Chapter 3

FINGERPRINT CLASSIFICATION AND VERIFICATION

3.1 Introduction

Fingerprints are the most widely used biometrics for the verification of user identity because of their high acceptability, immutability and uniqueness [56]. Here immutability refers to the persistence of the fingerprints over time whereas uniqueness is related to the individuality of ridge details across the whole fingerprint image. Fingerprints are commonly employed in biometric systems such as civilian and commercial devices for user identity proof. Another widespread use of fingerprints are in forensic science to support criminal investigations.

3.1.1 History of fingerprints

User identification through fingerprint biometric is a quite matured technique over the past century and experts have developed accurate procedures for determining the similarity of two prints. However, proficiency has been achieved at the cost of efficiency. It is surprising that the problem is still far from solved. A brief history of the science of fingerprints and applications are reported as follows:

1823 J. E. Purkinje, professor of anatomy at the University of Breslau, published his
thesis discussing nine fingerprint patterns.

1858 W. J. Herschel, a British Administrator in Hoogly district in India used fingerprints on civil contracts.

1880 Dr. Henry Faulds, a Scottish doctor in Tokyo, Japan recognized the importance of fingerprints as a means of identification and devised a classification method. In 1880, he published an article in the Scientific Journal, "Nature" (nature), mentioning printer ink as a method for obtaining fingerprints for personal identification.

1882 Gilbert Thompson of the U.S. Geological Survey in New Mexico, used his own thumb print on a document to help prevent forgery, which is the first known use of fingerprints in the United States.

1882 Alphonse Bertillion, French anthropologist, devised a method of classifying and identifying people known as Bertillion System.

1891 Juan Vucetich, an Argentine Police Official made the first fingerprinting of criminals.

1892 Sir Francis Galton, a British Anthropologist published the first book on fingerprints, establishing the individuality and permanence of fingerprints and defines the first classification system for fingerprints. According to his calculations, the odds of two individual fingerprints being the same were 1 in 64 billion. He also identified the characteristics by which fingerprints can be identified as minutiae is also known as Galton details.

1901 Sir Edward Henry, an Inspector General of Police in Bengal, India developed the first official system of classifying fingerprints in India and eventually spread throughout the world.

1905 U.S. Military adopts the use of fingerprints soon thereafter, police agencies began to adopt the use of fingerprints.
1908 The first official fingerprint card was developed.

1924 Formation of ID Division of FBI.

1980 First computer data base of fingerprints Automated Fingerprint Identification System, (AFIS) was developed. Presently there are nearly 70 million cards, or nearly 700 million individual fingerprints entered in AFIS.

2012 INTERPOL's Automated Fingerprint Identification repository exceeds 150,000 sets fingerprints for important international criminal records from 190 member countries.

2014 America's Largest Database, AFIS repository in America is operated by the Department of Homeland Security's US Visit Program, containing over 120 million persons' fingerprints, many in the form of two-finger records.

2014 The world's largest database, the Unique Identification Authority of India, also known as 'Aadhaar' operates the world's largest fingerprint (multi-modal biometric) system with over 560 million fingerprint, face and iris biometric records. The Aadhaar project has the ambitious goal of eventually providing reliable national ID documents for 1.2 billion Indian residents.

3.1.2 Fingerprint as a biometric

Fingerprint verification is a pattern recognition problem. Fingerprint patterns are compared with the help of developed algorithms and similarity between them is used to verify identity. Automatic fingerprint verification comprises of feature extraction, fingerprint classification and fingerprint matching. The effectiveness of feature extraction depends on the quality of the images, representation of the image data, the image processing models, and the evaluation of the extracted features. At the first stage of the fingerprint classification process, the image is only represented as a matrix of grey scale intensity values. Feature extraction is a process through which geometric primitives within images are isolated in order to describe the image structure, i.e., to
extract important image information and to suppress redundant information that are not useful for classification and identification processes. Thus fingerprint features and their relationships provide a symbolic description of a fingerprint image. Fingerprint classification is an important step in any fingerprint verification system because it significantly reduces the time taken in identification of fingerprints especially where the accuracy and speed are critical. To reduce the search and space complexity, a systematic partitioning of the database into different classes is highly essential. Fingerprint matching is done generally at two levels: at coarse level, fingerprints are classified into six distinct groups viz., whorl, arch, tented arch, left loop, right loop or twin loop and at fine level, matching is performed by extracting minutiae i.e., ridge ending and branching points. Global ridge shape provides important clues about the global pattern configuration of a fingerprint image. One of the most desirable features of a fingerprint representation and verification method is transformation invariance under translation, rotation and scaling. This indicates that the problem of identifying fingerprints is of a topological nature, rather than geometrical.

3.2 Prior related work

Based on our survey related to fingerprint classification [7], it has been observed that there exist different classification methods based on the features of fingerprints. The structural features present structural classification approaches, which are mostly based on syntactic pattern matching and graph matching. Also there exist various heuristic approaches based on singularities and ridge structures. In the neural network approach, the existing applications of neural networks can also be applied.

Based on our study, it has been observed that the classification methods are of eight major categories as follows.

A. Structural approach: The structural classification approaches classify input fingerprints based on the interrelationships of low-level features. It uses syntactic and graph based pattern matching approach.
i) Syntactic pattern matching: In syntactic pattern recognition an analogy is drawn between the structure of the input data and the syntax of a language. The input data is represented by a sequence of primitives which is considered to be a sentence of a language. Every class has an associated set of rules (or grammar) that describes how to build new sequences (sentences). Classification is performed by determining which grammar most likely produces a given input sequence.

ii) Graph matching: In this approach, two graphs are given as input, graph matching algorithms attempt to determine whether or not the graphs are isomorphic. For each fingerprint class, a model graph is created that has a structure typical of that class.

B. Heuristic rule approach: In this automated fingerprint classification approach, the knowledge of human experts is codified using a system of heuristic rules based on the singularity features, ridge features or a combination of singularity and ridge features.

i) Singularity structure-based approach: Since singularities are local features they are very sensitive to noise. Having detected a fingerprints’ singularities (core and delta points), heuristic rules based on their number and location can be used to classify fingerprints accurately.

ii) Global ridge structure-based approach [60]: Use the calculation of orientation fields to represent the global geometric shape of fingerprints based on analyzing the global geometric shape of the fingerprint. Twin loops can be recognized by the fact that they are the only global geometric shape that has two turns with opposite signs.

iii) Singularities and global ridge structure-based approach: The singularities perform very poorly on noisy images. The global ridge structure features are also difficult to deal with the large intra class variations and small interclass variations of fingerprint classes. Some systems overcome these limitations by using both singularities and ridge structures.
C. Neural approach: The research work of the applicability of neural networks to fingerprint classification began in the early 1990s and became one of the most commonly used classifiers for fingerprint classification systems. Researchers have developed different neural based classification approaches.

i) The neural approach developed by NIST for the FBI in 1990 [61]: This research formed the basis for the PCASYS system (Pattern-level Classification Automation System for Fingerprints) [62] PCASYS uses the core of loops, the upper core of whoels and a well-defined feature of arches and tented arches. The fingerprint's directional image is registered with respect to the centre of the fingerprint image. The dimensionality of the orientation field is reduced using the KL transform. Next, a probabilistic neural network (PNN) is used to classify the feature vector.

ii) The neural approach based on wavelet features: Wavelets form the basis of the FBI's fingerprint image compression scheme [63]; however, wavelets are sensitive to rotations and translations. A feed-forward neural network with a single hidden layer was trained to classify feature vectors consisting of 64 wavelet coefficients.

iii) The neural approach based on SOM: SOM's are based on Kohonen learning and are used for dimensionality reduction. A modified version of SOM also used that includes a certainty parameter to handle fingerprints [7]. The features being used for classification are the fingerprint's orientation field and some certainty measures.

iv) The neural approach based on fuzzy-network classifier: It combine the advantages of fuzzy logic techniques and neural networks and offer algorithms for learning and classification. The neural network is used to automatically generate fuzzy logic rules during the training period. It is based on singularity features that include the number of core and delta points, the orientation of core points, the relative position of core and delta points and the global direction of the orientation field. The authors Mohamed
and Nyongesa [66] point out that noise and preprocessing errors lead to an intra class variation among fingerprints.

D. Combining structural and statistical features: Structural features are extracted from the orientation field using a line tracing algorithm. Prominent flow lines are represented by strings of symbols that encode information about their endpoints and curvature. A three-layer feed-forward artificial neural network with six sub networks (one for each class) is used for classification.

E. Clustering approach: It uses a k-means based classifier. An unlabeled feature vector is assigned to the most common class of its three-nearest neighbors. Using clustering and three-nearest neighbors is certainly more powerful than simply using a single nearest neighbor, and it still has a low computational complexity. Clustering was performed on 500 samples, each labeled as either a whorl, left loop, right loop or arch. The features used were the orientation vectors in the area surrounding a fingerprint's core. Through experimentation, the authors found that using nine clusters had the best performance, and these clusters were found using a k-means clustering algorithms.

F. Using multi-space KL transform: The KL transform reduces the dimensionality of a feature space while minimizing the average mean-squared error. The multi-space KL (MKL) transform is a generalization of the KL transform that uses multiple subspaces for classification [68]. One subspace is trained for each fingerprint class and fingerprints are characterized by their distances to the subspaces. MKL has a strong ability to distinguish the fingerprint classes.

G. Support vector machines (SVMs): SVMs are based on statistical learning theory. SVMs are binary classifiers that work by finding the optimal separating hyper plane in the feature space [69]. One advantage of SVMs is their strong ability to classify vectors with high-dimensions. SVMs are applied to the problem of fingerprint classification using the FingerCode representation of the fingerprint. SVMs are a powerful classifier and good results were presented.
Hybrid classifiers: Uses a two-stage classifier based on FingerCodes. The algorithm first uses a k-nearest Neighbour classifier to determine the two most likely classes of the fingerprint. SVMs have been shown to be well suited for classifying FingerCodes [69], so by using SVMs instead of neural networks the accuracy of this system may be improved further. Classes are determined by the two most common classes of the k-nearest neighbours to the vector in the feature space. During the second stage, the fingerprint's class is determined by a neural network trained specifically to distinguish those two classes.

3.3 Proposed Method of Fingerprint Classification and Verification

3.3.1 Background of our work

Real-time image quality assessment can greatly improve the accuracy of identification system. The good quality images require minor preprocessing and enhancement. Conversely, low quality images require major preprocessing and enhancement. To test a fingerprint recognition algorithm, large databases of sample images are required to estimate error. But collecting large databases of fingerprint images is not a trivial task both in terms of money and time. An automatic recognition of people based on fingerprints requires that the input fingerprint be matched with a large amount of fingerprints in a database. To reduce the search space and hence the computational complexity, it is desirable to classify these fingerprints in an accurate and consistent manner so that the input fingerprint be matched against an appropriate subset of fingerprints in the database.

3.3.2 Proposed method

3.3.2.1 Conceptual Framework: The flow diagram of the proposed fingerprint verification method is depicted in Figure 3-1.
3.3.2.2 Feature Extraction: One of the fundamental steps before classification is the core point extraction. This step is particularly important since a reference center is required in order to correctly compare two fingerprints. Automated core detection can only find the most likely center of the image without regard whether there is a meaningful core exists or not. Furthermore, the alignment according to the core point only partially remedies the misalignment of two fingerprints.

The fingerprint feature vector generation phase consists of five steps. Initially the block direction of the image is estimated. Next, the certainty value associated with each block is calculated. Then the segmentation of the fingerprint image is carried out so that noisy and corrupt parts that do not carry valid information are deleted. Meanwhile, the core point to be taken as the reference center is extracted. Then a 16x16 block direction surrounding the core are identified towards construction of the feature vector.

i) Block Direction Estimation: The block direction estimation program operates on the gray level fingerprint image and then obtains an approximate direction for each image block with size 16x16. The technique found in [71]
was used to calculate the horizontal and vertical gradients of each pixel and then combines all the gradients within the block to get an estimated direction. It has large consistency with the ridge flows of the original fingerprint image in most testing cases.

ii) Finding Certainty Values: The certainty values are computed simultaneously during the direction estimation as adopted by [64]. The certainties are the magnitudes of vectors representing the flow directions of ridges in a grid for the extracted information on flow direction for the grid. The certainty vectors are of the same size as the block directional image. All the directions are then normalized into the domain from 0 to, and the certain values are in the interval from 0 to 1.

iii) Segmentation: An inevitable problem after step I is that after discarding the image areas the background noise may not be eliminated completely. It is needed to extract the tightly bounded fingerprint region from the image. To achieve high accuracy we accomplish (i) the segmentation task by techniques like histogram equalization, image enhancement and coarse segmentation by Fourier transform, image binarization and (ii) interesting region location by morphological operations.

iv) Finding of core or reference point: This step is particularly important since a reference center is required in order to correctly compare two fingerprints images. We used a variant of [71] to detect the core point. Unlike [71] map all the block directions to an interval from 0.5 to 05.8 our experimentation, we consider 0.5 or the corresponding value of the core.

v) Regulate the feature vector: we extract a 16x16 block centered at the core and then reconstruct it as a 1x256 vector. The parts of the block in the background region or outside the image region will not affect the vector values in further process. The same operations are enforced to the certainty vector.

3.3.2.3 Classification: Fingerprint classification is a technique to assign a fingerprint
into one of the several pre-specified types. Fingerprint classification can be viewed as a coarse level matching of the fingerprints. An input fingerprint is first matched at a coarse level to one of the pre-specified types and then at a finer level, it is compared to the subset of the database containing that type of fingerprints only. Fingerprints are classified into six categories: arch, tented arch, left loop, right loop and whorl and twin loop as in the Figure 3-2. Details

Figure 3-2: Fingerprint Classification

about the coarse level classification and fine level matching is reported below.

A. Coarse level classification using MSOM

Algorithm for self organizing map: Assume that output nodes are connected in an array of and the network is fully connected (all nodes in input layer are connected to all nodes in output layer), as shown in Figure 3-3. In this type of neural network, the learning rate is kept large at the beginning of training epochs and decreased gradually as learning proceeds. SOM has been used in this work as a basic classifier, where each fingerprint image is described by 256x256 pixels divided into 16x16 blocks. The orientation of each block is used as input to the neural network so the input layer consists of 256 nodes. Each image can fall into one of the
Figure 3-3: Basic structure for a well-trained fingerprint SOM, where winning node is 2, so the input vector X is of class 2.

five classes (right loop, left loop, whorl, arch, tented arch), so the output layer consists of five nodes. The learning rate which we used is $\alpha = 0.5$ and neighborhood radius $R = 0$. This process is designed to facilitate the identification process whenever the system is used, thus the searching time will be reduced as the system search only in one cluster. The learning algorithm [64], [65] of such network can be described as in Figure 3-4.

SOM construction and training Steps are as follows.

The neighborhood function is a window centered around the winning node $d_{\text{min}}$, whose radius decreases with time. In the implementation, the radiiuses are simply set to decrease from the map size $m$ to 1 during all the K runs. The
learning rate function $L(t)$ is also a decaying function. It is kept large at the beginning of training and decreased gradually as learning proceeds.

1. Construct an $m \times m$ SOM and initialize all the weights.
2. Input a fingerprint vector: $\{x_1,x_2,...,x_{256}\}$.
3. Find the winning node $d_{\text{min}}$,
$$d_{\text{min}} : = \min \{ ||x - w_i|| \},$$
where $||.||$ denotes the Euclidean norm and $w_j$ is the weight vector connecting input nodes to output node $j$.
4. Update the weight vectors:
$$w_{ij}(t+1) = w_{ij}(t) + L(t)[x_i(t) - w_{ij}(t)]N(j,t)x_{c_i}$$
where $w_{ij}$ is the $j^{th}$ component of the weight vector $w_j$,
$L(t)$ is the learning rate and $N(j,t)$ is the neighborhood function.
5. Repeat steps 2-4 till update is not significant.

Figure 3-4: Algorithm for the conventional SOM

Instead of original SOM algorithms for training and classification, the modified SOM algorithm using certainty vector as parameter, as depicted at [64] is also used for classification. Here each fingerprint is associated with a certainty vector $c$. The steps are shown in Figure 3-5.

1. Construct an $m \times m$ SOM and initialize all weights.
2. Input a fingerprint vector:
$$X_c\{x_1,x_2,...,x_{256}\} = cx + (1-c)x_{avg}$$
where $x_{avg}$ is the vector holding the average values $x_k$ ($k$ from 1-256) over the whole training sample space.
3. Find the winning node $d_{\text{min}}$ where:
$$d_{\text{min}} : = \min \{ ||c(x-w_j)|| \}$$
4. Update the weight vectors:
$$w_{ij}(t+1) = w_{ij}(t) + L(t)[x_i(t) - w_{ij}(t)]N(j,t)x_{c_i}$$
where: $w_{ij}$ is the $j^{th}$ component of the weight vector $w_j$.
$L(t)$ is the learning rate and $N(j,t)$ is the neighbor hood function.
5. Repeat 2-4 till Update is not significant.

Figure 3-5: Algorithm for the conventional MSOM
A free SOM toolbox [68] is adopted to assist in the classification.

Result of SOM and MSOM: For coarse level classification we have used FVC2000 and FVC2004 datasets and the results are shown in Table 3.1

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Training</th>
<th>Testing</th>
<th>5 x 5 map</th>
<th>8 x 8 map</th>
<th>8 x 8 map</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOM</td>
<td>60</td>
<td>20</td>
<td>88%</td>
<td>86%</td>
<td>92%</td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>40</td>
<td>90%</td>
<td>91%</td>
<td>94%</td>
</tr>
<tr>
<td></td>
<td>600</td>
<td>40</td>
<td>91%</td>
<td>93%</td>
<td>94.6%</td>
</tr>
<tr>
<td>MSOM</td>
<td>60</td>
<td>20</td>
<td>89%</td>
<td>92%</td>
<td>93.47%</td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>40</td>
<td>91%</td>
<td>92.2%</td>
<td>95%</td>
</tr>
<tr>
<td></td>
<td>600</td>
<td>40</td>
<td>93.3%</td>
<td>94%</td>
<td>95.52%</td>
</tr>
</tbody>
</table>

B. Fine Level grouping using Minutiae Clustering: Here the steps followed are

**B.1 Minutiae Extraction:** We have used CUBS fingerprint feature extraction tool [73] for minutiae feature extraction. This tool provides a graphical user interface for minutiae feature extraction and visualization. It also allows the user to manually identify new minutiae or remove spurious ones.

**B.2 Minutiae clustering for fingerprint similarity metric:** After extracting the feature point set \( N \) from a fingerprint, we have clustered the feature points using Kmeans algorithm with variable number of clusters. The procedure minutiaeCluster is shown in Figure 3-6.

**Procedure minutiaeCluster( )**

*Input*: Fingerprint minutiae \( M_t \), core point \( P_c(x, y) \), cluster no. \( m \)

*Output*: \( k \) representation of minutiae groups

1. For \( i = 1 \) to \( n \)
2. Read the minutiae \( M_t \) for each class of fingerprints given by MSOM
3. Call KmeansFing(\( M_t, m, P_c(x, y) \)) to identify \( k \) groups;
4. Next i.
5. Next, we report our minutiae graph generation technique.

Figure 3-6: Procedure for minutiaeCluster( )
### B.3 Graph generation over clustered minutiae space

After obtaining the clustered minutiae space, cluster-graphs are generated.

**Procedure minutiaeGraph()**

**Input:** Cluster centroids $C_i(x,y)$

**Output:** Centroids distance matrix, $D_{ij}$ and graph plot.

```
1. For i=1 to m do
2. Read the centroids $C_i(x,y)$;
3. For j=1 to m do
4. Call DistMatrix$(C_i,C_j)$; to perform Euclidean distance
5. Next j
6. Next i
7. For i=1 to m do
8. For j=1 to m do
9. Call minDist$(D_{i,j})$
10. Draw graph $D_{i,j}$
```

#### Figure 3-7: Procedure for minutiaeGraph()

### 3.3.2.4 Index Generation:

In the proposed system, a SOM based classification approach [64], is used for coarse level classification whereas, a graph-theoretic approach is used to analyze the process of fingerprint comparison for finer level matching. Finer level matching is supported by extraction of minutiae i.e. ridge ending and branching points.

From the minutiae graph, an index is generated for each fingerprint, which is unique, i.e., no two fingerprint images will have the same index. An index is generated based on four parameters:

1. **Number of vertex**
2. **Degree of each vertex**
3. **Highest degree.**
4. **Number of vertices with same degree.**
The generated index for each minutiae graph is then saved to the database.

3.3.2.5 Matching: Fingerprint matching is done at two levels. At coarse level, fingerprints are classified into whorl, arch, tented arch, left loop, right loop and twin loop. Coarse level classification is good only for faster detection of the class type of a given input fingerprint. At finer level, matching is performed based on the minutiae (i.e. ridge ending and branching points) information.

3.4 Transformation Invariance

The proposed graph based index can be found to be advantageous due to its transformation invariance. To establish the proof of transformation invariance we adopt two approaches i.e. (a) graph-based and (b) distance-based. Next we report each of these approaches.

(a) Graph-based approach We are interested in establishing that two fingerprints which are similar must be matched correctly. To do that we observe the differences comparing the minutiae cluster graphs due to small perturbations in the seed points. To accomplish this, we consider the graph obtained from the minutiae feature point clusters.

Definition I: MinutiaeGraph The graph obtained from the minutiae clusters is referred as the MinutiaeGraph. This graph contains all the topological properties of the minutiae clusters. It tells which vertices are connected, but it does not contain any geometric measures, such as the lengths of the edges and angles they form at each vertex.

Definition II: Fingerprint Equivalence: Two fingerprints F and F/ are equivalent, i.e. \( \text{diff}(F,F/) = 0 \), if their respective MinutiaeGraphs are isomorphic, else they are distinct.

Using isomorphism classes of minutiae graphs to compare fingerprints has the following advantages
1. No effect on the comparison, due to any unintentional rotation, translation, or scaling factor while recording a fingerprint.

2. A slight perturbation in the location of minutiae points will result in a topologically equivalent minutiae graph.

The cluster graphs of each individual impression are tested for isomorphism, proving their transformation invariance. The graphs obtained from two types of fingerprint, each having three impression of each; represent the isomorphic graphs as shown in figure 3. Next we report the distance based approach.

(b) Distance based approach A distance measure for the purpose of object matching should have the following properties.

(1) It should have a large discriminatory power.

(2) Its value should increase with the amount of difference between the two objects. The operation of image matching consists of computing a measure of similarity between two images based on their features.
We have used Hausdorff distance and Modified Hausdorff distance (MHD) [74], [75] between two sets of minutiae maps (points) associated with the fingerprint and thus proving the invariance of different fingerprint impressions. Based on [74] we define the Hausdorff distance as follows.

**Hausdorff distance:** Given two finite point sets \( M_{m_1, m_2, \ldots, m_p} \) and \( N_{n_1, n_2, \ldots, n_q} \), the Hausdorff distance is defined as

\[
H(M, N) = \max(h(M, N), h(N, M))
\]

where

\[
h(M, N) = \max_{m \in M} \min_{n \in N} \|m - n\| \quad (3.1)
\]

and \( \| \cdot \| \) is the underlying norm on the points of \( M \) and \( N \). The function \( h(M,N) \) is called the directed Hausdorff distance from \( M \) to \( N \). \( h(M;N) \) in effect ranks each point of \( M \) based on its distance to the nearest point of \( N \) and then uses the largest ranked such point as the distance. The Hausdorff distance \( H(M,N) \) is the maximum of \( h(M, N) \) and \( h(N, M) \).

Thus it measures the degree of mismatch between any two shapes described by the sets \( M \) and \( N \). Our choice of Hausdorff distance is based on its relative insensitivity to perturbations in feature points, and robustness to occasional feature detector failure or occlusion [74].

![Figure 3-9: The directed Hausdorff distance is large just because of a single outlier.](image)
3.5 Performance Evaluation

To evaluate the performance of the reported fingerprint classifier, we have tested the false accept rate (FAR) against the accuracy of the method. FAR is a measure of the fingerprints that are accepted by a certain fingerprint class, while not belonging to that particular class. An example of an event that increases the FAR of a RL (right loop type) class is a non-RL, for instance LL, fingerprint being classified as a RL fingerprint. The FAR of a particular fingerprint class is mathematically modeled as in equation 3.2. The details of the environment used, dataset used and results are as follows:

(a) Environment used: The experiment was carried out on a workstation with Intel Dual-Core processor (1.86 GHz) with 1 GB of RAM. We used MATLAB 7.2 (R2006a) version in windows (64-bits) platform for the performance evaluation.

(b) Datasets used: In order to evaluate the performance of the classifier, dataset is used from FVC2000 and FVC2004. Also we have used a synthetic fingerprint generator SFinGe [89] to create at zero cost, large databases of fingerprints, thus allowing recognition algorithms to be simply tested and optimized. This synthetic fingerprint generator captures the main interclass and intra-class variations of fingerprints in nature are well enough [71]. The image size is of 300 x 300 pixels.

(c) Experiment Result and analysis: From the result of our experiment we have obtained and hence proved as shown in the Figure 3-8, the following two lemmas.

Lemma1: Graph-based feature index for any fingerprint image remains invariant subject to translation.

Lemma2: Graph-based feature index for any fingerprint image remains invariant subject to any rotational transformation.

\[
FAR = \left( \frac{F}{S} \right) \times 100 \quad (3.2)
\]

Where \( F \) is the total number of fingerprints that are wrongly accepted and \( S \) is the total number of fingerprints that are to be recognized. For a good fingerprint
classifier, the average FAR value should approach 0% and the value less than 20% are sufficient. The results obtained can be depicted as in the Figure 3-8. From the figure we can see that our method has both better performance and efficient, therefore is more suitable for fingerprint verification application. As the FAR value approaches to 0%, the accuracy level approaches to 100%

![Figure 3-10: FAR vs. Accuracy of MSOM based fingerprint classification](image)

3.6 Discussion

A limited survey on some of the popular fingerprint classification methods was carried out and found capable of identifying four or five classes with an accuracy level of (80-95)%.

The proposed fingerprint classification method works with an accuracy level of 95.52% for coarse level classification and presents a new approach for graph based fine level matching of fingerprints and their template generation. Also we have reported two techniques, viz. graph based and distance based approach, for proving robustness of various fingerprints.

Although fingerprint classification and matching techniques have developed drastically over times, there are scopes for developments which will make the process more efficient and accurate. A multiple SOM based approach can be used to enhance the performance.
Next chapter describes an iris authentication scheme and its experimental results.