An Efficient Approach for Fingerprint Recognition

Tohid sedghi (1) – Majid fakheri (2) – Behbood mashoufi (1)

(1) Department of Electrical Engineering, Urmia university, Urmia, Iran
St.t.sedghi@urmia.ac.ir
(2) Department of Computer Engineering, Islamic Azad University, Urmia Branch, Urmia, Iran
www.urmia.ac.ir

Abstract: Human fingerprints are rich in details called minutiae, which can be used as identification marks for fingerprint verification. The goal of this paper is to develop a complete system for fingerprint verification through extracting and matching minutiae. A neural network is trained using the back-propagation algorithm and will work as a classifier to locate various minutiae. To achieve good minutiae extraction in fingerprints with varying quality, preprocessing in form of binarization and skeletonization is first applied on fingerprints before they are evaluated by the neural network. Extracted minutiae are then considered as a 2D point pattern problem and an efficient algorithm is used to determine the number of matching points between two point patterns. Performance of the developed system is evaluated on a database with fingerprints from different people and experimental results are presented.

Index Terms: Algorithm, Back propagation, Fingerprint, Neural nets.

I. INTRODUCTION

Skin on human fingertips contains ridges and valleys which together forms distinctive patterns. These patterns are fully developed under pregnancy and are permanent throughout whole lifetime. Prints of those patterns are called fingerprints. Injuries like cuts, burns and bruises can temporarily damage quality of fingerprints but when fully healed patterns will be restored. Through various studies it has been observed that not two persons have the same fingerprints, hence they are unique for every individual [1]. Today there exist many different automatic systems with various algorithms that solve this pattern recognition problem with very good results. However, this doesn’t means that automatic fingerprint recognition is a fully solved problem. Designing algorithms that gives robust solutions to this particular problem is still a difficult and challenging task. Automatization of the fingerprint recognition process turned out to be success in forensic applications. Achievements made in forensic area expanded the usage of the automatic fingerprint recognition into the civilian applications. Fingerprints have remarkable permanency and individuality over the time. Also the many years of satisfying experience in law enforcement, using the fingerprints as identification marks. The observations showed that the fingerprints offer more secure and reliable person identification than keys, passwords or id-cards can provide. With decreasing cost of fingerprint readers and cheap increasing computer power, automatic fingerprint recognition gives an efficient and inexpensive alternative to ordinary solutions in person identification [1]. Examples such as mobile phones and computers equipped with fingerprint sensing devices for fingerprint based password protection are being produced to replace ordinary password protection methods. Another limitation is that this paper takes only to consideration so called one-to-one matching method, otherwise also called verification. This means that the comparison is performed only once and that is between template (the pre-stored fingerprint(s) on a safe place in the system) and fingerprint of the person to confirm that his identity is the same as the templates.

II. PREPROCESSING

In this section the whole process of binarization, from filter mask design, orientation image estimation and smoothing to final enhancement filtering and post processing and detection and recognition are presented. For better understanding the approach here is proposed a diagram in Fig 1.

![Fig 1: Block-diagram of the total system design.](image)

All grayscale fingerprint images used in this paper and processes by this method are taken from the same database therefore originate from the same sensor. By carefully choosing the different parameters, the process is adjusted towards that certain fingerprint sensor and no further manipulation of the process is needed. In the Figure 2 is possible to see some advantages and disadvantages of this method. The binarization method is capable to filter out some small cuts and fills small gaps or holes in the ridges. The disadvantage is that some minutiae can be interchanged like termination to bifurcation and backwards. Also some small details in ridges can disappear.
Skeletonization is performed on so called "negative image" of the binarized fingerprint images since above specified rules uses 1 that represents the ridges and 0 for representing the background. The binarized fingerprint images uses exact the opposite signs. A negative image is simply formed by performing a logical NOT operation on the binarized fingerprint. The examination of the pixels is done in iterations where the first two rule sets are applied in turns. The pixels that can be erased are marked and first at end of each iteration are removed from the image. This process is repeated until there are no more pixels that can be removed from the image. Then the second skeletonization process is started to remove the remaining pixels to produce the 1-pixel wide lines. This process takes only one iteration. After that the image is converted back and the skeleton of the binarized fingerprint image is found. The flow diagram is shown in Figure 3

Fig 2: Advantages and disadvantages of the binarization process.

Fig 3: Flow diagram of the skeletonization process.

III. FEATURE EXTRACTION AND NEW APPROACH

Extracting minutiae from the skeleton of the fingerprint requires a method that is able to distinguish and categorize the different shapes and types of minutiae. This is a classification problem and can be solved by constructing and training a neural network which work as a classifier. Training of the neural network is conducted with the back-propagation algorithm. The whole network is build up with simple processing units, structures. The neurons are structured in layers and connections are drawn only from the previous layer to the next one. Training data is divided into three different pattern classes; termination, bifurcation and no minutiae. The neural network output layer consists of three neurons, each representing one class of the training data. By training the neural network so it activates the right neuron corresponding to the pattern class, the classification of the input patterns can be achieved. Size of the training data is chosen to 5 × 5 windows. The 3 × 3 window doesn’t view much information and the 7×7 window show to much information which is unnecessary. Example on how the different sizes of the window results on the training patterns are shown in Figure 5. The size of the window is deliberately odd so there is exactly one pixel in the center.

Fig 4: The steps between each iteration in the skeletonization process.

Fig 5: Different window sizes for the training data.

A total of 23 different fingerprint skeletons have been used to collect the necessary patterns for the three classes. A total of 1951 different patterns have been gathered. The different classes had 84 termination patterns, 388 bifurcation patterns and 1479 no minutiae patterns. Selection of the patterns was carefully done so the different minutiae types are found in the center. Even patterns with minutiae slightly off center are classified as no minutiae to avoid false classification towards that particular minutia type nearby. Examples of some patterns for each class can be seen in Figure 6.
To find an optimal size and shape of the neural network for good realization, training test on various nets was conducted. To measure how fast the different neural networks learns a total error in each epoch has been calculate. Three neural networks were tested, two with single hidden layer and one with two hidden layers. The single hidden layers had 50 and 60 neurons respectively and two hidden layers had 25 neurons each. All neural networks was trained for 5000 epochs and then compared how fast and low the error falls. A graph with the error function calculated for each neural network is shown in Figure 7. The “hl1 50” denotes one hidden layer with 50 neurons, “hl1 60” denotes one hidden layer with 60 neurons and “hl2 2525” denotes neural network with two hidden layers with 25 neuron in each layer. It can be clearly seen that the neural network with two hidden layers converges fastest and lowest. The two single hidden layered are about the same. The interesting thing is to see how much improvement the extra hidden layer does.

Finally the extraction of the minutiae from the fingerprint’s skeleton is done by sliding the 5 × 5 window though the image. Each portion is then presented to the trained neural network which classifies it into one of the three classes. If the data sent to the neural network is a minutia coordinates (x, y) of the pixel in the picture are stored in a vector. The coordinates (x, y) represents the middle pixel in the window corresponding to the global coordinates in the picture. To speed up and make things easier in the extraction of the minutiae, an assumption is made. The portions of data that are viewed by the neural network are only those who have a black pixel in the middle. Since the minutiae are made only from the thinned lines there is no need to examine data where lines are off center in the data portion. Input is made by stapling columns in the data portion into one column vector. Since the middle pixel is discarded the column has 24 bit plus a 1 bit as a bias. The neural network has two hidden layers with 25 neurons in each. Output layer consists of 3 neurons, each representing one class of the patterns. The shape of this neural network is actually the same as the neural network in the Fig 8 (b) Example of how well the neural network works as a classifier is shown in the next Fig 8. Notice the accuracy of the finding the minutiae in the center in the zoomed section of the image in the Fig 8 (b).

An algorithm is needed that localize the maximum number of mutual points in the two point patterns. The algorithm described in this chapter is based on literature [1] and a scientific paper [7]. The point patterns are constructed only on positions (x, y) of minutiae in the plane. The minutiae type and orientation which provides extra information are disregarded due to possible type alternation and noise in orientation. The alternation can be caused by varying pressure between fingertip and the sensor and also by binarization process. Low pressure can cause that bifurcation minutiae appear as termination minutiae in the fingerprint image. On other hand high pressure can cause termination minutiae appear as bifurcation minutiae in the fingerprint image. Alternations of minutiae types by the binarization process has been pointed out. Bad quality of fingerprint image gives noisy orientation image and therefore false minutiae orientation. Alternation and false orientation of the minutiae gives higher risk that not all mutual points are detected. Since point patterns are based on positions of minutiae in fingerprint they form distinctive patterns. With enough points in each pattern the positions (x, y) of the minutiae are the only information that is needed for good matching results. By using only (x, y) coordinates of minutiae yields that less memory is needed for implementation of this algorithm. The matching is performed on the two point patterns P with m number of points {p1, p2 . . . pm} and Q with n number of points {q1, q2, . . . , qn}. The algorithm is made invariant.
between the vectors with $S = \pi - \theta$. 

Denotes the position $\theta$ is in the interval $[0, 359.75]$ degrees with is converted from radians to degrees and that has the biggest peak in the $M(\theta)$ for which $a$ is found. The following pseudo code shows how the so called Matching Pairs Support (MPS) is calculated. The MPS value $w_{ia}$ is the number of most common $\theta$ between the vectors with $S_{\text{MIN}} < s < S_{\text{MAX}}$ where $S_{\text{MIN}} = 0.98$ and $S_{\text{MAX}} = 1.02$. The value $s$ is chosen around 1 because if the point patterns $P$ and $Q$ are originating from the same person and sensor the scale should be 1. Due to plasticity of the skin the points can shift the position to some extent and introduce noise to the coordinates. Therefore some variation in $s$ is needed to be taken in a consideration. After each calculation towards pair $p_i \leftrightarrow q_a$ the cumulative sum $M(\theta)$ is updated. The $M(\theta)$ is array where $\theta$ denotes the position in the array that is increased with 1. To make things easier the $\theta$ is converted from radians to degrees and quantized to 0.25 precision and is denoted $\theta^\circ$. The original $\theta$ is in the interval $[-\pi, \pi]$, the converted and quantized $\theta^\circ$ is in the interval $[0, 359.75]$ degrees with steps of 0.25 in-between. When all pairs $p_i \leftrightarrow q_a$ has been exploited, the accumulator sum is searched for the peak. The $\theta^\circ$ that has the biggest peak in the $M(\theta)$ is the corresponding rotation between $P$ and $Q$ for the $p_i \leftrightarrow q_a$ pair. The following pseudo code shows how the above described MPS works. Set the accumulator sum $M(\theta^\circ_{-+}) = 0$ for all $\theta^\circ$

$$S_{\text{MIN}} = 0.98$$

$$S_{\text{MAX}} = 1.02$$

for $j = 1, ..., m$, $i \neq j$

for $b = 1, ..., n$, $a \neq b$

$$S = \frac{q_a q_i}{p_b p_f}$$

if $S_{\text{MIN}} < s < S_{\text{MAX}}$

$$\theta = \theta_{q_a q_i} - \theta_{p_b p_f}$$

$$\theta^\circ = \text{quantize}(\theta\frac{180}{\pi})$$

end

end

Search the $M(\theta^\circ)$ for which $\theta^\circ$ it has the biggest peak. Return the biggest value labeled as $w_{ia}$ together with the according $\theta^\circ$. The developed matching algorithm is tested on two pairs of minutiae point patterns. The thresholds are chosen as following $\Delta S = 0.01$, $\Delta \theta = 0.50$ and $d_1 = 16$. The first pair of patterns $P$ with 58 points and $Q$ with 63 points is originating from fingerprints belonging to the same person. One of the fingerprints is rotated to test the ability of rotation invariance build in to the algorithm. The result can be seen in Fig 8 where in the upper row are the point patterns before matching. In the lower row are the point patterns after the matching has been completed and the boxes that has turned blue indicate found matching pair. The principal pair is marked with black box. Totally 48 points has been determined as matching pairs which is about 83% and the $Ems = 21$.

The Q pattern is the same as in the previous example, only the P pattern has been changed. Selected from the fingerprints similar looking as the fingerprint in pattern Q. Test is performed to see how many matching pairs the algorithm is able to detect. The result can be seen in Fig 3.8 where in the upper row are the point patterns before matching. In the lower row are the point patterns after matching. The blue boxes indicates the matching pairs and the principal pair is highlighted with a black box. Totally 48 pairs has been found which is 83% and $Ems = 21$.

In the second pair of patterns $P$ with 54 points and $Q$ with 63 points is originating from two different people. The Q pattern is the same as in the previous example, only the P pattern has been changed. Selected from the fingerprints similar looking as the fingerprint in pattern Q. Test is performed to see how many matching pairs the algorithm is able to detect. The result can be seen in Fig 3.8 where in the upper row are the point patterns before matching. In the lower row are the point patterns after matching. The blue boxes indicates the matching pairs and the principal pair is highlighted with a black box. Totally 48 pairs has been found which is 83% and $Ems = 21$.

Fig 8: An example of point pattern matching between two fingerprints that are originating from the same person. A 58 and 63 point are searched for matching points. In the upper row are the point patterns before matching and in the lower row are the point patterns after matching. The blue boxes indicates the matching pairs and the principal pair is marked with black box. Totally 48 points has been determined which is about 83% and the $Ems = 21$. 

PAPER ID: NEEC2010-F-023
Fig 9: An example of point pattern matching between two fingerprints that are originating from the different persons. A 54 and 63 point are searched for matching points. In the upper row are the point patterns before matching and in the lower row are the point patterns after matching. The blue boxes indicate the matching pairs and the principal pair is highlighted with a black box. Totally 25 pairs has been found which is 46% and Ems = 84.

IV. Experimental Results

In previous chapters the diverse parts of the fingerprint verification system has been in depth explained. In this chapter the performance evaluation of the developed system is in detail described and the experimental results are presented. 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 of good quality with varying rotation and translation were collected into the database. This database was then used to evaluate the performance of the fingerprint verification system. The matching is divided into two groups: examination of the identical fingerprints and examination of the non-identical fingerprints. The data of interest is the percentage of matched minutiae and mean squared error.

Table 1 shows the results for the matching of the identical fingerprint only. The values are the minimum, maximum and average of percentage matched minutiae and mean squared error.

<table>
<thead>
<tr>
<th></th>
<th>Min</th>
<th>Max</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ems</td>
<td>5.9720</td>
<td>41.010</td>
<td>19.069</td>
</tr>
<tr>
<td>[%] matched minutiae</td>
<td>52.381</td>
<td>93.478</td>
<td>81.128</td>
</tr>
</tbody>
</table>

Table 1: Results for the matching of the identical fingerprint.

To get a better view on how the two groups are separated, the data is plotted in the matched minutiae vs. mean squared error plane. The plotted data is viewed in Figure 4.1. The two groups are more of less forming clusters around the calculated average values.

Fig 10: Position of the examined fingerprint pairs for the two groups, plotted in the matched minutiae vs. mean squared error plane. The matched minutiae are given in the percentage.

It can be seen that the cluster of the non-identical examples is some what directed. The smaller number of matched minutiae is, the smaller Ems gets. While with increasing number of matched minutiae, the Ems gets bigger. This is because the smaller number of matched minutiae offers better chance to be found closer to each other. While greater number of matched minutiae is often obtained by the minutiae distance matching method. The Ems is in this case strongly related to the value chosen for the distance threshold d1 in the matching algorithm. The bigger d1 is the bigger Ems gets for the larger number of matched minutiae between the two non-identical fingerprints. This also explains why the non-identical cluster has greater overall Ems. Notice also the two examples of the non-identical fingerprints matching which has the Ems = 0 with very little percentage of the matched minutiae. This is due to the fact that fingerprints in those examples are that much different, that the matching algorithm found only one point pair in each. The cluster of the identical examples is rounder except the two examples closer to the 50% of the matched minutiae boundary. Usually this is because of the two identical fingerprints have smaller mutual area and therefore less minutiae are being matched together. The overall Ems is lower due to mutuality of the minutiae patterns that are being matched together. The matched minutiae aligns better to each other and the Ems gets smaller. Analysis of the experimental results shows that the developed system is capable of separating the identical from non-identical fingerprints. The separation is mainly in the percentage of the matched minutiae part. By choosing $\theta_{mp} = 52$ the percentage of matched minutiae is classified as identical match if the value is larger then $\theta_{mp}$. The $\theta_{mp}$ is a threshold for the matched minutiae in percent. It should be pointed out that this type of separation is quit dangerous. The separation between the non-identical and identical clusters is marginal of only about 1.5%. There should be at least one more checkup in form of...
examining the levels of the Ems. A threshold $ms$ is introduced to specify the maximum tolerance of Ems that identical matched fingerprints can have. The threshold is chosen to $0 ms = 42$. This method should be considered as a minimum of requirements for separating the two clusters apart from each others. Furthermore, good separation of the two groups is highly depending on the threshold values chosen for the diverse algorithms. In the test most of the parameters specified during the paper reminds unchanged. The only parameters which has been modified are $Δs \rightarrow 0.03$, $Δ θ \rightarrow 0.7$ and the $d1 \rightarrow 13$. The two fingerprints that are being matched should also have less than 50% of the difference in the number of minutiae. The larger difference in number of minutiae in the two point patterns is the higher chance there is to have higher number of matched minutiae. This is due to way the percentage of the matched minutiae is calculated. The formula is

$$\text{percentage} = \frac{\text{number of matched minutiae}}{\min([m \ n])}$$

(1)

where $m$ and $n$ are the sizes of the minutiae patterns. Therefore, fingerprints with higher difference in number of minutiae gives better chance for higher percentage of the matched minutiae.

**Conclusion**

The theory behind the fingerprint verification based on minutiae matching, was in detail studied. With obtained knowledge the complete system has been designed and implemented in Matlab. The performance of the developed system was evaluated on database with 2 fingerprints from 20 different people. The database was assembled from pre-stored fingerprints in FVC2002/Db1 a found on DVD accompanying the book [1]. The test showed that the system and algorithm is fully capable of distinguishing the related fingerprints apart from the non-related fingerprints. The system has proved to be robust towards translation, rotation and/or missing minutiae between the matched fingerprints.

**REFERENCES**


**Dr. Behbood Mashoofi** was born in Khalkhal, Iran, in 1964. He received the B.S. M.S and PhD degree in electrical engineering from Amirkabir university of Tehran, Iran in 1986, 1992 and 1997 respectively. He works in Engineering Department as an assistant professor at University of Urmia. His research interests include intelligent computing and optimization in signal processing. He has more than 15 journal and conference publications in subject of his interests. E-mail: b_mashoufi@yahoo.com

**Tohid Sedghi** was born in Urmia, Iran, in 1986. He received the B.S degree in Electronic engineering from IAU University of Urmia, Urmia, Iran, in 2007. He is now a M.S student in electrical engineering in University of Urmia, Iran. His research interests include image processing and intelligent computing and steganalysis. E-mail: St_t.sedghi@urmia.ac.ir

**Mir Majid Fakheri** was born in Urmia, Iran, in 1986. He received the B.S degree in Electronic engineering from IAU University of Urmia, Urmia, Iran, in 2006. He is now a M.S student in electrical engineering in University of Urmia, Iran. His research interests include image sensing and intelligent computing and watermarking. E-mail: st_m.fakheri@urmia.ac.ir