

Classification of Fingerprint Images Using Neural Networks Technique

Dr. Ebtesam Najim Abdullah AlShemmary

College of Education for Girls, University of Kufa, Najaf, Iraq
Email: dr.alshemmary@uokufa.edu.iq

Abstract

Automatic fingerprint identification is one of the most important biometric technologies. In order to efficiently match fingerprints in a large database, an indexing scheme is necessary. Fingerprint classification, which refers to assigning a fingerprint image into a number of pre-specified classes, provides a feasible indexing mechanism. In the field of criminal investigation the task of classifying fingerprints consumes much time and labor. Numerous attempts have been made to automate the classification process using conventional image processing techniques but very few have been embraced by law enforcement agencies due to their limited successes in solving the problem. The reemergence of interest in neural networks in recent years has caught the attention of those involved in fingerprint recognition as they begin to recognize the potential advantages of a neural network approach. In this paper, we introduce a new approach to fingerprint classification based on both singularities and neural network analysis. Since noise exists in most of the fingerprint images including those in the NIST databases which are used by many researchers, it is difficult to get the correct number and position of the singularities such as core or delta points which are widely used in current structural classification methods. The problem is we may miss the true singular points and/or get false singular points due to the poor quality of fingerprint images. Classification based on exact pair of singularities will fail in such conditions. This paper presents some intermediate results on fingerprint classification adopting a neural network as decision stage, in order to evaluate the performance of automatic fingerprint classification using neural network techniques.

Index Term: Fingerprint Image, Feature Extraction, Classification, Neural Networks.

1. Introduction

The automated classification and matching of fingerprint images has been a challenging problem in pattern recognition over the past decades. Several approaches for the solution have been proposed, however none of them is considered complete and the problem is still an open question. However, for large database applications, since the automatic comparison of fingerprints by matching their minutiae sets is time consuming, matching a query fingerprint with the entire database would be computationally intensive. We match the fingerprint with a subset of the database (reduced search-space) using a fast classification process. Most classification methods are based on the Henry classes

which classify the fingerprint images into *arch*, *tented arch*, *left loop*, *right loop* and *whorl* using the accurate position and type of the singular points such as cores and deltas. Table 1 and the images in Figure (1) give the general idea.

Table 1: Fingerprint pattern classes and the corresponding number of singular points.

Pattern Class	Core	Delta
Arch	0	0
Tented Arch	1	1 (middle)
Left Loop	1	1 (right)
Right Loop	1	1 (left)
Whorl	2	2

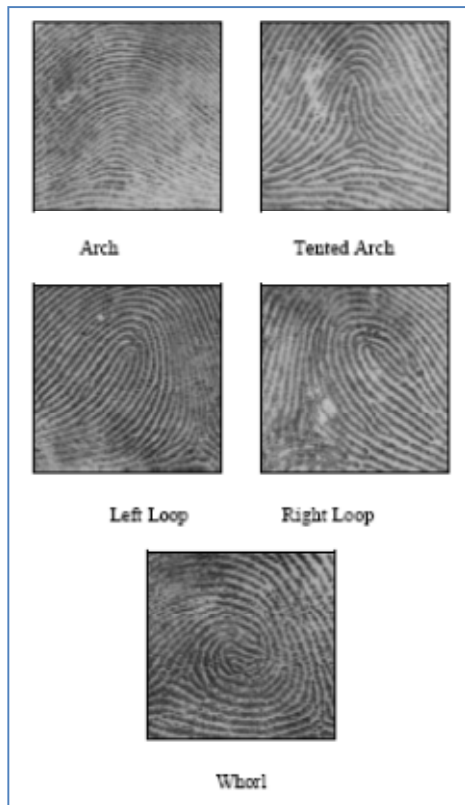


Figure 1: Five major fingerprint classes.

There are two types of features to be extracted from a fingerprint image: high-level or global features, and low-level or local features. The high-level features form special patterns of ridges and furrows in the central region of the fingerprint. And the low level features characterize an individual fingerprint image by detailed ridge characteristics or minutiae. The important high-level features are the core and delta points (also referred to as singular points or singularities). The core point is defined as the top most point on the inner most ridge and a delta point is defined as the point where three flows meet as stated by Jain [1]. These points are highly stable and also rotation and scale invariant. Thus, the number and the location of core and/or delta points are widely used by most of the classification methods [2,3,4]. The quality of the fingerprint images may be poor, due to noise such as that caused by scars, breaks, too oily or too dry, or having a partial image, usually with the delta point outside the print. Even in one image, different regions have different quality. So, all of these factors make it extremely hard to classify such images according to the rules in Table 1.

As a possible solution for the case of fingerprint matching the use of neural networks is suggested. Many reasons indicate the suitability of a connectionist approach, such as its robustness to noise and its adaptiveness. Furthermore, neural networks are trainable from examples and can be fine-tuned to fit the requirements of specific applications. Integration can be made between conventional statistical classifiers and neural network classifiers for fingerprint applications, which is expected to provide some motivating results on

the performance of neural networks for this particular task.

In the following sections, we will present the details of our fingerprint classification approach. Section 2 present our classification scheme. Section 3 presents the singularities detection method. Section 4 presents neural network classification and training. In Section 5, we present our experimental results on the NIST-4 database. The conclusions are presented in Section 6.

2. Classification

After a particular fingerprint has been analyzed and a coincident sequence worked out, it is stored along with its coincident sequence in its pattern type group (classification). Currently this task of classification is carried out manually as attempted automated systems using conventional image processing techniques have not been satisfactory. Classification of a fingerprint is achieved by examination of coarse-grained features (as opposed to minutiae). These are ridge lines, cores and deltas, see Figure (2). The core is approximately the centre of loop fingerprints. Arches do not have any cores. A delta is a point where ridges diverge. A whorl can contain two or more deltas. The precise location of the core and any deltas is extremely important if the correct (manual) classification of a particular print is to be made. Even very experienced fingerprint technicians can disagree on what the classification of a certain fingerprint should be. One advantage of any neural network which performs this task is that it will learn its own coarse-grained features, thus precise locations do not form any part of an input set.



Figure 2: Right loop showing features used for classification.

3. Singular Points Area

Let $I(x, y)$ denote the gray level of the pixel (x, y) in an $M \times N$ fingerprint image. Since most of the fingerprint images consist of the print itself, background and some handwritten letters or lines which may produce false singular points, it will be better to reduce the image to the print area only, Figure (3). The block wise average grayscale and standard deviation are used to segment the images. The block is considered as foreground if its grayscale mean and standard deviation satisfy some predefined standard, otherwise, the background. Then

two iterations of dilation and erosion as stated by Gonzalez and Woods [5] are used to remove holes resulting from in homogenous regions. Also, such a segmented result will be used in the feature extraction step to define the FingerCode (feature vector) for neural network. All the processes discussed below are carried out on such foreground regions. The block size we use is 8.

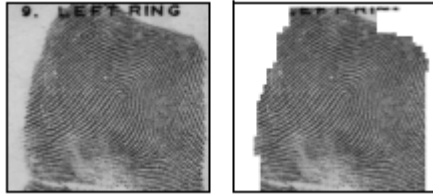


Figure 3: The original image and the image after segmentation.

3.1 Orientation Field Computation

We have adopted the following method which is presented by Jain [3] to compute orientation fields of fingerprints. We divide the input fingerprint image into blocks of (8×8) pixels.

$$V_x(i, j) = \sum_{u=i-\frac{W}{2}}^{i+\frac{W}{2}} \sum_{v=j-\frac{W}{2}}^{j+\frac{W}{2}} 2 g_x(u, v) g_y(u, v) \quad ,$$

$$V_y(i, j) = \sum_{u=i-\frac{W}{2}}^{i+\frac{W}{2}} \sum_{v=j-\frac{W}{2}}^{j+\frac{W}{2}} (g_x^2(u, v) - g_y^2(u, v)) \quad ,$$

$$\theta(i, j) = \frac{1}{2} \tan^{-1} \left(\frac{V_x(i, j)}{V_y(i, j)} \right)$$

Where $g_x(u, v)$ $g_y(u, v)$, are the gradients at each pixel. $\theta(i, j)$ is the direction of the block (i, j) . Finally, we use Gaussian smoothing operator to smooth the orientation fields in a neighborhood.

Since the gradient of a Gaussian filter can give a good estimate of the underlying oriented pattern as stated by Kass and Witkin [6] and Rao [7], we adopt its orientation as the local direction. The image is smoothed with a Gaussian filter whose impulse response is given by

$g(x, y) = e^{-(x^2 + y^2)/2\sigma^2}$, then the optimum 3×3 operators developed by Ando [8] are used to get the gradients g_x and g_y .

The orientation field is usually noisy, and in order to make singular points detection easier we smooth the orientation field using the method similar to the one developed by Jain, A. et al [9], see Figure (4).

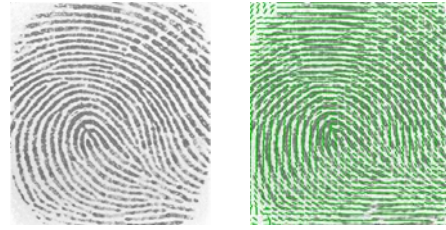


Figure 4: Orientation overlaid on the original images.

3.2 Singular Point Detection

Until now, a number of methods developed by Jain, L. C. et al [10], Halici and Ongun [11], Kawagoe and Tojo[4], Srinivasan and Murthy[12] have been proposed for the detection of singular points in fingerprint images, while the Poincare index which is derived from continuous curves is the most popular one. As for digital fingerprint images, a double core point has a Poincare index valued as 1, a core point $1/2$ and a delta point $-1/2$. So, the Poincare index is used first to find all the possible singular points in the orientation field.

The Poincare index at pixel (x, y) which is enclosed by a digital curve (with N_p points) can be computed as follows:

$$Poincare(i, j) = \frac{1}{2\pi} \sum_{k=0}^{N-1} \Delta(k)$$

$$\Delta(k) = \begin{cases} \delta(k) & \text{if } |\delta(k)| < \frac{\pi}{2} \\ \pi + \delta(k) & \text{if } \delta(k) < -\frac{\pi}{2} \\ \pi - \delta(k) & \text{otherwise} \end{cases}$$

$$\delta(k) = \theta(x_{(k+1) \bmod N_p}, y_{(k+1) \bmod N_p}) - \theta(x_k, y_k)$$

and it goes in a counter-clockwise direction from 0 to $N_p - 1$. For our method, N_p is selected as 4. Then the nearest neighbor cluster method is used to remove the false ones. Any types of potential singular points within some distance of each other are regarded as one cluster. After the clustering, we analyze the numbers of cores N_{ci} and deltas N_{di} in cluster i .

1. If $N_{ci} = N_{di}$, no singular point exists;
2. If $N_{ci} - N_{di} = 1$, the geometric center is regarded as a core point;
3. If $N_{ci} - N_{di} = -1$, the geometric center is regarded as a delta point;
4. If $N_{ci} - N_{di} = 2$, the geometric center is regarded as a double core;
5. Otherwise, no singular points exist in the cluster.

After this, there will be verification of such singular points. Five windows centered on the singular point are used to calculate their Poincare indices and the window size increases by 3 pixels. The singular points for which most Poincare indices conform to its type are verified the numbers of cores and deltas are N_c and N_d , see Figure (5).

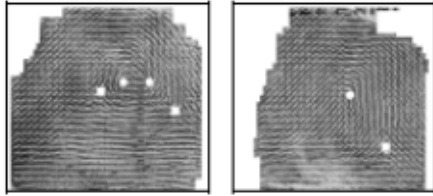


Figure 5: Singular points found (circle as core and square as delta).

Having found the core and delta points we can use their identified numbers are used to classify the fingerprint into one of the following three categories:

- (1) Arch with 0 core and 0 delta points,
- (2) Tented arch, left loop or right loop with 1 core and 1 delta point,
- (3) Whorl or twin loop with 2 core and 2 delta points.

To discriminate between a tented arch and a loop we consider the line drawn between the core and delta points is considered. Let the line have an angle β , with the horizontal axis.

In a tented arch, the direction of the connecting line is along the local direction vector of the orientation

field, while in a loop the line intersects the local direction vector transversely. Let γ_i , for $i = 1, \dots, m$, be the local direction angles along the line segment. If the averaged sum,

$$S = \frac{1}{m} \sum_{i=1}^m \sin(\gamma_i - \beta)$$

is less than a threshold, then the image is classified as a tented arch, otherwise it is a loop, Figure (6-b, c, d). The threshold value in the conducted applications was set to 0.2.

To discriminate between a left and right loop, a procedure is developed. Let the core point be denoted by C and the delta point be denoted as D, see Figure (6-c and d). Starting at C the direction vectors from one pixel to the next can be followed until meeting the boundary of the image at point B. Suppose the coordinates of C, D and B are (C_x, C_y) , (D_x, D_y) and (B_x, B_y) respectively, then the image is classified as a right loop if:

$$\frac{D_x - C_x}{D_y - C_y} > \frac{B_x - C_x}{B_y - C_y}$$

and as a left loop otherwise. This methodology was tested and found to work well and successfully, provided that the singular points are not located too close to the image boundaries. The results may be improved by careful consideration of the boundary process.

The classification algorithm is summarized here, Figure (7) essentially conducts a sequence of tests for determining the class of a fingerprint and applies simpler tests earlier in the decision tree.

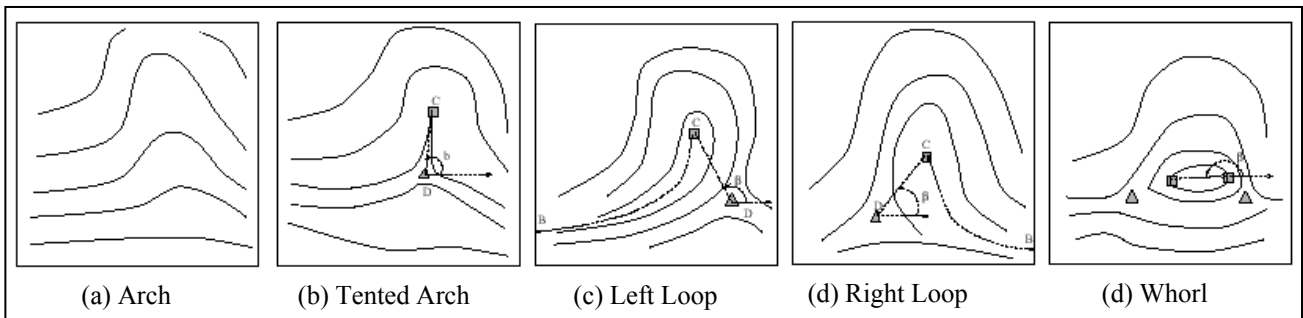
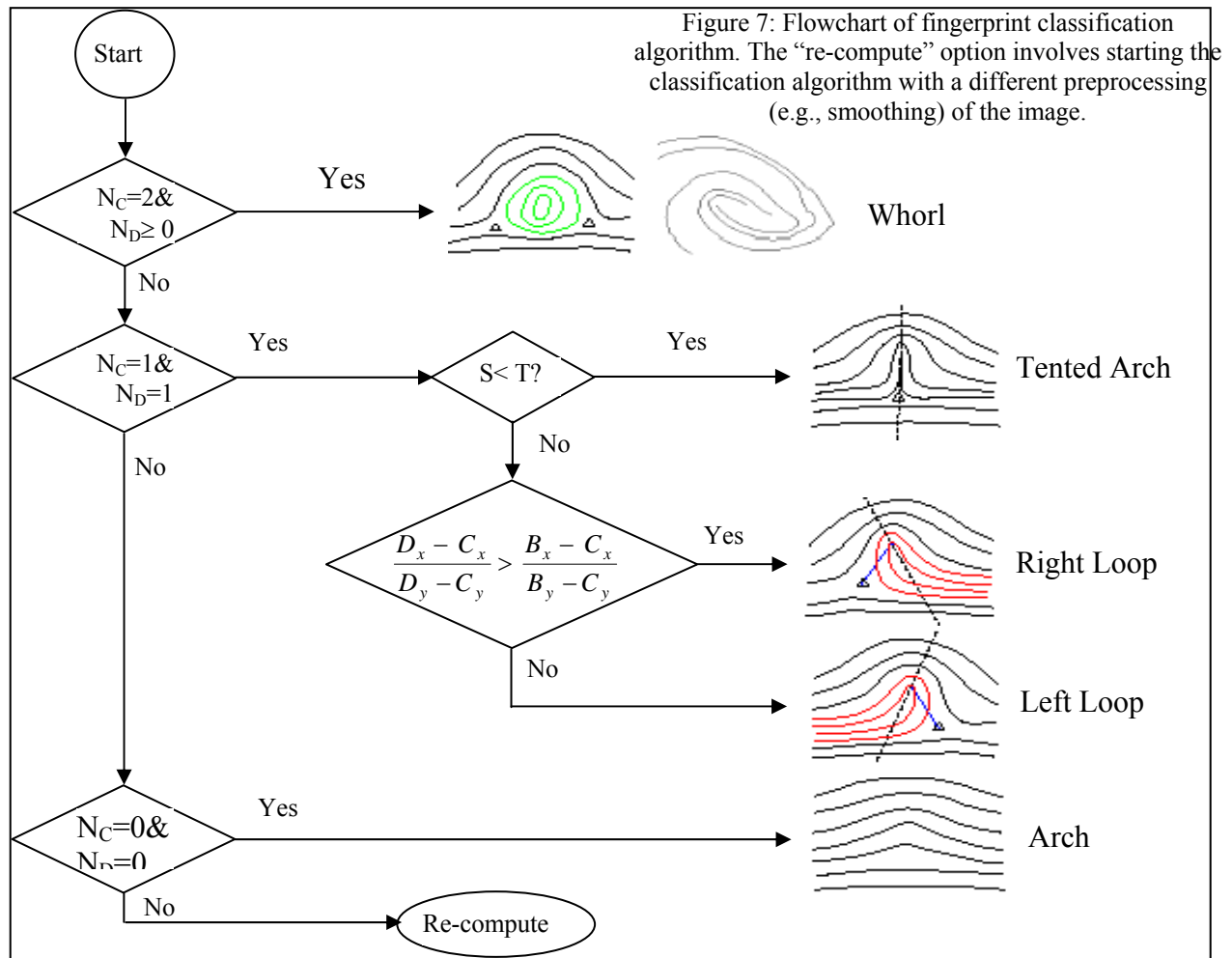


Figure 6: Five different fingerprint patterns showing core points (solid squares) and delta points (solid triangles).



3.3 Classification Rule:

The fingerprint classification algorithm classifies input fingerprints into *five* categories according to the number of singular points detected, their relative positions and the ridge structure [13]. A prototype of each class is shown in Figure (7). Let O' be the interpolated orientation field; N_C and N_D be the number of cores and deltas are detected from O' , respectively. The classification criteria used in the proposed algorithm are as follows:

1. If $(N_C = 2)$, then a whorl is identified.
2. If $(N_C = 2)$ and $(N_D = 2)$, then a whorl is identified.
3. If $(N_C = 1)$ and $(N_D = 1)$, then classify the input using the core and delta assessment (algorithm given in the previous section).
4. If $(N_C = 0)$ and $(N_D = 0)$, then an arch is identified.
5. If none of the above conditions is satisfied, then reject the fingerprint.

4. Automatic Neural Network Classification:

The extraction of relevant features of a pattern is not a trivial task. For the particular case of the feature extraction from fingerprint images several approaches have been developed, most of them based on special characteristics from the fingerprint patterns, such as ridge orientation and minutia detection [14,3]. The ridge orientation pattern of fingerprints is used to obtain feature vectors, which were used as inputs to statistical and neural networks classifiers. Several neural network models have been considered for the implementation of the fingerprint classifier. Only the use of a backpropagation network has been investigated [15,16].

4.1 Feature Extraction

The category of a fingerprint is determined by its global ridge and furrow structures. A valid feature set for fingerprint classification should be able to capture this global information effectively. The fingerprint representation developed in this paper is able to represent both the minutiae details and the global ridge and furrow structures of a fingerprint. For the purpose of classification, the proposed representation of this paper is adopted because it is effective in representing the global

ridge and furrow structures and it is invariant to individual minutiae details.

A fingerprint image is convolved with four Gabor filters ($\theta = 0^\circ, 45^\circ, 90^\circ$, and 135°) to produce the four component images. Thus, the feature vector is 400-dimensional (100×4). Acquired experimental results indicate that the four component images capture most of the ridge directionality information present in a fingerprint image and thus form a valid representation. This can be illustrated by reconstructing a fingerprint image by adding together all the four filtered images. The reconstructed image is similar to the original image without a significant loss of information. Using additional filters does not necessarily improve the directionality information in the reconstructed image. Since convolution with Gabor filters is an expensive operation, the use of additional filters will increase the classification time without necessarily improving the classification accuracy. In each component filtered image, a local neighborhood with ridges and furrows that are parallel to the corresponding filter direction exhibits a higher variation, whereas a local neighborhood with ridges and furrows that are not parallel to the corresponding filter tends to be diminished resulting in a lower variation. The spatial distribution of the variations in local neighborhoods of the component images constitutes a characterization of the global ridge structures which is captured by the average absolute deviation (variance feature) of grayscale values from the mean (AAD features).

4.2 Neural Net Design and Training

Typical pattern recognition systems are designed using two passes. The first pass is a feature extractor that finds features within the data which are specific to the task being solved. The second pass is the classifier, which is more general purpose and can be trained using a neural network and sample data sets.

Several types and topologies of NNs have been used for classification purposes. One of the main contributions of neural networks to pattern recognition focuses on implementing a standard backpropagation neural network, which is fully connected [15].

The neural-net classifier used in this paper is a three-layer perceptron-type trained with the Generalized Delta Rule algorithm [17]. A 3-layer neural network classifier is shown in Figure (8). The first layer is the input layer whose outputs are the values of the elements in the feature vector to be classified. Thus the number of input neurons is equal to the dimension of the feature vector plus one. The second layer is the hidden layer while the third layer is known as the output layer. Each neuron in the second layer computes the sum of all its inputs. This sum is then used in conjunction with a transfer function to determine that neuron's output. The number of hidden neurons is problem specific and must be determined by means of repeated experiments with different numbers of hidden neurons. The output layer has one neuron for each class. Each output neuron

performs the same calculation as the hidden layer neurons. A sigmoid transfer function for both hidden and output neurons was used. If a specific neuron in the output layer has a high output value for an input feature vector, the feature vector is labeled as belonging to that class for which the output neuron is high.

Each connection between the input layer and hidden layer, as well as each connection between the hidden layer and the output layer has a certain numerical value associated with it. All these numerical values are contained in the weight matrix W . The training of the neural net consists of determining the best weight matrix W for that specific problem. The initial values of the weight matrix are determined randomly but must adhere to specific rules. These rules state that the values must not be too big or too small and that they must be distributed randomly.

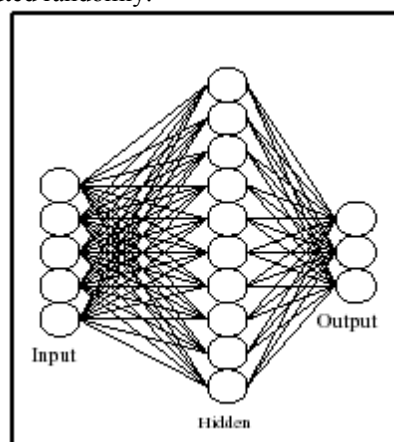


Figure 8: 3-Layer neural network classifier.

The neural net has several distinct advantages. The foremost is its ability to generalize very well due to its ability to generate very complex decision boundaries in feature space. If overtraining occurs, this can be a drawback if too complex decision boundaries are drawn. Another advantage is that the neural net does not require a huge amount of memory during classification. The number of computations required during classification is also very small.

5. Experimental Results

This section shows some implementation results. The training variables involved in the tests were: the number of cycles, the size of the hidden layer, and the learning parameter (which is a factor used during weight adjustment). A multi-layer feed-forward neural network was trained using a backpropagation training algorithm. The neural network has one hidden layer with (20) neurons, (400) input neurons corresponding to the (400) features, and 5 output neurons corresponding to the five classes. An accuracy of 92.7% was obtained for the five-class classification task. For the four-class classification task, an accuracy of 95.9% was achieved. The confusion matrix for the neural network classification is shown in Table 2 and Table 3.

Table 2: Confusion matrix for the neural network classification for the five-class problem.

True Class	Assigned Class				
	A	T	L	R	W
A	27 (96.4)	1 (3.6)	0 (0.0)	0 (0.0)	0 (0.0)
T	3 (17.6)	12 (70.6)	2 (11.8)	0 (0.0)	0 (0.0)
L	1 (3.8)	1 (3.8)	23 (88.5)	0 (0.00)	1 (3.8)
R	0 (0.0)	0 (0.0)	0 (0.0)	22 (100.0)	0 (0.0)
W	0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	30 (100.0)
Overall accuracy 92.7%					

Table 3: Confusion matrix for the neural network classification for the four-class problem.

True Class	Assigned Class			
	A	L	R	W
A	43 (95.6)	2 (4.4)	0 (0.0)	0 (0.0)
L	2 (7.7)	23 (88.5)	0 (0.0)	1 (3.8)
R	0 (0.0)	0 (0.0)	22 (100.0)	0 (0.0)
W	0 (0.0)	0 (0.0)	0 (0.0)	30 (100.0)
Overall accuracy 95.9%				

The supervised training was done with (400) sample sets selected randomly from the NIST-4. The training patterns were (277) presented randomly to the neural network and the remaining elements of the data set were used to test the neural network.

Figure (9) shows the behavior of the mean square error for the data sets, during training. The curves

correspond to learning constants (α) of (0.1, 0.3, 0.5 and 0.7). It could be observed that fast convergence of the networks is achieved in all cases. The training phases were divided in five stages of (1000) epochs, with intermediate tests to evaluate the resulting mean square error of the neural networks. The results of these intermediate tests are given in Table 4.

Table 4: Resulting mean square errors during intermediate tests.

Epoch	$\alpha = 0.1$	$\alpha = 0.3$	$\alpha = 0.5$	$\alpha = 0.7$
1000	0.000279	0.000309	0.000282	0.000347
2000	0.000234	0.000238	0.000209	0.000281
3000	0.000227	0.000231	0.000202	0.000248
4000	0.000224	0.000229	0.000201	0.000227
5000	0.000222	0.000227	0.0002	0.000206

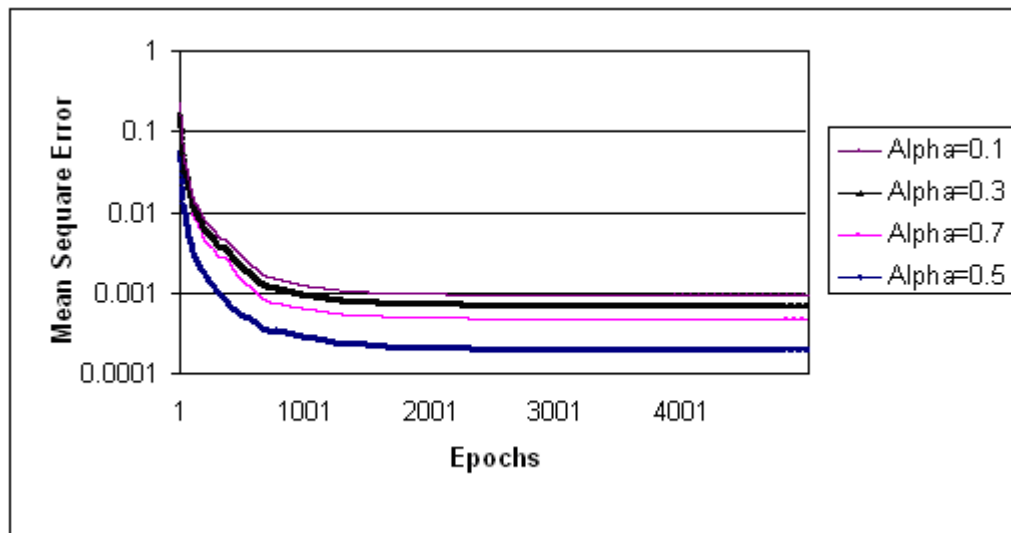


Figure 9: Mean square error during learning phase.

Figures (10 & 11) show the neural network classifier training error vs. the number of hidden neurons and vs. number of samples, respectively. The best performance is obtained for 12 and 20 hidden neurons.

The classifier performance is low for 2 hidden neuron. However, it increases to a maximum of 100% as the number of hidden neurons is increased to 20.

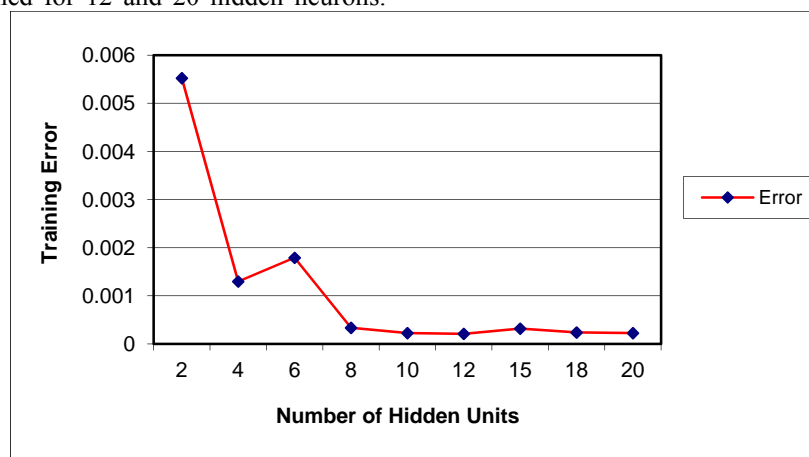


Figure 10: Neural-net training error vs. number of hidden neurons.

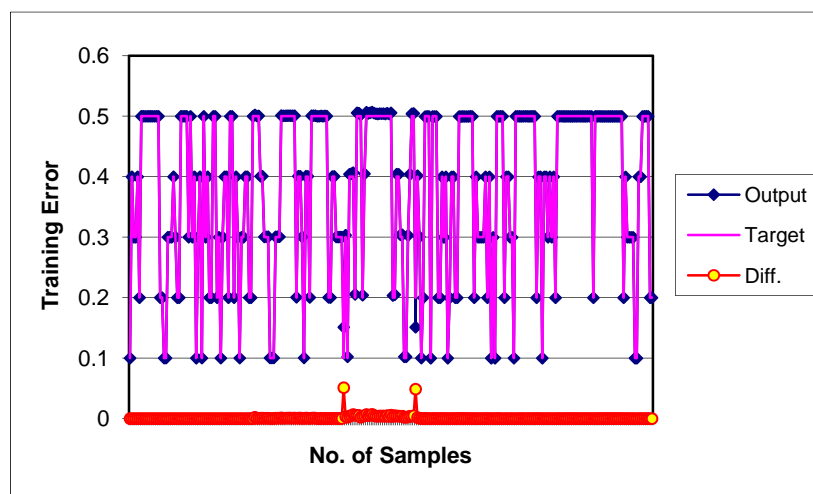


Figure 11: Neural-net classifier performance vs. number of samples

6. Summary and Conclusions

In this paper the problem of fingerprint classification as well as different possible solutions were described. Two methods of fingerprint classification were described and investigated, the well known Rule-Based classification and Neural Network classification.

In Rule-Based approach, features extracted from the directional image (singular point "core and delta") and the variance-based features are tested in the Neural Network based approach.

The classification algorithm we propose also works well when false singular points exist or true singular points are missing. If good enhancement methods are applied to the images or scale-variant orientation field smoothing methods are used, the result should be improved further. Also, the detection of singularities provides a good reference point for image registration which is mostly used for verification or matching and classification.

The developed system takes about 4 seconds on a Laptop Intel Pentium-M Processor 1.6 Ghz to classify one fingerprint. Good classification performance was obtained in general. However, the main problem which hampers the classification of a fingerprint image is still not solved. Some of the problems can be solved with preprocessing techniques but much better performance can be obtained if the images are clear and of good quality to start with. In general, good classification of fingerprint images is possible, but performance will only be perfect once the quality of the original print is of an acceptable standard.

It is shown that the simple variance-based features proposed in this paper work quite well. However, it is expected that better performance can be achieved by extracting richer, more discriminatory features from the filtered images in the feature extraction algorithm.

References

- [1] Jain, A., Prabhakar, S. and Hong, L., "**A Multichannel Approach to Fingerprint Classification**", IEEE Trans. on Pattern Analysis and Machine Intelligence, vol.21, no.4, pp. 348-359, (1999).
- [2] Cappelli, R., Lumini, A., Maio, D., and Maltoni, D., "**Fingerprint Classification by Directional Image Partitioning**", IEEE Trans. on Pattern Analysis and Machine Intelligence, vol.21, no.5, pp. 402-421, (1999).
- [3] K. Karu and A. Jain, "**Fingerprint Classification**", Pattern Recognition, vol.29, no.3, pp.389-404, 1996.
- [4] Kawagoe, M. and Tojo, A., "**Fingerprint Pattern Classification**", Pattern Recognition, vol.17, no.3, pp. 295-303, (1984).
- [5] Gonzalez, R. C. and Woods, R. E., "**Digital Image Processing**", Addison-Wesley, (1993).
- [6] Kass, M. and Witkin, A., "**Analyzing Oriented Patterns**", Computer Vision, Graphics, and Image Processing 37, pp. 362-385, (1987).
- [7] Rao, A. R., "**A Taxonomy for Texture Description and Identification**", Springer-Verlag, (1990).
- [8] Ando, S., "**Consistent Gradient Operators**", IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 20, no.3, pp. 252-265, (2000).
- [9] Jain, A., Hong, L. and Bolle, R., "**On-Line Fingerprint Verification**", IEEE Trans. on Pattern Analysis and Machine Intelligence, vol.19, no. 4, pp.302-314, (1997).
- [10] Jain, L. C., Halici, U., Hayashi, I., Lee, S. B. and Tsutsui, S., "**Intelligent Biometric Techniques in Fingerprint and Face Recognition**", CRC Press., (1999).
- [11] Halici, U., and Ongun, G., "**Fingerprint Classification Through Self-Organizing Feature Maps Modified to Treat Uncertainties**", Proc. of the IEEE, vol. 84, no. 10, (1996).
- [12] Srinivasan, V. S. and Murthy, N. N., "**Detection of Singular Points in Fingerprint Images**", Pattern Recognition, vol. 25, no. 2, pp.139-153, (1992).
- [13] A. Jain and S. Pankanti, "**Fingerprint Classification and Matching**", in Handbook of Image Processing, A. Bovik (ed.), Academic Press, pp. 821-835, (2000).
- [14] N. Rath, S. Chen and A. Jain, "**Adaptive Flow Orientation Based Feature Extraction in Fingerprint Images**", Pattern Recognition, vol. 28, no. 11, pp. 1657-1672, (1995).
- [15] I. Aleksander and H. Morton, "**An Introduction to Neural Computing**", Chapman and Hall, U.K., (1990).
- [16] D. Ali and K. Ali, "**Two Dimensional Pattern Recognition in Neural Networks**", Artificial Intelligence and Simulation Proceedings of the SCS., San Diego, U.S.A., February, (1988).
- [17] R. Schalkoff, "**Artificial Neural Networks**", McGraw-Hill Companies, USA, (1997).