Neural Network based Minutiae Extraction from Skeletonized Fingerprints

Josef Ström Bartůněk, Mikael Nilsson, Jörgen Nordberg, and Ingvar Claesson
Department of Signal Processing, School of Engineering, Blekinge Institute of Technology
Box 520, SE-372 25 Ronneby, Sweden
E-mails: jsb@bth.se, mkn@bth.se, jno@bth.se, icl@bth.se

***Abstract*— Human fingerprints are rich in details denoted minutiae. In this paper a method of minutiae extraction from fingerprint skeletons is described. To identify the different shapes and types of minutiae a neural network is trained to work as a classifier. The proposed neural network is applied throughout the fingerprint skeleton to locate various minutiae. A scheme to speed up the process is also presented. Extracted minutiae can then be used as identification marks for automatic fingerprint matching.**

I. INTRODUCTION

Fingertips in contact with smooth surfaces produce distinctive patterns, fingerprints which can be used for personal identification. In 1888 Sir Francis Galton found that fingerprints are rich on details in form of discontinuities in ridges. The most common minutiae in the fingerprints are the *terminations* and the *bifurcations*. Terminations are sudden ridge endings in a fingerprint and bifurcations are ridges that split into two new ridges. He also discovered that such features are permanent during an individs lifespan [1]. Therefore minutiae based matching is a viable method for fingerprint recognition.

Due to the varying quality of fingerprints, some preprocessing is usually required. Consequently an enhancement algorithm is applied on gray-scale images to improve and separate fingerprints from the background. This process is denoted binarization, the first preprocessing step, see Fig. 1. Some of the most frequent methods are directional filters [2], [3], [4], [5]. There are also other methods such as investigating the sign of the second directional derivative of the image intensity surface [6].

Minutiae are determined only by discontinuities in ridges, these are totally independent of ridges thickness. By minimizing data that represents minutiae without corrupting it, a more effective and fast minutiae extraction can be achieved. Thinning the ridges to only 1-pixel wide lines preserves minutiae with minimum data usage. This process is denoted skeletonization and follows the binarization. It is usually an iterative method, either sequential or parallel [7], [8].

The next step is the extraction of the minutiae from the skeletonized fingerprint, see Fig. 1. A method that handles this simply examines the nearest neighbor pixels around a pixel that belongs to a 1-pixel wide line [9]. Another method [10] studies the relationship between the thinned ridges and depending on the flow, it detects and extracts the various minutiae. Of course the risk with the

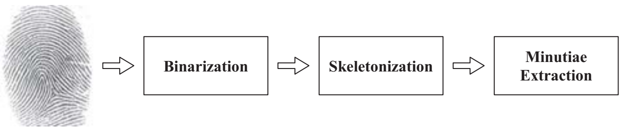

Fig. 1. Minutiae extraction with the preprocessing steps.

binarization and skeletonization is that some important details of a fingerprint might be removed. Therefore there are algorithms [11], [12] that are extracting the minutiae directly from the gray-scale image through ridge line following. A more recent minutiae extraction algorithm [13] analyzes intensity patterns in local area centered around a selected pixel.

The algorithms above have different rules or ad-hoc methods to handle the various situations that arise when extracting minutiae. This makes it more or less difficult to cover all possible situations. The method proposed in this paper is based on learning a neural network to recognize minutiae patterns in skeletonized fingerprint images. A well defined training set with suitably chosen size on data portions yields that no additional ad-hoc rules are required. Also a scheme of speeding up the processing is suggested. However, methods extracting minutiae from the skeletonized fingerprints are heavily dependent on the preprocessing stage. Producing high quality skeleton fingerprints relies on properly performed binarization and skeletonization. The methods used to produce skeleton fingerprints in this paper are based on, binarization [2], [3] and skeletonization [8].

II. MINUTIAE EXTRACTION

A. Neural Network

A multilayered fully connected feed forward neural network is proposed to serve as a minutiae classifier, see Fig. 2. The structure of the neural network has 24 inputs plus 1 input as bias in the input layer. It has two hidden layers with 25 neurons in each layer. The output layer has only 3 neurons; each representing a class of the patterns. A bipolar sigmoid function has been chosen as the activation function, named after the S shaped form and allowing both positive and negative values [14].

To establish which class the input pattern belongs to a so called *MAXNET* is added on the output layer, comparing the three values in output vector $\mathbf{y}$ from the

1-4244-0549-1/06/$20.00 ©2006 IEEE.

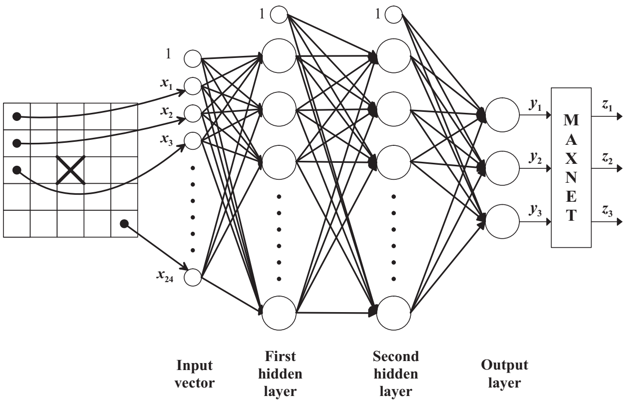

Fig. 2. The 5×5 local areas from the skeletonized fingerprint is the input to the classifier.

neural network. The output neuron that has the maximum value is the activated neuron. Hence, the input pattern is classified belonging to the same class that the neuron represents. The different output values in vector $z = [z_1 \quad z_2 \quad z_3]$ from the *MAXNET* are: Maximum value in y_1 yields $z = [1 \quad 0 \quad 0]$ and implies that the input pattern is a termination. When y_2 is maximum $z = [0 \quad 1 \quad 0]$ implying that the input pattern is a bifurcation. When y_3 is maximum $z = [0 \quad 0 \quad 1]$ implying that the input pattern is a nonminutia.

B. Training Method

Neural networks determines its weights by learning. Back-propagation [14] is one of many different learning algorithms widely applied for neural network training. For every input vector $\mathbf{x}$ that is presented to the neural network, there is a predefined desired response of the network in a vector $\mathbf{d}$. The desired output of the neural network is then compared with the real output by computing an error $\mathbf{e}$ from vector $\mathbf{d}$ and output vector $\mathbf{y}$. The correction of the weights in the neural network is done by propagating the error $\mathbf{e}$ backward from the output layer towards the input layer, thereby the name of the algorithm. The weight change in each layer is done with a steepest descent algorithm [14]. The back-propagation algorithm can be setup in many different ways and its fine-tuning is essential for proper behavior of the algorithm.

The initial weights are randomly chosen with the normal distribution in the range $[-0.5, 0.5]$. The bipolar sigmoid function that has been selected as activation function is

$$f(u) = \alpha \tanh(\beta u)\cdot \tag{1}$$

The parameters α and β are chosen as

$$\alpha = 1.7159 \tag{2}$$

and

$$\beta = \frac{2}{3}\cdot \tag{3}$$

This gives the function $f(u)$ characteristics such that $f(-1) = -1$,$f(1) = 1$ and a nearly linear transition with a slope close to one. The training is conducted through on-line learning, where the weights are updated after each pattern that is presented to the network. The patterns are selected randomly so the neural net does not learn the order of the patterns in the training data. To get a good realization of the neural network, the learning step η is made adaptive. The adaptation is done by *schedule annealing* also denoted *search then converge* [14], and η is given by

$$\eta(epoch) = \frac{\eta_0}{1 + epoch/\tau}\cdot \tag{4}$$

Epoch denotes the cycle where all patterns in the training data have been presented to the neural network. After each epoch the learning step is decreased according to the above stated equation, where τ and η_0 are parameters. In this paper the τ was chosen to 5000 and η_0 to 0.01.

The problem with two hidden layered neural network is that it can easily be over-trained. To avoid this unwanted effect the neural network should not be trained longer than necessary. Therefore the output from the *MAXNET* is compared with desired output to determine whether the correct neuron has been activated. The number of falsely classified patterns are counted after each epoch and is considered as a measure of how many patterns there are left to be learned. When the *MAXNET* measure is zero, all patterns have been exploited sufficiently by the neural network and the training can by stopped to avoid possible over-training.

C. Training Data

Training data is divided into three different pattern classes; termination, bifurcation and non-minutiae. The 3×3 pixel window does not regard much information and the 7×7 pixel window shows too much information. Therefore training data size is chosen to a 5×5 pixel window. Examples on how the different sizes of the window results on the training patterns are shown in Fig. 3. The size of the window is deliberately an odd number so there is a single pixel in the center.

A total of 23 different fingerprint skeletons have been used to collect the necessary patterns for the three classes. Then the neural network is trained and tested on the 23 fingerprints. Patterns that are falsely classified or minutiae missed by the neural network are added to the training database, and the neural network is retrained in a bootstraping manner. When the neural network classifies all patterns correctly in the 23 fingerprints, collection of

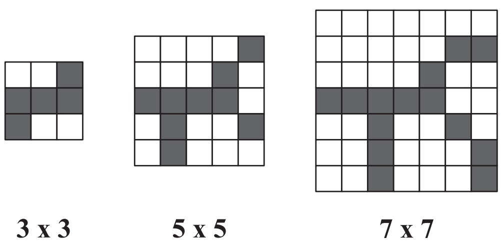

Fig. 3. Different window sizes for the training data.

the new patterns is stopped. A total of 1951 different patterns have been gathered. The different classes have 84 termination patterns, 388 bifurcation patterns and 1479 nonminutiae patterns. Patterns were carefully selected so the detection occurs in the center of the minutia. Patterns with minutiae off center are classified as nonminutiae to avoid overlap detection.

D. Extraction

The minutiae in the skeletonized fingerprints are made from the thinned lines. Therefore only local area patterns with lines intersecting the center (black pixel in the middle) are viewed by the classifier. This yields three improvements to the minutiae extraction process:

1) A skeleton fingerprint image has only about 30% of black pixels. This means that 70% of the data portions are not examined by the neural network. This gives faster and more accurate minutiae extractions, since the window can slide pixel-by-pixel through the image.
2) Since lines are centered in the data portions, the patterns where lines are off center does not need to be included in the training set. The training set can thereby be made smaller.
3) Because the mid-pixel in the data portion is always black, it does not give any new information to the neural network. Therefore the mid-pixel can be discarded from the input to the neural network and the input vector is thereby smaller.

Finally the extraction of the minutiae from the fingerprint's skeleton is done by sliding the 5×5 pixel window though the image. How the portions are presented to the classifier is illustrated in Fig. 2. The content of the input vector to the classifier is made of zeros (black pixels) and ones (white pixels). Since the mid-pixel is always black, it can be discarded and the column has 24 input bits plus a 1 as a bias. If the data sent to the neural network is a minutia, the coordinates (x, y) of the pixel in the picture are stored. The coordinates (x, y) represents the mid-pixel in the window corresponding to the global coordinates in the picture.

III. Experimental Results

The method of extracting minutiae from skeletonized fingerprints presented in this paper was evaluated by implementing it into the whole fingerprint recognition system. A database was assembled from pre-stored fingerprints in FVC2002 Db1_a found on DVD accompanying the book [1]. Two fingerprints from 20 different people were collected into the test database. The methods used in preprocessing are the binarization [3], [2] and skeletonization [8]. The proposed system extracts minutiae from the skeletonized images. The extracted minutiae (pointes) are then matched according to a point matching algorithm described in [15]. Two tests are preformed, identical (originating from the same person) and all combinations of nonidentical fingerprints. The result is presented in Fig. 4 as a plot between mean square error E_{ms} and

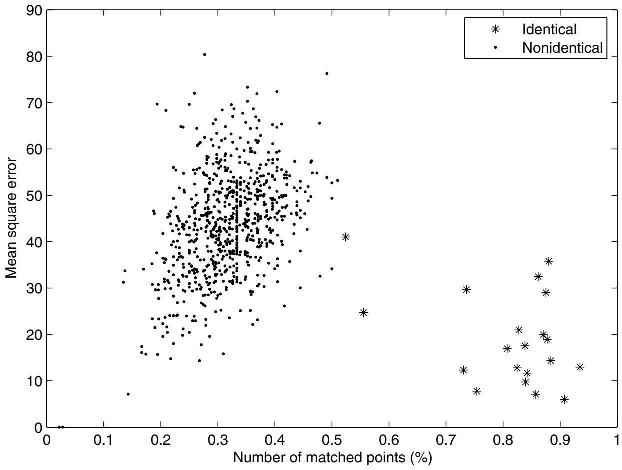

Fig. 4. Position of the examined fingerprint pairs for the two groups, plotted in the matched minutiae vs. mean squared error plane.

the number of matched minutiae in percentage. E_{ms} is the measurement of how much these matched points differ. The percentage number is calculated by dividing number of matched points with the minimum number of points from the two point patterns. It can be seen that the identical and nonidentical matching are creating two separable clusters, hence relaibile verification is possible.

Examples of how the neural network acts as a classifier is shown in the Fig. 5. By examining the skeletonized fingerprints, it can be seen that the classifier is highly accurate in extracting the precise location of the minutiae. Furthermore, the system has good fingerprint rotation invariant behaviour. It is successful in recognizing many different types and shapes of minutiae.

IV. Conclusions

In this paper an automatic minutiae extraction from skeletonized fingerprints is presented. A neural network was trained from carefully selected patterns to classify bifurcations, terminations and nonminutiae. Since the training was done with skeletonized patterns the classifier is universal towards all the skeleton fingerprints without adjustments. By only analyzing patterns with a black pixel in the middle of the local area, the speed gain is around 70%.

References

[1] D. Maltoni, D. Maio, A.K. Jain, and S. Prabhakar, *Handbook of Fingerprint Recognition*. Springer; 1st edition, 2003, ISBN 0-387-95431-7.

[2] L. O'Gorman and J.V. Nickerson, "Matched filter design for fingerprint image enhancement," in *Proc. Int. Conf. on Acoustic Speech and Signal Processing*, 1988, pp. 916–919.

[3] L. O'Gorman and J.V. Nickerson, "An approach to fingerprint filter design," *Pattern Recognition*, vol. 22, no. 1, pp. 29–38, 1989.

[4] B.G. Sherlock, D.M. Monro, and K. Millard, "Fingerprint enhancement by directional fourier filtering," *Vision, Image and Signal Processing, IEE Proceedings-*, vol. 141, no. 2, pp. 87–94, 1994.

[5] L. Hong, Y. Wan, and A. Jain, "Fingerprint image enhancement: Algorithm and performance evaluation," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 20, no. 8, pp. 777–790, 1998.

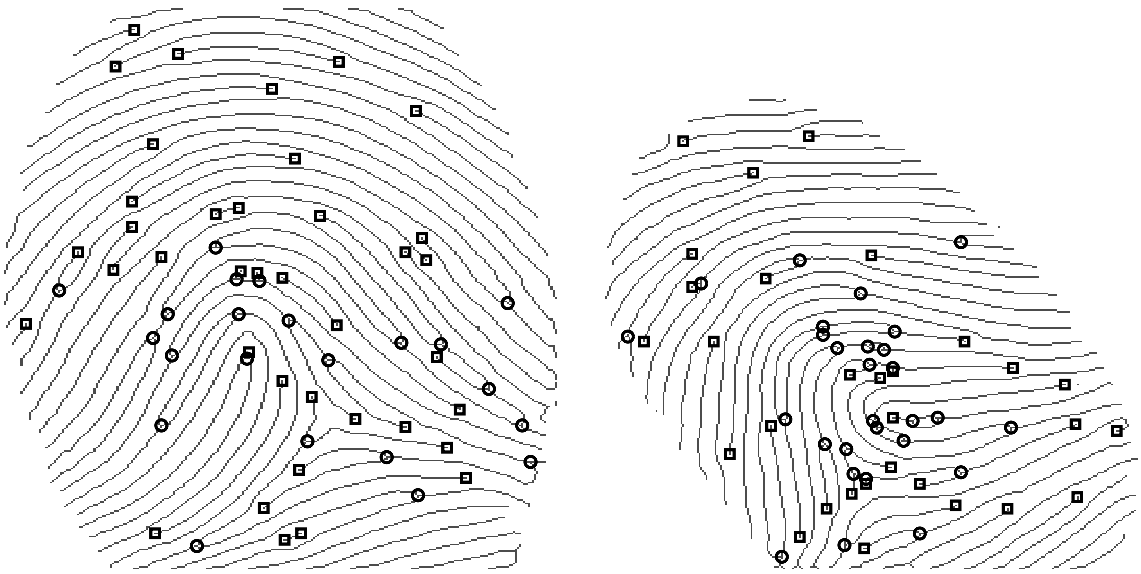

Fig. 5. Result of minutiae extraction. Terminations are marked with squares and bifurcations with circles.

[6] M. Tico, M. Vehvilainen, and J. Saarinen, "A method of fingerprint image enhancement based on second directional derivatives," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2005, pp. 985–988.

[7] L. Lam, S.W. Lee, and C.Y. Suen, "Thinning methodologies-a comprehensive survey," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 14, no. 9, pp. 869–885, 1992.

[8] J.S. Kwon, J.W. Gi, and E.K. Kang, "An enhanced thinning algorithm using parallel processing," in *Proc. Int. Conf. on Image Processing*, 2001, pp. 752–755.

[9] N. Ratha, S. Chen, and A.K. Jain, "Adaptive flow orientation based feature extraction in fingerprint images," *Pattern Recognition*, vol. 28, no. 11, pp. 1657–1672, 1995.

[10] A. Farina, Z.M. Kovacs-Vajna, and A. Leone, "Fingerprint minutiae extraction from skeletonized binary images," *Pattern Recognition*, vol. 32, no. 5, pp. 877–889, 1999.

[11] D. Maio and D. Maltoni, "Direct gray-scale minutiae detection in fingerprints," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 19, no. 1, pp. 27–41, 1997.

[12] X. Jiang, W.Y. Yau, and W. Ser, "Detecting the fingerprint minutiae by adaptive tracing the gray-level ridge," *Pattern Recognition*, vol. 34, no. 5, pp. 999–1013, 2001.

[13] M. Gamassi, V. Piuri, and F. Scotti, "Fingerprint local analysis for high-performance minutiae extraction," in *Proc. Int. Conf. on Image Processing*, 2005, pp. 265–268.

[14] S. Haykin, *Neural Networks: A comprehensive foundation*. Prentice-Hall; 2nd edition, 1999, ISBN 0-13-273350-1.

[15] S.H. Chang, F.H. Cheng, W.H. Hsu, and G.Z. Wu, "Fast algorithm for point pattern -matching: Invariant to translations, rotations and scale changes," *Pattern Recognition*, vol. 30, no. 2, pp. 311–320, 1997.