Fingerprint Image Enhancement with Neural network and effective extraction of minutiae

Dr. Meka James Stephen
WELLFARE INSTITUTE OF SCIENCE TECHNOLOGY AND MANAGEMENT (WISTM)

Fingerprint Image Enhancement with Neural Network and Effective Extraction of Minutiae

M. James Stephen
Asso. Professor, Dept. of I.T
ANITS, Visakhapatnam, INDIA
jamesstephenm@yahoo.com

P.V.G.D Prasad Reddy
Professor, Dept. of CS & SE
Andhra University, Vizag, INDIA
prasadreddy.vizag@gmail.com

Suresh Satapathy
Professor, Dept of CSE,
ANITS, Visakhapatnam, INDIA
sureshsatapathy@ieee.org

P.Swathi, R.Durganjani,
Ch. sruthi , 4/4 B.Tech, I.T
ANITS, Visakhapatnam, INDIA
swathi3190@gmail.com

***ABSTRACT: Among* the various biometric identifications available, fingerprints authentication is the oldest and the most reliable source of authentication. Identification of fingerprints basically relies on minutiae extraction. In most cases the finger print images that are obtained are of poor quality due to various reasons such as scars, dirt, non-uniform ink intensity and skin diversities. So enhancement of image prior to extraction increases the consistency. In this paper we proposed a new methodology for finger print image enhancement where we start with ridge orientation using neural network that follows dividing the image into white block, black block and grey block, which we called ternarisation, then we binarise the image using the concept of pixel aggregation. And finally we apply thinning for effective minutiae extraction.**

Keywords: filling, neural networks, thinning, image enhancement, ridge orientation, pixel aggregation, Ternarisation.

I. INTRODUCTION

Since ages fingerprint authentication has been a reliable source of authentication and its technology has developed over the years. It is used in the fields of forensic sciences, criminal investigation and computer sciences (access and control management). Fingerprint matching typically goes through a sequence of processes, mainly counting ridge orientation estimation, image segmentation, enhancement, ridge and minutiae detection, and matching [1,2]. It is estimated that roughly 10% of the fingerprint encountered during verification can be classified as 'poor' [3] which basically leads to improper extraction of minutiae and ultimately compromises the quality and accuracy of system . Improper contact with the device [4], skin condition and non-uniform ink intensity ultimately leads to noise in image. This leads to creation of bogus minutiae and ignorance of legitimate minutiae. The basic aim of this project is to reduce the effect of noise and improve the extraction of genuine minutiae. Here we use neural networks to estimate whether the ridge orientation is true or false. The trained neural network [5] returns a large value for true ridge orientation and a low value for false ridge orientation. Based on the values returned the image is segmented into white blocks, black blocks and grey blocks. White blocks are those which does not require further enhancement and are the area of interest .The black blocks consist the pixels with very low value and cannot be further resolved whereas the grey blocks need enhancement. This process is called ternarisation as we divide the image into three blocks. Later we use the concept of pixel aggregation to decide whether a pixel is included in a ridge (white block) or a valley (black block). The outcome is a binary image containing two levels of information, the foreground ridges and the background valleys. Finally thinning [6] is applied to preserve the connectivity of ridges and at same time produce a skeletonised image which is further used for genuine minutiae extraction. We used the old existing method of CN for minutiae extraction [12].

II. WHY USE NEURALNETWORK?

Fingerprint authentication requires proper extraction of minutiae for which the image has to be enhanced using appropriate techniques. Here we used neural networks which is a recent development and is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. NNs, like people, learn by example to estimate the true ridge orientation which further helps in segmenting the image. So as neural network can handle impression data effectively, we used it.

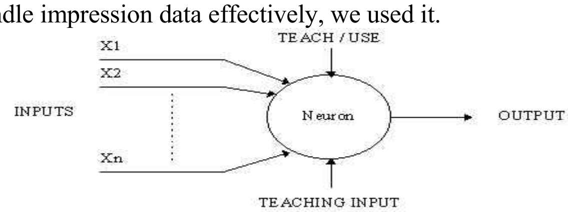

Figure 1. simple neuron

III. PROPOSED METHODOLOGY FOR FINGERPRINT ENHANCEMENT

In this section we will discuss the series of steps to create an enhanced image.

1. Estimation of ridge orientation using neural networks
2. Segmentation
3. Ternarisation
4. Binarisation through pixel aggregation
5. Thinning

A. Estimation of ridge orientation

Ridge orientation is the basic feature of a fingerprint Image, and ridge orientation estimation is almost the Prelude to fingerprint matching,

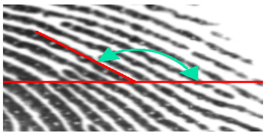

Figure 2. ridge orientation

1) Existing Approaches for estimation of ridge orientation

There are already some methods for fingerprint ridge orientation estimation in literatures [7]. Most of them are based on relationship of pixel intensity between pixels [7]. Among them, the most popularly used method for ridge orientation estimation is the gradient-based method or its variant. The gradient-based method is easy to be interfered by noise. In order to correct the false orientations caused by noise, [8] uses low pass filtering to smooth the orientation field. Sherlock [9] generates the orientation field based on the location of singularities, including core and delta. This method fails to distinguish two fingerprints which have the same locations and numbers of singularities but the different orientation field pattern. This method is improved by Vizcaya [10] and Araque [11]. Estimating ridge orientation of noisy image is still a not full-solved problem. Gradient based method operates on the fact that for high quality ridges the Proceedings of the 5th WSEAS Int. Conf. on Signal Processing, Robotics and Automation, Madrid, Spain, February 15-17, 2006 (pp158-164) gradient of most edge pixels has its direction almost orthogonal to the ridge orientation. However, low quality image block may contain edges which is not that of the ridges but of the noise caused by ridge interrupts, wet, dirt, etc. These spurious edges will produce false ridge orientations. And at the same time, modeling orientation field depends more or less on the locating of singularities which in return is dependent on the ridge orientation estimation itself. In order to correctly estimate ridge orientation in noisy ridge areas, we propose a machine learning based method.

2) Orientation estimation using neural network

Proposed a method to estimate the fingerprint image quality by training a BP neural network which, containing 11 input nodes, 11 hidden nodes and 1 output node, responds to correct ridge orientation of ridge block (of high quality or manually recoverable) with a large value, and responds with a small value to those blocks which contain no ridges or contain manually unrecoverable ridges or are of falsely estimated orientations. The training method is demonstrated in [12]. For each image block, a feature vector <C1,C2,……C11> [12] is computed to be fed into the network which will respond to the vector with a value. The responded value by the trained network to a specific block is depended on the orientation, because the items from C5 to C11 of the input vector <C1,C2,……C11> [12] have a close

relationship with the estimated ridge orientation. Suppose that the image is divided into unoverlapped blocks like in [12], and let W (i,j) denote the block at the *ith* row and the *jth* column. And the ridge orientation is quantified into 16 orientations: the *kth* orientation is $k \cdot \pi/16$ ($0 \le k < 16$). For each block W(i,j), 16 vectors, denoted as $<C1,C2,\ldots\ldots C11>_k$ ($0 \le k < 16$), can be computed, $<C1,C2,\ldots\ldots C11>_k$ corresponding to the orientation $k \cdot \pi/16$. For each block, feed the 16 vectors to the network and obtain 16 responded values, respectively. The trained network would generally respond with large values to the vectors corresponding to the orientation close to the true ridge orientation, and respond with small values to other vectors. We use these responded values to each block to estimate the ridge orientation. In the following sections we use orientation number i for representing the orientation $i\pi/16$. Let W be a image block. Net (W,i) represents the responded value of the trained network to the block W on orientation $i\pi/16$. The process is as follows:

(1) Estimate the responded value fields of the image I by the network on each orientation;
(2) Filtering the responded value fields in the orientation domain by low-pass filtering;
(3) Filtering the fields in the image domain by low-pass filtering;
(4) Orientation selection.

3) Computation of Responded Value Field

Let R[k](k=0,1,2,…,15) denote the responded value field of the image I by the network on the *kth* orientation ($k\pi/16$), and R[k](i,j)=Net(W(i,j),k) represents the responded value by the network to the block W(i,j) (ith row and jth column) on the *kth* orientation. Fig.1 gives out the gray representation of responded value field of image on the 16 quantified orientations (0, $\pi/16$, $\pi/8$, $3\pi/16$, $\pi/4$, $5\pi/16$, $3\pi/8$, $7\pi/16$, $\pi/2$, $9\pi/16$, $5\pi/8$, $11\pi/16$, $3\pi/4$, $13\pi/16$, $7\pi/8$ and $15\pi/16$).

For a responded value field on a certain orientation in Fig.3, the white block indicates it is fairly possible that the corresponding ridge block is of the correct orientation.

Or0	Or1	Or2	Or3	Or4	Or5	Or6	Or7
Or8	Or9	Or10	Or11	Or12	Or13	Or14	Or15

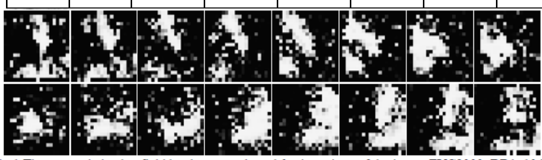

Figure 3. the responded value field by the network on 16 orientation of image. (Or indicates 'Orientation')

B. Segmentation

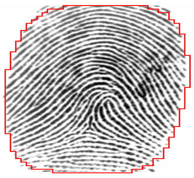

Figure 4. segmented image

The variance image in Figure 4 shows that the central fingerprint area exhibits a very high variance value, whereas the regions outside this area have a very low variance. Hence, the value attained by applying the above said method using neural network is used to separate the fingerprint into
1. White block –block with true ridge orientation
2. Black block –block which contains false ridges
3. Gray block – not sure either white or black

The black block is avoided totally since it contains false ridge values. The white block contains ridges that have true ridges and the gray block area need further enhancement before minutiae extraction is started.

C. Ternarisation

The gray blocks which has uncertainty is enhanced and it is sent through tenarisation stage where each pixel of gray block is assigned one of the three values of 1,0, or x. At first stage a block B is selected which is in gray block. A pixel P is selected in the block if the grey value of pixel is greatest, then the pixel value is set to a binary value one if the gray value of pixel is the lowest it is set to the value l and the remaining pixels are set to value of x. The outcome is a ternary image containing three levels of information. The pixels with x value are the pixels which are doubtful are needed to be included in either the ridge or the valley for accurate and genuine minutiae extraction. The threshold value of pixels are calculated for those pixels whose value is equal to 'x'.Since an image should contain only two values the value of 'x' need to be converted into either 1 or 0 based the threshold value which is accomplished in the method of pixel aggregation.

D. Binarization through Pixel aggregation

Most minutiae extraction algorithms operate on binary images where there are only two levels of interest: the black pixels that represent ridges, and the white pixels that represent valleys. Binarisation is the process that converts a grey level image into a binary image. This improves the contrast between the ridges and valleys in a fingerprint image, and consequently facilitates the extraction of minutiae. For the purpose of binarisation we used and implemented the concept of segmentation through pixel aggregation.

1) Algorithm:-

A. Nominate few pixels, which are different from others
B. Name the Nominated pixels as Seeds
C. Assume each Seed represents a region
D. If neighborhood pixel is similar to seed pixel then include neighborhood pixel into the region corresponding to seed. This is done as follows

Compute the difference between seed and neighborhood pixel

d=|Seed-Pixel|

If d<=threshold value then include the pixel into region of the seed.

E. Apply step 4 repeatedly & expand the regions.
F. Permit to grow regions till they cannot grow
G. As the output we get the few regions.
H. If these regions satisfy all the essential conditions then they become Segments.
I. Stop the Region Growing formulation until no more pixels satisfy the criteria for inclusion in that region.

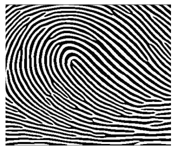

Figure 5. binarised image

The outcome is a binary image containing two levels of information, the foreground ridges and the background valleys.

E. Thinning

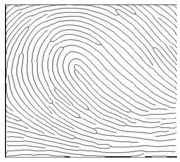

Figure 6. Figure 6: thinned image

The final image enhancement step typically performed prior to minutiae extraction is thinning. Thinning is a morphological operation that successively erodes away the foreground pixels until they are one pixel wide. A standard thinning algorithm [13] is employed, which performs the thinning operation using two sub iterations. This algorithm is accessible in MATLAB via the `thin' operation under the *bwmorph* function. Each sub iteration begins by examining the neighborhood of each pixel in the binary image, and based on a particular set of pixel-deletion criteria, it checks whether the pixel can be deleted or not. These sub iterations continue until no more pixels can be deleted. The application of the thinning algorithm to a fingerprint image preserves the connectivity of the ridge structures while forming a skeletonised version of the binary image. This skeleton image is then used in the subsequent extraction of minutiae

IV. MINUTIAE EXTRACTION

After a fingerprint image has been enhanced, the next step is to extract the minutiae from the enhanced image.
The Crossing Number (CN) method is used to perform minutiae extraction. This method extracts the ridge endings and bifurcations from the skeleton image by examining the

local neighborhood of each ridge pixel using a 3X3 window. The CN for a ridge pixel *P* is given by [12]: where *Pi* is the pixel value in the neighborhood of *P*. For a pixel *P*, its eight neighboring pixels are scanned in an anti-clockwise direction.

After the CN for a ridge pixel has been computed, the pixel can then be classified according to the property of its CN value. For each extracted minutiae point, the following information is recorded:
1. *x* and *y* coordinates,
2orientation of the associated ridge segment, and
3 type of minutiae (ridge ending or bifurcation).

V. IMPLEMENTATION

All the methods and algorithms described in this paper were implemented using MATLAB V6.5 on the Red Hat Linux operating system. The experiments were performed on a Pentium 4 - 2.4 Ghz with 512MB of RAM. When testing the performance of the enhancement algorithm, the computational time was not measured. We performed the experiments to see the results of each stage in the enhancement algorithm and to assess how well each stage performs. The experiments were conducted on a set of synthetic test images. Each synthetic image is formed by a series of circular patterns equally sized apart via the *circsine* function [14]. In addition, synthetic noise is generated via the MATLAB function random using a normal distribution. The majority of the real fingerprint images used in the experiments was obtained from the National Institute of Standards (NIST) fingerprint data set [15]. Each image is an 8-bit grey-level image scanned at approximately 500-dpi resolution and of size 832 X 768 pixels. For the purpose of conducting the experiments in MATLAB, the images were converted from their original Wavelet Scalar Quantization

(WSQ) format to the Portable Network Graphics (PNG) image format. In addition to the NIST data set, experiments were also conducted on images from the 2002 Fingerprint Verification Competition (FVC2002) database [15].

VI. EXPERIMENTAL RESULTS

In this section, we compare two methods for estimating fingerprint orientation field: method A—Gradient based method [16] followed by Guassian low pass filtering [16]; method B—the Proposed method.

TABLE I. BLOCK COUNTS OF INCORRECT ORIENTATION

	The number of ridge blocks of incorrect orientation	
	Method A	Method B
Image1	25	0
Image2	18	3
Image3	10	2
Image4	60	15
Image5	62	0

The results of table shows that the number of ridge blocks of incorrect orientation is almost equal to zero in almost all cases so we can conclude that applying neural network to fingerprint image separates true ridges from false ridges .accurately which helps in correct segmentation of image .After the fingerprint image is enhanced, it is then converted to binary form, and submitted to the thinning algorithm which reduces the ridge thickness to one pixel wide which increases the accuracy of system tremendously in extraction of minutiae.

VII. CONCLUSION AND FUTURE WORK

The primary focus of the work in this project is on the enhancement of fingerprint images, and the subsequent extraction of minutiae. Use of neural network for estimating the ridge orientation helped in better enhancement of image and proper segmenting of image.The use of neural network has reduced the rate of false minutiae extraction to a great extent. Ternarisation operation a method developed to separate uncertain and certain ridges .Minutiae extraction is followed by image postprocessing for eliminating false minutiae. Further study into the statistical theory of fingerprint minutiae. In particular, the Tu and Hartley [17] approach can be investigated to determine the number of degrees of freedom within a fingerprint population. These results can then be used to help us better understand the statistical uniqueness of fingerprint minutiae.

REFERENCES

[1] Zhu E., Yin J.P., Zhang G.M., Fingerprint Matching Based on Global Alignment of Multiple Reference Minutiae, Pattern Recognition, 2005, 38(10): 1685-1694

[2] Zhu E., Yin J.P., Zhang G.M., Hu C.F.,Fingerprint Minutia Relationship Representation Proceedings of the 5th WSEAS Int. Conf. on Signal Processing, Robotics and Automation, Madrid, Spain, February 15-17, 2006 (pp158-164) and Matching Based on Curve Coordinate System, International Journal of Image and Graphics, 2005, 5(4): 729-744.

[3] D. Maio, D. Maltoni, A. K. Jain, and S. Prabhakar. and book of Fingerprint Recognition. Springer Verlag, 2003.

[4] Jain, A., Hong, L., Pankanti, S., and Bolle, R. An identity authentication system using fingerprints. In Proceedings of the IEEE (September 1997), vol. 85, pp. 1365–1388.

[5] Fingerprint Ridge Orientation Estimation Based on Neural Network EN ZHU, JIAN-PING YIN, GUO-MIN ZHANG, CHUN-FENG HU

[6] Guo, Z., and Hall, R. W. Parallel thinning with two-subiteration algorithms. Communications of the ACM 32, 3 (March 1989), 359–373.

[7] Kawagoe M., Tojo A., Fingerprint Pattern Classification, Pattern Recognition, 1984, 17(3): 295-303.

[8] Hong L., Wang Y.F., Jain A.K., Fingerprint Image Enhancement: Algorithm and Performance Evaluation, IEEE Transactions on Pattern Analysis and Machine Intelligence, 1998, 20(8): 777-789.

[9] Sherlock B., Monro D., A Model for Interpreting Fingerprint Topology, Pattern Recognition, 1993,26(7): 1047-1055.

[10] Vizcaya P., Gerhardt L., A Nonlinear Orientation Model for Global Description of Fingerprints, Pattern Recognition, 1996,29(7):1221-1231.

[11] Araque J., Baena M., Chalela B., Navarro D., Vizcaya P., Synthesis of Fingerprint Images, Proc. Int. Conf. on Pattern Recognition, 2002,2:422-425.

[12] Zhu E., Yin J.P., Hu C.F., Zhang G.M.Quality estimation of fingerprint image based on neural network _ Proceedings of International Conference on Natural Computing, LNCS 3611, (2005) pp.65-70.

[13] Guo, Z., and Hall, R. W. Parallel thinning with two-subiteration algorithms.Communications of the ACM 32, 3 (March 1989), 359–373.

[14] Kovesi, P. MATLAB functions for computer vision and image analysis. School of Computer Science and Software Engineering, The University of Western Australia. http://www.cs.uwa.edu.au/»pk/Research/MatlabFns/index.html Accessed: 20 August 2003.

[15] Maltoni, D., Maio, D., Jain, A. K., and Prabhakar, S. Handbook of Fingerprint Recognition. Springer, 2003.

[16] Hong L., Wang Y.F., Jain A.K., Fingerprint Image Enhancement: Algorithm and Performance Evaluation, IEEE Transactions on Pattern Analysis and Machine Intelligence, 1998, 20(8): 777-789.

[17] Tu, P., and Hartley, R. Statistical significance as an aid to system performance evaluation. In ECCV (2) 2000 (2000), vol. 85, pp. 366–378.