. Thus a feasible accuracy-time trade-off must be devised here.

References

[1] Maltoni, Davide, et al. Handbook of fingerprint recognition. Springer Science & Business Media, 2009.

[2] L. Hong, Y. Wan, A. Jain, "Fingerprint Image Enhancement: Algorithm and Performance Evaluation", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 20, No. 8, August 1998, pp. 777-788.

[3] T. Jea, V. Govindaraju, "A minutia-based partial fingerprint recognition system", Pattern Recognition, Vol. 38, No. 10, 2005, pp. 1672-1684.

[4] A. Farina, Z. M. Kovacs-Vajna, A. Leone, "Fingerprint minutiae extraction from skeletonized binary images", Pattern recognition, Vol. 32. No.5, 1999, pp. 877-889.

[5] L. Liu, T. Jiang, J. Yang, C. Zhu, "Fingerprint registration by maximization of mutual information", IEEE Transactions on Image Processing, Vol. 15, No. 5, 2006, pp. 1100-1110.

[6] C. Dadlani, A.K. Passi, H. Sahota, M.K. Kumar, "Fingerprint Recognition Using Minutia-Based Features", Graduation Project Documentation, EE851: Biometrics, IIT Delhi.

[7] T. Ohtsuka, D. Watanabe, H. Aoki, "Fingerprint Core and Delta Detection by Candidate Analysis", IAPR Conference on Machine Vision Applications, 2007, pp. 130-133.

[8] H.H. Ammar, Y. Tao, "Fingerprint registration using genetic algorithms", Proc. 3rd IEEE Symposium on Application-Specific Systems and Software Engineering Technology, pp. 148-154, 2000.

[9] I.M.M. El Emary, M.M. Abdulkareem, "Fingerprints registration using genetic algorithm." Proc. of the 15th WSEAS international conference on Computers, World Scientific and Engineering Academy and Society (WSEAS), pp. 91-97, 2011.

[10] N. Bhargava, R. Bhargava, M. Mathuria, M. Cotia, "Fingerprint Matching using Ridge-End and Bifurcation Points", Proc. International Journal of Computer Applications, ICRTITCS, pp. 12-15,2012.

[11] R. Thai, "Fingerprint image enhancement and minutiae extraction", The University of Western Australia, 2003.

[12] I.G. Babatunde, O.C. Akinyokun, A.B. Kayode, O. Olatunbosun, "Adaptive and faster approach to fingerprint minutiae extraction and validation", International Journal of Computer Science and Security (IJCSS), Vol. 5, Issue 4, 2011, pp. 414-424.

[13] R. Bansal, P. Sehgal, P. Bedi, "Minutiae extraction from fingerprint images - A Review", IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 5, No. 3, September 2011, pp. 74-85.

[14] I.G. Babatunde, O.C. Akinyokun, O.S. Olusegun, "An Investigation into the Impact of False Minutiae Points on Fingerprint Matching", International Journal of Database Theory and Application, Vol. 7, No. 3, 2014, pp. 159-178.

[15] B.M. Popović, Lj Mašković, M.V. Banđur, "Spurious fingerprint minutiae detection based on multiscale directional information", Elektronika ir Elektrotechnika, Vol. 79, No. 7, 2015, pp. 23-28.

[16] R. Gonzalez, R. Woods, S. Eddins, "Digital Image Processing Using MATLAB", Gatesmark Publishing, 2009.

Chiranjeeb Roy Chowdhury was born in Kolkata in 1992. He is a first year student of M.Tech. (Computer Science and Engineering) at University of Calcutta, Kolkata. He has passed B.Sc. (Comp. Sc) from St. Xavier's College (Autonomous), Kolkata and M.Sc. (Comp. Sc.) from University of Calcutta, Kolkata, securing first position in both the degrees. His primary interests include Cryptography & Network Security, Soft Computing, Image Processing, etc.
Email: chiranjeeb.roychowdhury@gmail.com

Dr. Banani Saha is an associate professor, Department of Computer Science and Engineering in University of Calcutta. She received her Ph. D (Tech) degree from University of Calcutta, India. Her research interests are Data Mining, Data Warehousing, Image Processing, Data Fusion etc.