Personal Multimodal Biometric Authentication Using Unsupervised Learning, Hidden Markov Model (HMM)

Major project report submitted in partial fulfilment of the requirements for the award of degree of

Master of Technology

In

Information Systems

Submitted By

ANKITA GUPTA

(2K11/ISY/02)

Under the Guidance of

Sh. Anil Singh Parihar

(Assistant Professor, Department of Information Technology)

DEPARTMENT OF INFORMATION TECHNOLOGY

DELHI TECHNOLOGICAL UNIVERSITY

BAWANA ROAD, DELHI-110042

SESSION 2011-13

CERTIFICATE

This is to certify that **Ms. Ankita Gupta (2K11/ISY/02)** has carried out the major project titled **Personal Multimodal Biometric Authentication Using Unsupervised Learning, Hidden Markov Model (HMM)** as a partial requirement for the award of Master Of Technology degree in Information Systems by Delhi Technological University.

The major project is a bonafide piece of work carried out and completed under my supervision and guidance during the academic session 2011-2013. The matter contained in this report has not been submitted elsewhere for the award of any other degree.

(Project Guide)
Sh. Anil Singh Parihar
Assistant Professor
Department of Information Technology
Delhi Technological University
Bawana Road, Delhi-110042

ACKNOWLEDGEMENT

It gives me a great sense of pleasure to present the thesis of M. Tech Major Project undertaken during M. Tech. Second Year. I owe special debt of gratitude to **Sh. Anil Singh Parihar,** Department of Information Technology, and Delhi Technological University, Delhi for his constant support and guidance throughout the course of my work. His sincerity, thoroughness and perseverance have been a constant source of inspiration for me. It is only his cognizant efforts that our endeavours have seen light of the day. I humbly extend my words of gratitude Prof. O. P. Verma, HOD, Department of Information Technology and other faculty members of the department for their valuable time and help when required.

Ankita Gupta

Roll No. 2K11/ISY/02

M.Tech (Information Systems)

Email: ankita.gupta.mail@gmail.com

ABSTRACT

Biometric authentication systems have been used since decades. Palmprint and finger knuckle prints are two such modalities that are universal and possess uniqueness. A variety of algorithms are available to extract features from these modalities and do the authentication process. In this report, use of a machine learning, unsupervised Hidden Markov Model algorithm is proposed to classify the users into genuine and imposter classes. In the following report, a multimodal system using palmprint and finger knuckle print has been proposed using a combination of Harris Corner Detector; SIFT descriptors and Continuous Density Hidden Markov Model (CDHMM). Here the states defining the origination of the observation feature vectors are hidden. The features are extracted using Harris Corner Detector and are described using Scale Invariant Feature Descriptor (SIFT). An approach is proposed to do the authentication at feature level as well as at score level. The log-likelihood computed by HMM and the parameters are maximised by Expectation-Maximization Algorithm. An iterative approach is used to increase the authentication rates and to get the correct number of states in each Hidden Markov Model of each user at feature level and for genuine and imposter classes at score level. The various fusion methods at score level are experimented for the PolyU, IITD palmprint and PolyU finger knuckle print database. The authentication rates obtained are as high as 99% GAR at 0.01 FAR for PolyU palmprint database that are comparable to other methods of authentication at score level. The highest GAR was recorded using SUM fusion rule. The authentication rates are high for feature level authentication as well for both knuckle prints and palmprints. GAR was recorded as high as 97% for right middle knuckle finger print at 0.01 FAR.

Table of Contents

Chapter 1: Introduction

Conventional personal authentication systems that are based on knowledge (e.g., password) or physical tokens (e.g., ID card) are not able to meet strict security performance requirements of a number of applications that are available today. These applications generally make use of computer networks and thus affect a large number of people, and have control over financially valuable and privacy-related projects (e.g., e-commerce). And thus these authentication systems are vulnerable to imposter attacks. Biometric authentication systems provide good alternative for such conventional authentication methods by providing high performance rates.

Biometric authentication refers to the automatic identification of a person based on human physiological or behavioural or chemical characteristics or traits. Fingerprints, iris, palmprints are all covered under physiological characteristics. Gait, keystrokes are behavioural traits of any human. Human perspiration is a chemical trait which is comparatively less used for authentication. Any human trait which has uniqueness, distinctiveness, permanence and can be easily collected can be used as biometric trait. Biometric has found great use in defence sector, civil applications, preventing financial frauds and forensics.

1.1 Biometric Systems:

Each user must be first enrolled in the system by collecting template images of the user and storing it in the database. This template is then used for future use, when the individual needs to be identified. In early years of biometric, templates were captured by offline modes. In offline modes previously captured images are only processed. These offline modes generally include inked acquisition of images. These inked images were then converted digital images. These methods were mostly restricted to finger print only. With the advent of new technologies, offline capturing modes were over ruled by online modes as they were not well suited for real time recognition systems [29]. Online system includes various sensors to capture the images. There are specialized systems for capturing the image of the modality. Like for capturing knuckle print there is a triangular box that allows the user to keep his/her finger on this block in a certain manner, allowing the capturing device to capture the image [12].

In any biometric system, every captured image goes through the following processes: Pre-processing, Feature Extraction and Feature Matching. Pre-processing step is basically extraction of Region of Interest (ROI). The image captured by the imaging device may contain area that may not be of importance for the purpose of recognition, in that case only the region of interest should be extracted. The extraction method differs with each modality.

Once the ROI has been extracted, features are extracted. Sometimes, before extracting the features the image quality is enhanced so that good features can be extracted [30]. Different features are extracted using different algorithms. The algorithm varies with the type of information to be extracted. Depending on the feature to be extracted, there are algorithms like PCA, LDA, DCT, Gabor Filter [1, 13, 14] etc. After getting the feature vector of an image to be tested, the matching is done is between the captured image and the templates stored in the database. If the captured image matches with any of the images stored in the database, it is a genuine user else it is an imposter. A number of matching techniques are available, like Euclidean Distance, Cosine Similarity, and Hamming Distance etc.

The biometric authentication systems have extended from finger and palmprints to face, iris, gait and other modalities. Use of more than one modality has also been considered a very good alternative to have high recognition rates and thus less imposter attacks. In addition, failure to enrol situations can also be reduced in multimodal biometric systems [3]. Also, physically challenged people may use multimodal biometric systems for authentication. Every modality has its own pros and cons. Though fingerprint is the most widely used modality, palmprints and knuckle prints also offer very high recognition rates in real time recognition. Some of the few biometric modalities are listed below.

1.2 Few Biometric Modalities:

1. **Palm:** It is the ventral surface of hand, representing area between wrist and the root of the fingers. A palmprint basically includes textures, wrinkles, principle lines and ridges. These features are extracted using various feature detection techniques [1, 2]. The palm is a universal trait and possesses uniqueness in terms of the features it contains. It provides a very large area as compared to finger prints and hence more number of features can be extracted from the palmprints. There may be cases where the principle lines of two different users may match, so extracting features like principle lines for authentication is not recommended. So extracting point features

using algorithms like Harris is more efficient than using principle lines as features. These features differ with every other user and hence are helpful in recognition.

2. **Finger knuckle:** It refers to the inherent skin pattern of the outer surface around the phalangeal joint of one's finger [12]. Finger knuckle is user centric, contactless and unrestricted access control [22]. Finger knuckle prints have high textural region and hence good quality features can be extracted from the knuckle prints. Moreover, every user can provide a large number of samples from all the fingers of a human hand. Four samples can be collected from middle and index finger of both left and right hands of human. The knuckle prints are have their own uniqueness and hence are used for user recognition.
3. **Face:** Facial images are the most common biometric characteristic used to make personal recognition. Face recognition is a nonintrusive method. The approaches to face recognition are based on (a) the location and shape of facial features such as the eyes, eyebrows, forehead, nose, moustache, lips and chin, and their spatial relationships, or 2) the overall analysis of the face image that represents a face as a weighted combination of a number of canonical faces [27]. There are certain considerations with which a system shall work so that facial recognition system can work well in practice. The system shall automatically (a) detect whether a face is present in the acquired image; (b) locate the face if there is one; and (c) recognize the face from a general viewpoint [10].
4. **Gait:** Gait refers one's own unique way in which he walks. It is a complex spatial-temporal biometric. In various low security applications, gait can be used for verification. Gait based systems are input sensitive and are computationally expensive as it uses the video-sequence footage of a walking person as the input, where various movements of each articulate joint are measured [28]. However, there is a difficulty in this system that the consistency in the input may not maintained over a long period of time, as there can be variations in body weight, injuries which may affect the joints etc.
5. **Finger:** A finger print is a unique pattern of ridges and valleys on the surface of the fingertip. Fingerprints of each finger of the same person are different. The finger prints of identical twins too are different [10]. That is why the matching accuracy of fingerprints is very high. Though it faces some problems at the time of authentication like that of bruises, cuts and other burn marks.

1.3 Operating Modes:

A biometric system can operate in three modes:

1. **Enrolment:** When a user is using the biometric system for the first time, he/she must be enrolled in the system. A biometric sensor is used for acquiring the user's biometric information. The data so acquired is checked for its quality and the relevant features are thus extracted out of it using various feature extracting algorithms [31]. This feature vector data is stored in a database along with user's identity and other necessary information for further Authentication process in the later stages.

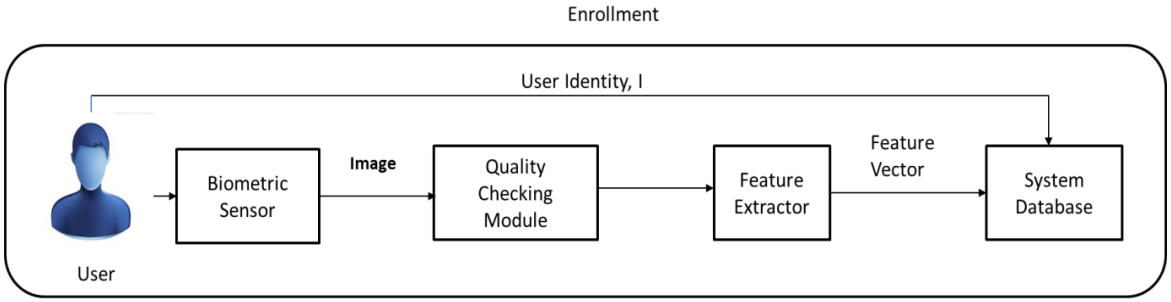

Figure 1.1: Enrolment Mode

2. **Identification:** Identification, one of the functionalities provided by a biometric system is one to many (1: many). This mode tells the person "Who am I..?" The data is acquired again for authentication of the data. There can be both positive and negative identification. A positive identification system determines the identity of the user from a known set of identities and thus tells the user if he is known to the system without explicitly claiming an identity. However, in Negative identification, also known as screening, the user conceals his true identity. The main objective of Negative identification is to find out whether the identity matches to the provided information about the user. It can be used in preventing issues of multiple credential records for one person and also in preventing from claiming multiple benefits under multiple names. In both the positive and negative identification, the captured image is matched with all the templates stored in the database. As a result, the system then gives the identity of the individual with the highest degree of template similarity with the input provided, or otherwise, declares that the user is not enrolled, if any kind of similarity does not exist [32].

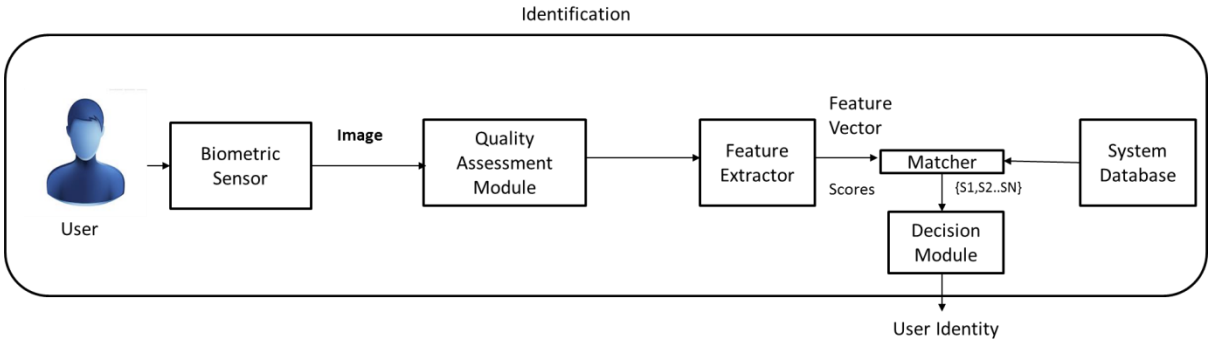

Figure 1.2: Identification Mode

3. **Verification:** Verification equates to "Be I who I claim to be". In this one to one (1:1) approach is followed. It is a comparison with the template of that user who she/he claims to be [9]. It involves comparison with templates corresponding to the claimed identity. In the verification process the system verifies the genuineness of the claim of the user, what he claims his identity to be. Here the captured data is compared with the template corresponding to claimed identity only. If there exists a high degree of similarity, the claim is said to be "genuine" and is accepted, else, is said to be "impostor" and rejected [32, 34].

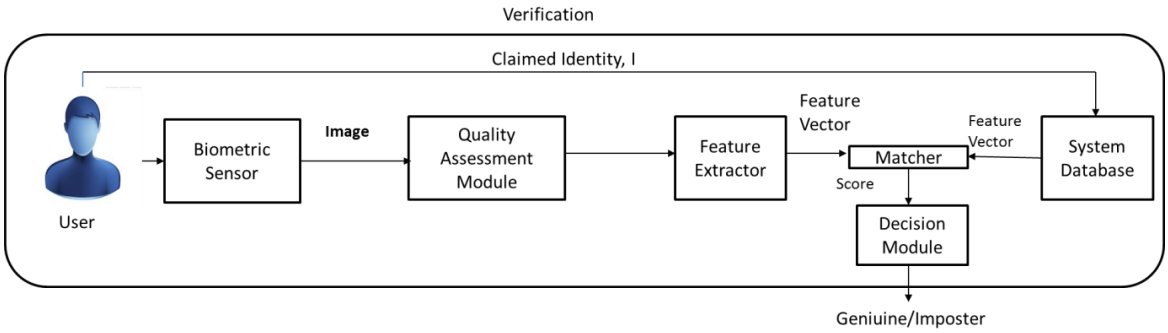

Figure 1.3: Verification Mode

1.4 Multimodal Biometric Systems:

A multimodal biometric system is much more preferred than unimodal systems. They have many advantages over the systems that employ only one modality in the authentication process. Some of the advantages are: they can increase the authentication rate, they can address problems of non-universality and thus decreasing FTE. Moreover, multiple data reduces the effect of noisy data [32]. They are more resistant to spoof attacks. Though they provide a lot of advantages over unimodal systems but face few limitations also like they are

expensive and require more number of computations. Fusion of modalities can be done at sensor level, at feature level, at score level or at decision level [9].

Sensor Level Fusion: The raw data from the sensors are combined in the sensor level fusion, this combining of sensory data or data derived from sensory data such that the resulting information is in some sense better than would be possible when these sources were used individually is sensory fusion. In this level, the input signal is the result of sensing a biometric data with two or more sensors [31, 39]. The raw data which is acquired from the multiple sensors is processed to generate new data. Features can hence be extracted from this new data. For example, in case of face biometrics, both 2 dimensional texture information and 3 dimensional depth information can be synthesised together to form a 3D model of the face or a panoramic face mosaic. This panoramic face mosaic could then be subjected to feature extraction and matching.

However, sensor level fusion is restricted by certain limitations. It can be performed if

(a) The sources are the samples of same biometric trait obtained from multiple compatible or

(b) Multiple instances of the same biometric trait generated using a single sensor.

In sensor level fusion, the correspondence between points in raw data must be known in advance, if not known in advance; they shall be reliably estimated [32].

Feature Level Fusion: Feature level fusion means fusing different feature sets, which have been extracted from multiple biometric sources, to represent an individual. Within this process, the matching features are initially extracted from the biometric sample, then fusion is used to synthesise all the features into one biometric data. This level can be applied to extraction of features from unimodal system as well as multimodal system.

A unimodal system is the one where different features are extracted from a single biometric signal, whereas in multimodal system, combinations of feature levels are extracted from different biometric characteristics [25].

A single feature set is can be calculated as a weighted average of the individual feature sets when the feature sets are homogenous [33]. For example, using the multiple fingerprint impressions of a user's finger can be used to generate a mosaic of fingerprint minutiae. However, when the feature sets are non-homogeneous, they can be concatenated to form a single feature set. For example, biometric modalities like face and hand geometry may be augmented with the Eigen-coefficients of the face in order to construct a new high-dimension feature vector Feature selection schemes can then be applied to reduce the dimensionality of

the resultant feature set, i.e. a minimal feature set can be thus obtained from the high dimensional feature vector.

If the features set are incompatible to each other or the multiple feature sets correspond to different samples of the same biometric, it is not possible to concatenate, as feature space of every biometric trait varies

Score Level Fusion: Score level fusion also known as match score measures the similarity between the input and template biometric feature vectors. Fusion is said to be done at score level when there is consolidation of the match scores output by different biometric matchers, which is then fused to generate a single scalar score at the final recognition decision. For example, the match scores generated by the face and hand modalities of a user may be combined via the simple sum rule in order to obtain a new match score which is then used to make the final decision [31]. Techniques such as logistic regression may be used to combine the scores reported by the two sensors. Match scores fused by the individual matchers need not necessary be homogenous. Also for the match score level, the outputs of the individual matchers need not be on the same numerical scale.

.

Decision Level Fusion: Decision-level fusion involves fusion of sensor information that is preliminary determine d by the sensors. Each sensor can capture multiple biometric data and the resulting feature vectors individually classified into the two classes—accept or reject. The classifier gives its decision regarding the presence absence of a genuine individual. The decisions from multiple classifiers are then fused in order to generate the final decision. When each matcher outputs its own class a single class label can be obtained. Various techniques like majority voting, behaviour knowledge space, etc. may be employees to arrive at the final decision. Some of the decision level fusion methods include Bayesian inference, classical inference, weighted decision methods and Dempster–Shafer method [35, 36].

Out of all these fusion levels, score level has been considered most promising.

Some score fusion rules:

1. **Sum Rule:** It is one of the most efficient fusion rules. The sum rule involves summing up of all the scores obtained from different traits/modalities. The result obtained is the fused score ($score_{fusion}$). In the given formula, *k* is the number of traits.

$$score_{fusion} = \sum_{n=1}^{k} S_n \quad (1.1)$$

2. **Min Rule**: Min rule is one of the score fusion rules. In this rule, the minimum of all scores is taken as the final score for making decision.

$$score_{fusion} = \min(S_1\ S_2\ \ldots\ldots S_k) \quad (1.2)$$

3. **Max Rule**: Max rule is a rule in which maximum of all the scores is taken as the score for decision.

$$score_{fusion} = \max(S_1\ S_2\ \ldots\ldots S_k) \quad (1.3)$$

4. **Product Rule**: Product Rule is one of the least used fusion rule. In this rule, the scores are multiplied, and the result is used as the final fused score.

$$score_{fusion} = \prod_{n=1}^{k} S_n \quad (1.4)$$

1.5 Biometric Performance Measures:

In any biometric system, any two samples of the same biometric modality of single user, cannot match exactly. They do differ due to imperfect imaging conditions, changes in user physiological or behavioural traits, ambient conditions etc. and thus results in some errors in the authentication [10]. When a user is who is an imposter and is genuinely accepted by the system then such an error is known as failure to match or false accept. When a genuine user and is rejected by the system then such an error is called failure to non-match or false reject. A threshold *t*, a parameter decides the trade-off between **False Accept Ratio (FAR) and False Reject Ratio (FRR).** If more secure system is required then threshold is increased and as a result FRR is also increased. When the threshold is decreased, FAR is increased. The **Genuinely Accepted Rate (GAR)** of any biometric system is given by (1-FRR) or (100-FRR), whichever is applicable.

1.6 Machine Learning in Biometrics:

Machine Learning has also been used in biometrics to classify the users in the genuine and imposter classes. Machine learning can be categorised into supervised and unsupervised

learning. In case of supervised learning, the model defines the effect that one set of observations, which are known as inputs, has on outputs which are another set of observations. The inputs are provided in the beginning and outputs at the end of the causal chain. The models can include intermediating variables between the inputs and outputs. In such cases everything is known beforehand. The leading variable that cause the input observations are known. In real practice, models for supervised learning often leave the probability for inputs undefined. There are Supervised algorithms like GMM, Bayesian Networks, and Decision trees [15-17, 19] etc. which have been used in biometrics.

While, in unsupervised learning, all the observations are assumed to be produced by latent variables, that is, the observations are assumed to be at the end of the causal chain. This model is not required as long as the inputs are available, but if some of the input values are unknown, it is not possible to infer anything about the outputs. If the inputs are also fabricated, then missing inputs cause no problem since they can be considered latent variables as in unsupervised learning.

Unsupervised algorithms like Hidden Markov Model, Expectation Maximization algorithm have also been used. Zheng *et al* utilized HMM using fusion of few face databases [18]. Zhang *et al* used Hidden Markov Model for palmprint authentication using Sobel operators aligned at different angles [20]. It has also been used in authentication using Keystroke Dynamics [21]. Originally, HMM was used in speech recognition process [4]. In case of unsupervised methods used in biometrics, and particularly Hidden Markov Model, less work has been reported. Moreover, it has been used only with very basic and simpler methods like observations obtained using Sobel operators. To increase the authentication rates using HMM, use of more complex features have been proposed in this thesis.

Chapter 2: ROI and Feature Extraction

Any biometric system does a few processing steps to authenticate the user. The image is captured using the acquisition sensors, is processed to get an image which is suitable for feature extraction. This processed image is actually the ROI of the captured image. This image is now stored in the database. The database contains all the images of the users those who need to get an access to another system by authenticating themselves.

Thus, in any biometric authentication system there are basic five phases.

1. Image Acquisition
2. Image Pre-Processing (ROI Extraction & Image Enhancement)
3. Feature Detection
4. Matching
5. Database

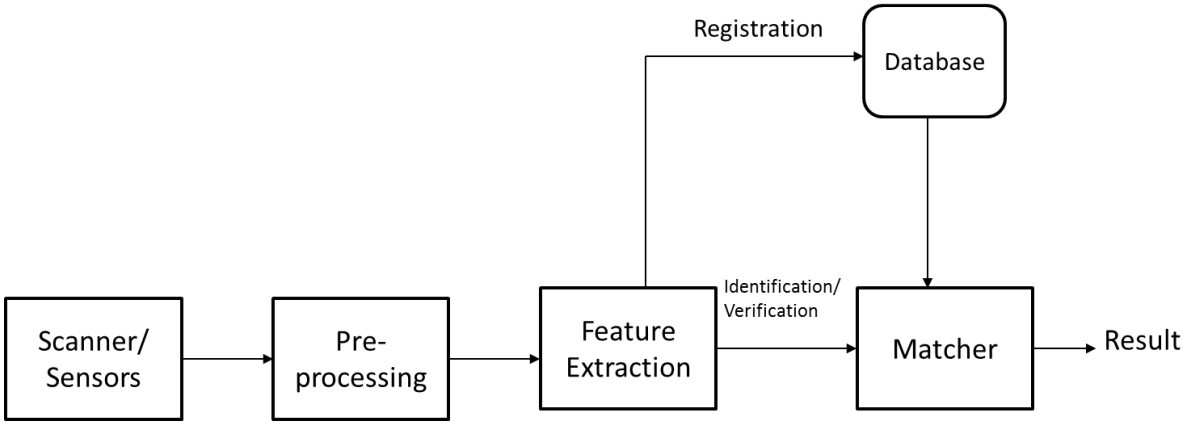

Figure 2.1: Steps in Biometric Authentication System

2.1 Image Acquisition:

Image acquisition is the first step in biometric authentication system. In this an image of the biometric trait is taken. This image is either stored in the database for future needs, or is used for matching purposes with the already saved images. This image acquisition should be accurate and clear enough that it gives good authentication results when required. There are special sensors to capture each type of biometric trait. Sometimes, a user may not be able to enrol or give a clear image of its trait due some reasons like environmental conditions leading to sweat on fingers and palms or due non availability of the trait. In such cases the image acquired by sensors are not much appropriate for the purpose of authentication. This results

in error which is known as Failure to Enrol (FTE) [34]. If a user is unable to enrol in the system properly, he/she may be rejected by the system even if he/she is a genuine user of the system. The image acquisition can be in a constrained environment or in an unconstrained environment. In case of constrained environment, the user is asked to place his hand or knuckle in accordance with the pegs placed on the acquisition device [12]. While in case of unconstrained acquisition, the user is free to place the hand, knuckle in any position. No pegs or triangular blocks are used in case of unconstrained environment. There are sensors which do not require touching the sensors to get their image acquired. Such systems are known as touch-less systems [37]. A biometric sensor can be defined as a security system device that captures unique physical traits with the help of digital technology.

There are various types of biometric sensors:

1. **Optical sensors**

   An optical sensor is a device that converts light rays into electronic signals. Like a photo-resistor, a physical quantity of light is measured and is translated into a form that is readable by the instrument. Generally, the optical sensor is part of a larger system which has a source of light, a measuring device and the sensor itself. The sensor is connected to an electrical trigger, which responds to a change in the signal which is observed within the light sensor and captures the image. Optical sensors are generally used in capturing fingerprints [38].

2. **Thermal sensors**

   Thermal sensors are devices used to measure the temperature of a trait. There are 2 kinds on temperature sensors: 1) contact sensors and 2) noncontact sensors. In these sensors, the palm is slide over the sensor to measure the temperature difference between the various portions of the palm. They are generally used in capturing faces [40].

3. **Capacitive sensors**

   Capacitive sensors are constructed from many different media, such as copper, Indium tin oxide (ITO) and printed ink. The image is captured against a surface of silicon integrated circuit. It is one of the most expensive and most sensitive sensors [41].

4. **Electric field sensors**

   The electric field sensors use the electric field that is formed between two conductive layers. An antenna is also there in case of electric field sensors which measures the

electric field. A field is created between the palmprint and the semiconductor that makes the image.

5. **Ultra sound sensors**

   An ultra sound sensor measures the properties of sound waves with frequency above the human audible range. They are based on three basic principles: the Doppler Effect, time of flight and the attenuation of sound waves. They use frequencies that are of very high range. These frequencies penetrate the epidermal layer of the skin and thus create an image. Such sensors are generally used to produce 3D image of any trait.

2.2 Image Pre-Processing:

The second step is pre-processing the captured image. The image that is being captured may contain some area/part that does not contain useful information. So, it is necessary to extract only that area from the image that contain the features and the useful information that may be used in authentication process. This area is known as Region of Interest (ROI). There are specific algorithms which have to be followed to extract ROI from the acquired images.

- **ROI Extraction of Palmprint:**

The image captured using the sensors of a palm may contain the extra area, captured along with palm of the user. It may contain outer black region, fingers that are not required in feature extraction. So such areas must be cropped out and only the Region of Interest must be stored in the database. Initially Gaussian filter is applied to the image and then followings steps are followed.

1. Binarization
2. Hand Contour Tracing
3. Region of Interest detection

1. **Binarization:**

A threshold (t) value is selected using Otsu's method which is used to binarize the image I (captured palmprint image) [11].

$$BI(i,j) = \begin{cases} 0 \; if \; I(i,j) \leq t \\ 1 \; otherwise \end{cases} \qquad (2.1)$$

where 0 means a black pixel and 1 means a white pixel.

BI is the binarized image obtained.

2. **Hand contour tracing:**

The contour of the hand image is obtained using contour tracing algorithm on binarized hand image.

Steps are:

1) **Find the start pixel:**

Hand image's boundary pixel can be chosen as a start point provided its left adjacent pixel is not an object. This can be done by starting at the bottom left corner of the image. This can be done by starting at the bottom left corner of the image, scanning each column of pixels from the bottom going in upward direction, starting from the leftmost column and then proceeding to the right. This process is done until an object pixel is encountered.

2) **Find next best pixel:**

Let p be the current pixel boundary pixel and p1, p2 and p3 and its adjacent pixels. If p1 is object's pixel than this will be the next boundary pixel, if p2 is object's pixel it will be next boundary pixel, if p1 and p2 are both boundary pixel than p3 will be next boundary pixel. In case all three neighbour pixels are boundary pixel, then the new set of neighbourhood pixels is considered. This procedure is considered until an object's pixel is found. In case there is no object's pixel in the neighbourhood the pixel is considered as an isolated pixel.

3) **Termination:**

The tracing is terminated if

- pixel is an isolated pixel or
- Current pixel is the starting pixel i.e. contour is traced.

1. **Region of Interest(ROI) detection:**

Finger tips and valley coordinates are found using local minima and maxima on contour image. Two reference points V1 valley between little finger and ring finger and V2 valley between index finger and middle finger are selected from these points. A line is drawn between V1 and V2 and a perpendicular is drawn from the midpoint to the reference point. An ROI of size 128x128 is then extracted.

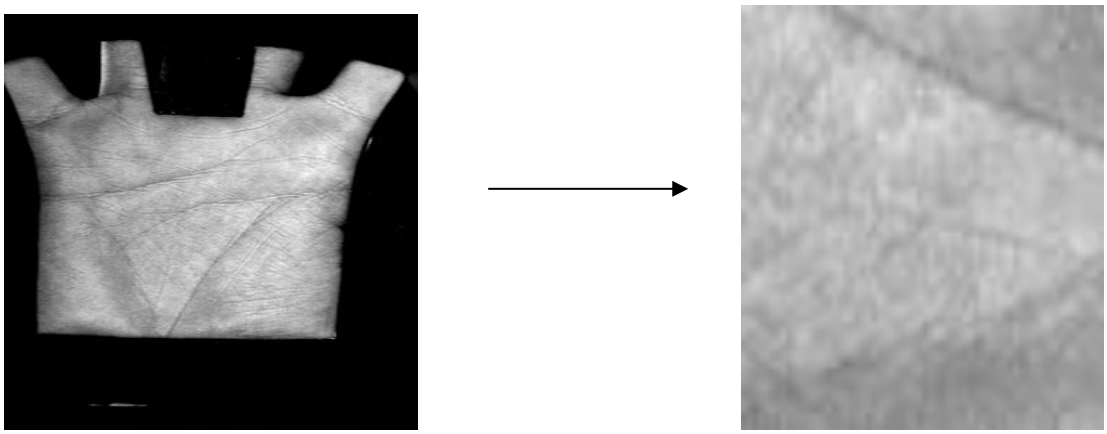

Figure 2.2: ROI extraction of palmprint image

- **ROI Extraction of Finger Knuckle Print:**

Similarly like palmprints, finger knuckle prints also contain some unwanted area that may hinder the authentication process. Finger Knuckle images may consist of other parts of finger including the knuckles. These other parts are undesirable and hence must be cropped out.

Finger Knuckle Print (FKP) images of different fingers are very different. Moreover, for the same finger, images collected at different sessions also vary because of the variation of spatial locations of the finger. Therefore, it is necessary to align FKP images by robustly constructing a local coordinate system for each image. With such a coordinate system, an ROI can be extracted from the original image in order for reliable feature extraction and matching [12].

While acquiring the image, the finger is always put flatly on the capturing block, so the bottom boundary of the finger is considered to be stable and taken as the *X* axis of the coordinate system. Determining *Y* axis is quite difficult, so the following steps are followed:

Step1: Down sample the image

Firstly, the image, *I* is down sampled using Gaussian smoothing to suppress the noise in the original image

Step2: Extract *X* axis of the image:

The bottom boundary of the finger, which is the *X* axis of the ROI coordinate system, can be extracted using Canny edge detector. The image is then cropped empirically for further processing.

Step3: Edge Map:

The Canny edge detector is applied to get the corresponding edge map of the image, I_E. At this step, each pixel on the image, I_E is given a code to represent local convex direction. Two curves are extracted, namely leftward convex or rightward convex. A value 1 is given to pixels that lie on leftward convex curve and -1 to the pixels that lie on the rightward convex curve. All other pixels are given 0 values in the image.

Step4: Extracting *Y* axis of the image:

A column is selected (one of the points from the *X* axis) and a window of size (d × h) is selected. *h* is the height of the image I_s and d is empirically decided. In this window a convexity magnitude is computed. The center of the phalangeal joint is represented by the minimum of the convexity magnitude and that will be chosen as the midpoint or represent *Y* axis. The final image is then cropped using the *X* and *Y* axis.

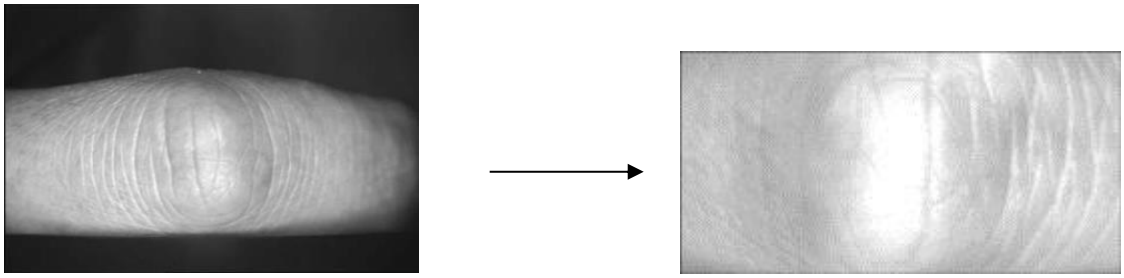

Figure 2.3: ROI extraction of Finger Knuckle Print (FKP).

- **Image Enhancement:** **T**he cropped region obtained from the captured image may contain some noisy data that may hinder the authentication process or may reduce the authentication rate. This noisy data hence must be removed using image enhancement techniques. There many image enhancement techniques like filtering, enhancements based on wavelet transforms, binarization, thinning etc.

  One of the most widely used techniques is enhancement using Gabor filter. Applying Gabor filter improves the textural features quality and thus can be easily extracted [1]. Gabor filter is convolved with the ROI image to get a high textured image.

$$G(x,y,\theta,v,\sigma)=\frac{1}{2\pi\sigma^2}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)\times\exp\left(2\pi i(vx\cos\theta+vy\sin\theta)\right) \qquad (2.2)$$

Here, *x* and *y* are the coordinates of the ROI image, *v* is the frequency of the sinusoidal wave, θ controls the orientation of the function, σ represents the standard

deviation of the Gaussian window. The Gabor convolved ROI has been termed as GROI.

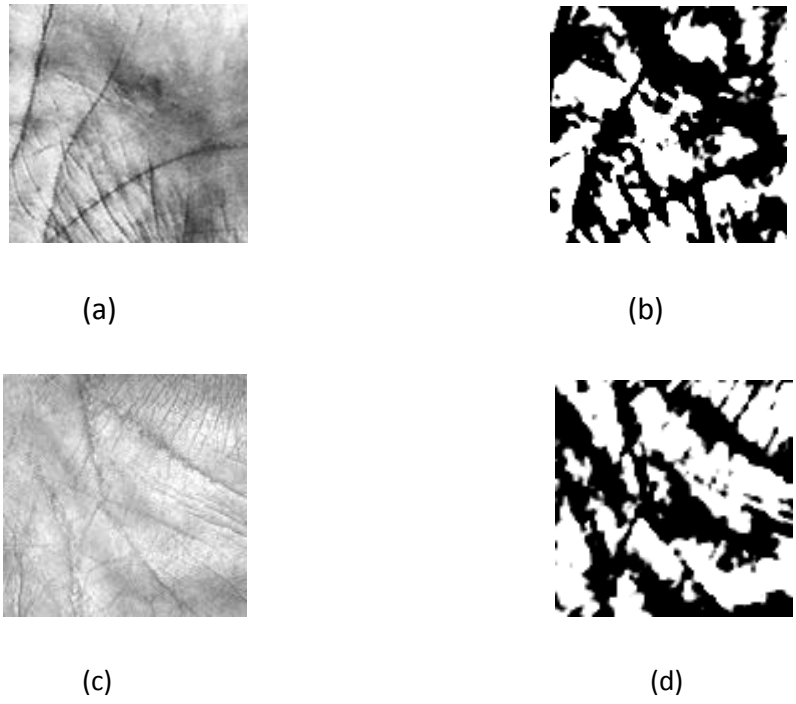

Figure 7 (a) ROI of IITD palmprint database, (b) GROI of IITD palmprint database,

(c) ROI of the PolyU palmprint database, (b) GROI of PolyU palmprint database

2.3 Feature Detection:

Feature detection is the third step in biometric identification system. Depending on the type of feature to be extracted, there are numerous algorithms available for feature extraction. Point features are more in number as compared to other features on the palmprint. So, point extracting techniques are required. In case the database used is a touch-less database then it is required that scale rotation invariant algorithms are used to extract the features from the ROI images. SIFT is one of the widely used algorithm to deal with scale and rotation invariances, but a combination of Harris Detector and SIFT descriptor has given more powerful results than SIFT detector.

Corners as features:

Interest points are those points of an image that can be easily extracted and have a well-defined position in the image. It can any isolated point of maximum or minimum intensity in the local neighbourhood or line endings .Corners are also considered as good interest points. Corners are used in shape analysis and motion analysis. Intersection of two edges can be termed as corner. A corner is also defined as point where there two different and dominant

edge orientations in a local neighbourhood of the point. Corners are good features that can be used in feature matching process. Hence, corners and features are interchangeably at times. There are many corner detection algorithms. Mathematically, corners are local image features characterized by locations where variations of intensity functions in both the directions (*X* and *Y*) are high. A corner differs from the edge in respect that in case of edge the partial derivative is high in a certain direction [26]. One of the very simple approaches is using correlation but it is a very expensive and suboptimal technique. A very efficient technique was proposed by Harris and Stephens, popularly known as Harris detector is explained below.

The Harris corner detector is a popular interest point detector that addresses the issues like invariance to rotation, scale, illuminations and noise present in the image. As per Harris, a corner is defined as a point where a significant change is observed in all directions when a small window is shifted around that point [42] as shown in figure.

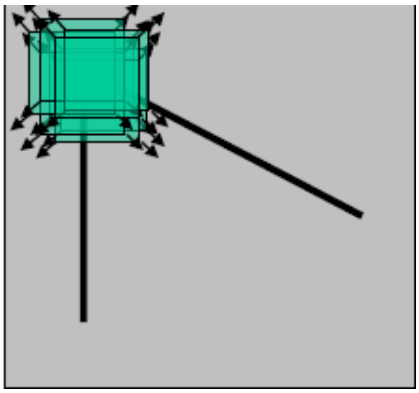

Figure 2.4: A corner point

Harris Corner extracts the various corner points from the image which are basically point features. It uses a scale at which the corner points are extracted. The appropriate scale is chosen from the range between the range of [6, 12]. The scale depends with image and the database used for extracting the corner points.

The Harris corner detector [6, 24] is based on the local auto-correlation function of a signal. Here the local auto-correlation function measures the local changes of the intensity with patches shifted by a small amount in different directions. Given a shift (Δx, Δy) and a point (x, y) of the ROI image *R*, the auto-correlation function is defined as,

$$c(x,y) = \sum_W \left[R(x_i,y_i) - R(x_i + \Delta x, y_i + \Delta y)\right]^2 \quad (2.3)$$

Where $R(x_i, y_i)$ denotes the image function and (x_i, y_i) is i^{th} point window (Gaussian) w centred at (x, y).The shifted image is approximated by a Taylor expansion truncated to the first order terms, where R_x and R_y denotes the partial derivatives in *x* and *y* respectively.

$$R(x_i + \Delta x, y_i + \Delta y) \approx R(x_i, y_i) + [R_x(x_i, y_i)\ R_y(x_i, y_i)] \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \quad (2.4)$$

After substituting, Equation 2.4 in Equation 2.3:

$$C(x, y) = \sum_W [R(x_i, y_i) - R(x_i + \Delta x, y_i + \Delta y)]^2 \quad (2.5)$$

$$= \sum_w \left(R(x_i, y_i) - R(x_i, y_i) - [R_x(x_i, y_i)\ R_y(x_i, y_i)] \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \right)^2 \quad (2.6)$$

$$= \sum_w \left(- [R_x(x_i, y_i)\ R(x_i, y_i)] \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \right)^2 \quad (2.7)$$

$$= \sum_w \left([R_x(x_i, y_i)\ R_y(x_i, y_i)] \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \right)^2 \quad (2.8)$$

$$= [\Delta x\ \Delta y] \begin{bmatrix} \sum_w (R_x(x_i, y_i))\ 2 & \sum_w R_x(x_i, y_i)\ R_y(x_i, y_i) \\ \sum_w R_x(x_i, y_i)\ R_y(x_i, y_i) & \sum_w (R_x(x_i, y_i))\ 2 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \quad (2.9)$$

$$= [\Delta x\ \Delta y]\ C(x, y) \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \quad (2.10)$$

C(x, y) captures the intensity structure in the neighbourhood of pixel *(x, y)*. Let λ_1 and λ_2 are the Eigen Values of matrix *C(x, y)*. There are 3 cases that should be considered for finding out the corner. A threshold is set for determining these cases.

1. If both the Eigen values are small, then the autocorrelation is flat, i.e. the window is of approximately same intensity.
2. If any one of the Eigen Value is high and other is low, so the local auto-correlation function is ridge shaped and the change in intensity is along only one direction implying that it is an edge.
3. If both the Eigen Values are high, the auto-correlation function is sharply peaked and hence the shifts in direction will result in high change in intensity levels indicating a corner point.

Once the corners are detected then Sift descriptor is applied, as described by David Lowe to get feature vectors [5]. Sift Descriptor is used to uniquely describe each corner (point) extracted so that the image can be correctly matched in the large database.

SIFT Descriptor:

Every corner point located using Harris Corner point has its own orientation. A window patch is selected around that corner point and sample image gradients and orientation are calculated. The gradient magnitude and orientation are calculated at a relevant scale at which the corner points are extracted. The orientations are bin into one of 36 bins each covering 10 degrees each pixel in the neighbourhood contributes to the bin corresponding to its orientation. The highest peak of the histogram is the dominant orientation of that corner. A 16×16 patch is considered centred on the point. The patch is decomposed into 4x4 pixel tiles so there will be 16 such tiles. For each such tile, a histogram of its pixels' gradient orientations with 8 bins, each covering 45 degrees is constructed. Again weight the contribution to the bin by the gradient magnitude. A circular Gaussian falloff from the point centre is taken which is half the descriptor window (8 pixels) as shown in figure. So this gives 128-dimensional vector which represents the feature vector. For each *k* corner points a descriptor is defined. Hence for one image the final feature vector obtained is *k*×128, a 2D vector.

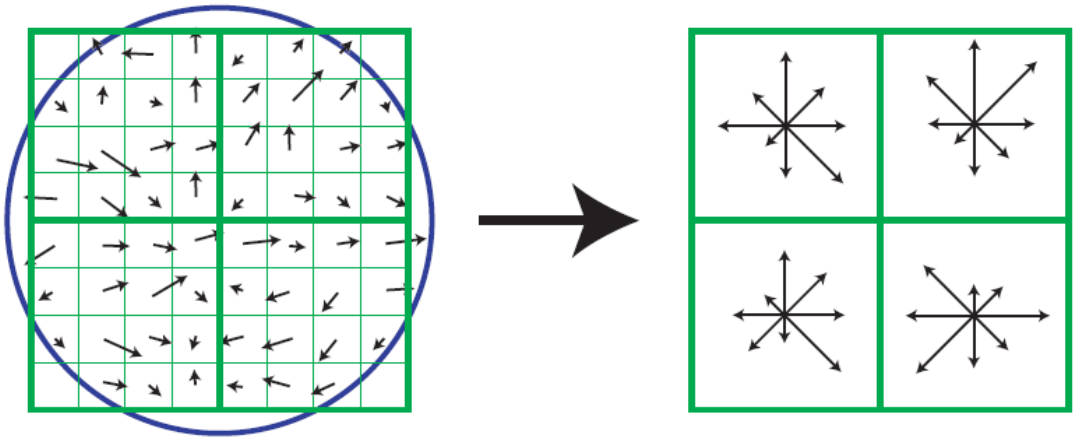

Figure 2.5: Image Gradients and Corner point descriptor obtained [5]

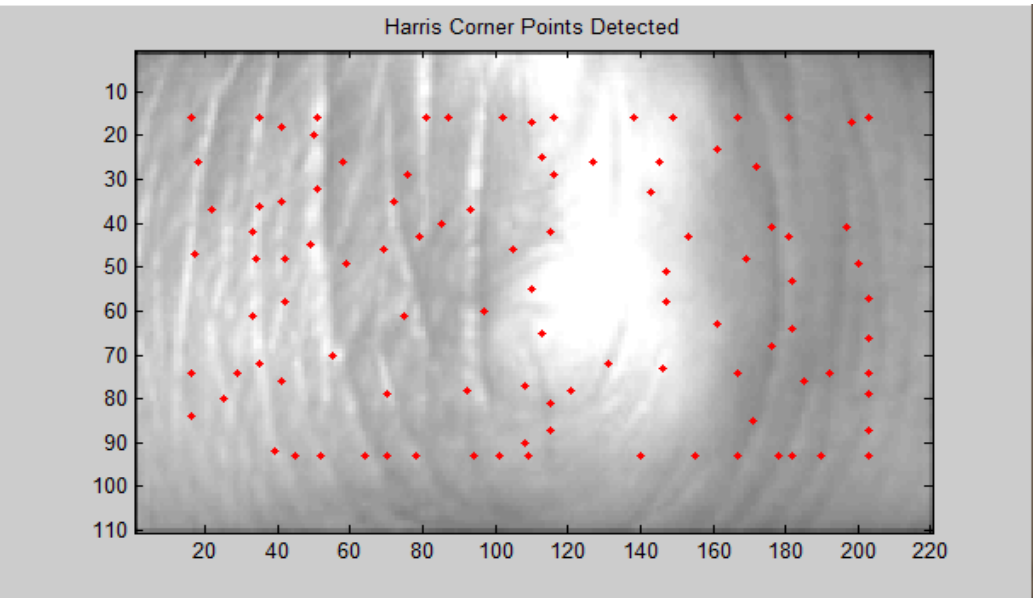

Figure 2.6: Harris Corner Points detected in a Finger Knuckle Print image

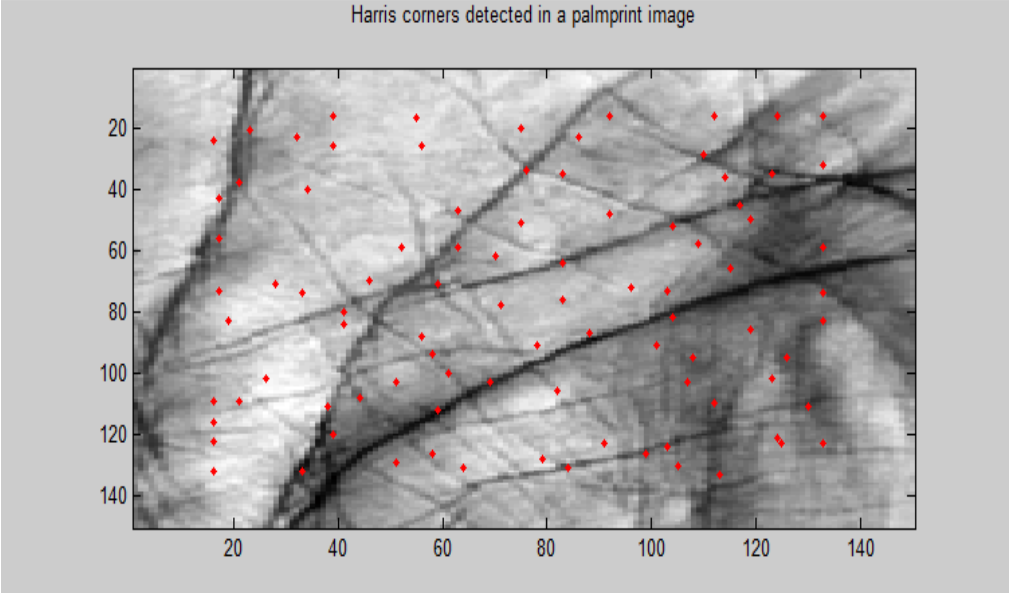

Figure 2.7: Harris Corners detected in a palmprint ROI

2.4 Matching:

After the features have been detected, these features are then used to verify the authenticity of the user. Algorithms that are widely used for matching the feature vectors are Euclidean Distance, Cosine Similarity, Hamming distance etc. These scores help in identifying /verifying the user.

The Euclidean distance between any two points, *a* and *b* is the length of the line segment connecting them. In Cartesian coordinates, if $a = (a_1, a_2 \ldots a_n)$ and $b = (b_1, b_2 \ldots b_n)$ are two points in Euclidean *n*-space, then the distance from *a* to *b*, or from *b* to *a* is given by:

$$d(a,b) = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + ...(a_n - b_n)^2} = \sqrt{\sum_{i=1}^{n} (a_i - b_i)^2} \tag{2.11}$$

The higher the distance value, *d* the less the two vectors match with each other. This means that the score value must be higher in case of an imposter and less in case of genuine.

Cosine similarity gives a measure of similarity between any two vectors. It computes the angle between the two vectors to give the similarity.

The cosine of two vectors is given by dot product formula

$$a.b = \|a\|b\|\cos\theta \tag{2.12}$$

Here, *a* and *b* are two feature vectors and θ represents the cosine similarity. The magnitude of similarity can be represented as follows:

$$similarity = \cos(\theta) = \frac{a.b}{\|a\|\|b\|} = \frac{\sum_{i=1}^{n} a_i \times b_i}{\sqrt{\sum_{i=1}^{n} (a_i)^2} \times \sqrt{\sum_{i=1}^{n} (b_i)^2}} \tag{2.13}$$

The similarity values ranges from −1 to 1. -1 represents that the feature vectors are exactly opposite and hence an imposter. 1 represents that feature vectors match exactly and hence it must be feature vectors of a genuine user. 0 usually indicates intermediate similarity and dissimilarity.

Chapter 3: Hidden Markov Model

In unsupervised learning, all the observed data is assumed to be originated by some hidden variables. Models that are developed using supervised learning often leave the probability for inputs undefined. An unsupervised model is not needed as long as the inputs are available, but if some of the input values are unknown, it is not possible to say anything about the outputs. If the inputs are also fabricated, then missing inputs cause no problem since they can be considered latent variables as in unsupervised learning [8]. Likewise, in Hidden Markov Model (HMM) the observation vector that is the sequence of either the feature vectors obtained after applying feature extraction algorithms or the scores obtained after matching process, are assumed to belong to some specific state. These states are hidden and thus are known as latent variables in HMM. So unsupervised learning method called HMM is used to model the observation vector and calculate the respective probabilities.

3.1 The Basic probability theory:

The mathematics behind the Hidden Markov Model method is pretty straightforward and easy to understand. In this section, certain basic probability theories have been described that are required to understand the Hidden Markov Model technique. The information is obtainable from many sources [44].

1. **Probability axioms:** Given a finite sample space *S* and an event *A* in *S*. We define $P(A)$ is the probability of *A*, then
   a. $0 \le P(A) \le 1$ for each event *A* in *S*.
   b. $P(S) = 1$.
   c. $P(A+B) = P(A) + P(B)$ if A and B are mutually exclusive events in *S*.

2. **Joint probability** : If A and B are random variables, then the joint probability function is $P(a,b) = P(A = a, B = b)$
3. **Conditional probability** : the probability of A conditioned on B is defined as
   $$P(A|B) = \frac{P(A,B)}{P(B)}$$
4. **Product rule:** from the definition of conditional probability, the product rule is
   $$P(A,B) = P(A|B)P(B) = P(B|A)P(A)$$

5. **Chain rule** : the chain rule is an extension of the product rule which we can write down in more generic form as:

   $$P(a_1, a_2, \ldots, a_n) = P(a_1 | a_2, \ldots, a_n) P(a_2 | a_3, \ldots, a_n) \ldots\ldots P(a_{n-1} | a_n) P(a_n)$$

6. **Bayes' rule**: Bayes' rule is an alternative method to calculate the conditional probability if the joint probability $P(A, B)$ is unknown. From the conditional probability, we get

   $P(A|B)P(B) = P(A, B)$ as well as $P(B|A)P(A) = P(A, B)$.

   Bayes rule is $P(A|B) = \dfrac{P(B|A)P(A)}{P(B)}$

3.2 EM (Expectation-Maximization Algorithm)

The EM algorithm is an algorithm which estimates the maximum likelihoods of the parameters of the given Hidden Markov Model in cases where some sort of information (states) is hidden, latent or missing. It is a statistical approach for maximizing the likelihood function in successive iterative steps. The parameters of the HMM are estimated by iterative EM algorithm and thus help in increasing the efficiency of the algorithm. In EM algorithm the estimation of parameters is based on some latent variables which are associated with each input data which requires initial set of some parameters which get finely tuned in every iteration to get the maximize the value of maximum likelihood.

The two applications of EM algorithm are:

1. The observation data has missing values. This may be due to some limitations or due to lack in collecting observation data.
2. The second application of EM algorithm occurs when likelihood function is analytically intractable and when the likelihood function can be simplified by assuming the existence of hidden parameters [43].

In case of HMM, the state distribution of observation sequence is hidden and hence affects the parameter values and maximum likelihood value of the observation sequence. So EM algorithm handles this problem. EM algorithm handles this by solving the linear equation simultaneously. Either the maximum likelihood value is known or we proceed to find the correct estimate of parameters using the latent variables and input data set or using an initial estimate of parameters the maximum likelihood function is found. It is also known as Baum-Welch Algorithm.

The algorithm comprises of two steps:

1. E-step: Likelihood of the given function is found.
2. M-step: The parameters that will maximize the likelihood are found.

In case of HMM, the parameters to be estimated using this algorithm are transition probabilities, prior probabilities and emission probabilities. An iterative approach is followed the parameters are estimated using M-step and then the likelihood function value is calculated. This number of iterations varies with model to model. The values are estimated till the convergence point is reached or the upper limit of number of iterations has reached. For first iteration, the parameter values are randomized. For further iterations, the previous iteration computed values of the parameters are considered. At times there may be cases when multiple maxima's are obtained, hence a local maxima is obtained and not the global maxima of the likelihood function.

3.3The Viterbi Algorithm:

The Viterbi algorithm is an algorithm which is used to find out the most probable sequence of the states for a given sequence of observations in Hidden Markov Model [45]. Most probable sequence of states can also be found using all possible sequence of states and thus finding probability for all the possible combinations of state sequence and the observation sequence. The combination which gives the maximum probability is considered as the most probable state sequence for the given observation sequence. Though this approach is viable but faces large number of computations if the number of states and the observation sequence is large. So, an algorithm is required which requires less computations and give probable state sequence. This Viterbi algorithm finds the most probable sequence using recursion process. In this first partial probability is defined which is the probability to reach a particular intermediate state. This is considered as the partial best state and path. For each intermediate state to the terminating state there is a best probable path to that state. At each intermediate state and end state, partial probability is maintained by using trellis. It is used to know the best partial path. Using this recursively helps in finding out the best probable state sequence for a given observation path. Hence this algorithm provides an efficient way by using recursion in computing the best probable sequence.

3.4 The Hidden Markov model

Leonard E. Baum and several colleagues was the first person to bring the theory of HMMs during the 1960s. In 1989, Lawrence *et al* published their tutorial on HMMs, which explains the theory of HMM in a more general form [4]. A Hidden Markov Model (HMM) is a probabilistic function of a Markov property. Many systems possess the property that given the present state, the past states of the system have no influence on the future states. This property of system is called the Markov Property, and the systems that exhibit these properties are called Markov chains. Based on the above mentioned Markov property, the conditional probability can be expressed as

$$P(q_t|q_1,\cdots\cdots,q_{t-1}) = P(q_t|q_{t-1}) \tag{3.1}$$

Where q_t is the random variable of the Markov system at time *t*.

The term "hidden" indicates that the system is a doubly stochastic process where the state $q_1^t = \{q_1,\cdots\cdots,q_t\}$ of the Markov chain is not directly observed (hence the term, "hidden"), but it is implicitly defined by a sequence $O_1^t = \{o_1,\cdots\cdots,o_t\}$ of the observed data that does not necessarily exhibit the Markov property.

A HMM is defined by the following conditional probabilities.

1. Given q_t , the distribution of y_t is independent of every other variable,

$$P\left(y_t\left|q_1^t, y_1^{t-1}\right.\right) = P\left(y_t\left|q_t\right.\right) \tag{3.2}$$

2. y_t is unable to affect q_t given the past.

$$P\left(q_{t+1}\left|q_1^t, y_1^t\right.\right) = P\left(q_{t+1}\left|q_t\right.\right) \tag{3.3}$$

HMM is characterized by the following five parameters:

1) The number of *N* hidden states within the model. Each state corresponds to a different exclusive state provided by the model.
2) The amount of *M* unique observations per state. These symbols are denoted as $V = \{v_1, v_2, \ldots, v_M\}$.

3) State transition probability distributions

   $A = \{a_{ij}\}$where$a_{ij} = P(q_{t+1} = S_j | q_t = S_i),\ 1 \le i, j \le N$

4) The emission probability distribution in state $j, B = \{b_j(k)\}$ where

$$b_j = P(v_k \ at\ t \,|q_t = S_j),\ 1 \le j \le N, 1 \le k \le M$$

   It defines the probability of having that observation in that particular state.

5) The prior probability $\pi_i = \{\pi_i\}$of being in state i at the beginning of the observations where$\pi_i = P(q_1 = S_i), 1 \le i \le N$.

The values of N, M, A, B, and π can be used to generate the observation sequence

There are three basic issues regarding HMMs that must be solved before an HMM can be used.

Problem 1:

Computing the probability of the observation sequence given a particular model

$\lambda = (A, B, \pi)$. This problem is known as evaluation problem. The solution of this problem measures how well a model matches the observation sequence. It gives the probability or log –likelihood which is an estimate of the belongingness of the observation sequence to the model.

Given a model, $\lambda = (A, B, \pi)$ how can $P(O|\lambda)$ be computed for each observation sequence?

Solution to Problem 1:

The probability of the observed sequence, written as $P(O_1^T)$, can be calculated by finding the joint probability of the observation sequence and the state sequence $P(O_1^T, q_1^T)$. The term $P(O_1^T, q_1^T)$ can be recursively factored using conditional probability and chain rules.

$$P(O_1^T, q_1^T) = P(O_T, q_T | O_1^{T-1}, q_1^{T-1})\, P(O_1^{T-1}, q_1^{T-1}) \quad (3.4)$$

$$= P(O_T | q_T, O_1^{T-1}, q_1^{T-1}) P(q_T | O_1^{T-1}, q_1^{T-1}) P(O_1^{T-1}, q_1^{T-1}) \quad (3.5)$$

$$= P(O_T | q_T)\, P(q_T | q_{T-1})\, P(O_1^{T-1}, q_1^{T-1}) \quad (3.6)$$

$$= P(q_1)\prod_{t=2}^{T} P(q_t|q_{t-1})\prod_{t=1}^{T} P(O_t|q_t) \quad (3.7)$$

Where

$P(q_1)$ represents the initial state probability of q at time 1

$P(q_t|q_{t-1})$ represents the probability of q at time t given q at time $t+1$

$P(O_t|q_t)$ represents the emission probability

We can get the desired probability by marginalizing (summing) over random variables

$$q_{1\ldots T}: \; P(O_1^T) = \sum_{q_{1\ldots T}} P(O_1^T, q_1^T) \quad (3.8)$$

Since a lot of computations are involved here, so a simpler method is there. A combination of forward and backward recursion algorithm reduces the complexity and number of computations.

The forward recursion:

The forward recursion calculates, $P(O_1^t, q_t)$, the probability of an observed partial sequence y_1^t for a given state q_t. We can rewrite this joint probability as a conditional probability in the product form: $P(O_1^t, q_t) = P(O_t|O_1^{t-1}, q_t)P(O_1^{t-1}, q_t)$. According to the HMM assumption, the term $P(O_t|O_1^{t-1}, q_t)$ can be reduced to $P(O_t|q_t)$. Thus, we need to calculate $P(O_1^t, q_1^{t-1})$ to complete the equation.

$$P(O_1^t, q_t, q_{t-1}) = P(q_t|q_{t-1}, O_1^{t-1})P(O_1^{t-1}, q_{t-1}) \quad (3.9)$$

$$= P(q_t|q_{t-1})P(O_1^{t-1}, q_{t-1}) \quad (3.10)$$

$$P(O_1^{t-1}, q_t) = \sum_{q_{t-1}} P(O_1^t, q_t, q_{t-1}) \quad (3.11)$$

$$= \sum_{q_{t-1}} P(q_t|q_{t-1})\, P(O_1^{t-1}, q_{t-1}) \quad (3.12)$$

Hence, we get the following equation.

$$P(O_t, q_t) = P(O_t \Big| q_t) \sum_{q_{t-1}} P(q_t \big| q_{t-1}) P(O_1^{t-1}, q_{t-1}) \qquad (3.13)$$

This equation exhibits a recurrence relation so that $P\left(O_1^t, q_t\right)$ can be calculated recursively. We define $\alpha_q(t) = P\left(O_1^t, q\ \right)$ then the above equation can be expressed as

$$\alpha_q(t) = P(O_t \Big| Q_t = q) \sum_r \left(Q_t = q | Q_{t-1} = r\right) \alpha_r\left(t-1\right) \qquad (3.14)$$

where Q_t is the state space at time *t*

The backward recursion

Once we complete the forward phase, we still need to calculate the backward phase in the algorithm. The backward phase calculates the partial probability from time *t+1* to the end of the sequence, given q_t.

$$P\left(O_{t+1}^T | q_t\right) = \sum_{q_{t+1}} P\left(q_{t+1}, O_{t+1}, O_{t+2}^T | q_t\right) \qquad (3.15)$$

$$= \sum_{q_{t+1}} P\left(O_{t+2}^T | q_{t+1}, O_{t+1}, q_t\right) P\left(O_{t+1} | q_{t+1}, q_t\right) P\left(q_{t+1}, q_t\right) \qquad (3.16)$$

$$= \sum_{q_{t+1}} P\left(O_{t+2}^T | q_{t+1}\right) P\left(O_{t+1} | q_{t+1}\right) P\left(q_{t+1}, q_t\right) \qquad (3.17)$$

In the backward phase, $P\left(O_{t+1}^T | q_t\right)$ can only be calculated once we have the information about $P\left(O_{t+2}^T | q_{t+1}\right)$. This is also called backward recursion.

We can define $\beta_q(t) = P\left(O_{t+1}^T | Q_t = q\right)$, and the above equation can be expressed as

$$\beta_q(t) = \sum_r \beta_r(t+1) P\left(O_{t+1} | Q_{t+1} = r\right) P\left(Q_{t+1} = r | Q_t = q\right) \qquad (3.18)$$

where Q_t is the state at time *t*

The forward-backward recursion gives us the essential information to calculate the probability of the observed sequence $P\left(O_1^T\right)$.

$$P\left(O_1^T\right) = \sum_{q_t} P(q_t, O_1^T, O_{t+1}^T) \quad (3.19)$$

$$= \sum_{q_t} P\left(O_{t+1}^T \middle| q_t, O_1^T\right) P\left(q_t, O_1^T\right) \quad (3.20)$$

$$= \sum_{q_t} P\left(O_{t+1}^T \middle| q_t\right) P\left(q_t, O_1^t\right) \quad (3.21)$$

$$= \sum_{q_t} \beta_{q_t}(t)\alpha_{q_t}(t) \quad (3.22)$$

Problem 2:

The second problem is the decoding problem. It states given the model and a sequence, what the optimal state sequence is. How can a state sequence $Q = q_1 q_2 \ldots . q_T$ be chosen for the observation sequence O and the model λ? We want to find the most likely state sequence q_1^T corresponding to a given observation sequence O_1^T. Since our interest is in the overall model performance rather than finding a specific sequence, we will use an approach to maximize the expected number of states for our HMMs.

Solution to Problem 2:

The state posterior probability, $P\left(q_t \middle| O_1^T\right)$, is the probability of being in a certain state at time t, given the observation sequence O_1^T. It can be expressed using the variables that we defined from previous forward and backward processes.

The probability, $P\left(q_t \middle| O_1^T\right)$, is simply the product of forward-backward variables and normalized by the joint distribution of the observation sequences. Hence we express it as the following equation:

$$P\left(q_t \middle| O_1^T\right) = \frac{P\left(q_t, O_1^T\right)}{P\left(O_1^T\right)} = \frac{P\left(O_1^T \middle| q_t\right) P(q_t) P\left(O_{t+1}^T \middle| q_t\right)}{P\left(O_1^T\right)} = \frac{P\left(O_1^t, q_t\right) P\left(O_{t+1}^T, \middle| q_t\right)}{P\left(O_1^T\right)} \quad (3.23)$$

Since $P\left(q_t \middle| O_1^T\right)$ is normalized by the joint distribution of the observation sequences. $\sum P\left(q_t \middle| O_1^T\right) = 1$. The most likely state is measured by maximizing $P\left(q_t \middle| O_1^T\right)$ for q_t.

Problem 3:

The learning problem: The learning problem is the most interesting of all three problems. If there is existing models, we can use this technique to find the model for a sequence and re-apply the sequence to the model for the learning and decoding problem. This is extremely useful. It is also the problem that our experiments are focused on. The technique is to apply the forward-backward algorithm to this problem and use the Baum-Welch Algorithm, also known as Expectation Maximization Algorithm to refine the model parameters.

How do we maximize $P(O|\lambda)$ by adjusting model parameters A, B, and π?

Solution to Problem 3:

The transition posterior probability, $P\left(q_t, q_{t-1} \middle| O_1^T\right)$, can be expressed as :

$$P\left(q_t, q_{t-1} \middle| O_1^T\right) = \frac{P\left(O_t \middle| q_t\right) P\left(O_1^{t-1}, q_{t-1}\right) P\left(O_{t+1}^T \middle| q_t\right) P\left(q_t \middle| q_{t-1}\right)}{P\left(O_1^T\right)} \tag{3.24}$$

Note that if we marginalize the probability of $P\left(q_t, q_{t-1} \middle| O_1^T\right)$ over all possible state q_{t-1}, the result is the probability $P\left(q_t \middle| O_1^T\right)$. Once we obtain the state posterior, we can calculate the expected number of time that a certain state q_t is visited by simply summing over the time index t: $\sum_{t=1}^{T-1} P(q_t \middle| O_1^T)$.

To calculate the expected number of time that a transition from state i transits to state j, we sum all the $P(q_t, q_{t-1} \middle| O_1^T)$ overtime index t, $\sum_{t=1}^{T-1} P(q_t = i, q_{t-1} = j \middle| O_1^T)$.

Hence, we can re-estimate the HMM parameter using the formulas that we describe as follows.

- The re-estimated initial state probabilities are simply the expected frequencies of the states at time $t = 1$.

- The re-estimated transition probabilities are the expected numbers of transition from q_t to q_{t-1} over the expected number of transitions

$$\frac{\sum_{t=1}^{T-1} P(q_t = i, q_{t-1} = j \mid O_1^T)}{\sum_{t=1}^{T-1} P(q_t \mid O_1^T)} \tag{3.25}$$

- The re-estimated emission probabilities are the expected number of time that a certain state i is visited for the specific observation symbols over the expected number of time in a particular state.

$$\frac{\sum_{t=1}^{T-1} {}_{O=o} P\left(q_t \mid o_1^T\right)}{\sum_{t=1}^{T-1} P\left(q_t \mid o_1^T\right)} \tag{3.26}$$

The observations can be discrete or continuous. If the observations are continuous as in case of the feature vectors or score vectors, the emission probability is computed using Gaussian Mixture Model. The GMM gives the probability density function which is given by formula.

$$B(O, q\ ) = \frac{1}{\sqrt{2\pi}\sigma} \exp - \left(\frac{(O-\mu)^2}{2\sigma^2} \right) \tag{3.27}$$

Here, *O* represents the particular observation in observation vector.

q is the particular state which is under consideration.

μ is the mean of the observations under the state *q*.

σ is the standard deviation of the observations that are assumed to be under state *q*.

In such cases, the Hidden Markov Model is known as Continuous Density Hidden Markov Model (CDHMM). In CDHMM, the each state is assumed to be made up of few mixing components. Selecting the number of components in each state was very difficult but the

fitting algorithm proposed in [7] can be used to automatically estimate the number of components per state.

3.5 Types of Hidden Markov Model

Generally there are two types of HMM:

1. **Ergodic Model:** In case all the transition probabilities are strictly positive then the state diagram is fully connected. This fully connected model is known as Ergodic model. There are no zero values in the transition probabilities.

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

2. **Left to Right Model:** Left to Right models are also known as Bakis Models. In this the transition matrix is upper triangular matrix. Here the lower triangular values are zero. The state transition can always occur in a higher state or to the same state.

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22} & a_{23} \\ 0 & 0 & a_{33} \end{pmatrix}$$

Chapter 4: Proposed Approach

The following proposed approaches aims at extracting Harris corners from the palmprint and finger knuckle print images. The finger knuckle print database comprises of left middle, right middle, left index and right index. The feature vectors obtained after extracting features are used by HMM, which involves unsupervised learning from the set of feature vectors without knowing to which state a particular feature vector belongs. Thought deciding the number of states in HMM is crucial for the high authentication rates, so the proposed iterative approach aims at finding the correct number of states. The observation data assumed here can be a feature vector or a matching score.

The following things have been proposed in the work:

1. Authentication using HMM at feature level.

2. Authentication by score fusion using HMM

The proposed approach aims at extracting point features from the palmprint and knuckle print images. The features (corners) are extracted using Harris Corner Detector. These features are then described using SIFT descriptor to form a feature vector. This feature vector acts as data in the classifier Hidden Markov Model for authentication at feature level. These vectors are also matched against feature vectors of other users to get the matching scores using Euclidean Distance or Cosine Similarity. The scores obtained are also used as data in HMM for authentication at score level.

The database of the images used in this approach is PolyU database and IITD database. The PolyU palmprint database is widely used contact-based constrained palmprint database. The experiments are performed with 2415 samples collected from 345 different palms. Seven samples of each user were used and the ROI of size 150×150 was taken for experimental purposes. This database employs user-pegs to restrict the hand-pose and image scale variations. The use of pegs in imaging devices is highly inconvenient to the users but helps in restricting the palmprint texture image variations. This database is constrained and contact-based database [46].

The IIT Delhi palmprint image database consists of the hand images of 235 people. The images were captured using a simple and touch-less setup. All the images are collected in the indoor environment and circular fluorescent illumination around the camera lens is employed.

The images are captures by posing the hand at different angles to give scale rotation variations in the images [48].

The Finger Knuckle Print database of PolyU is also a publically available database. FKP images were collected from 165 people. While collecting images, the subject was asked to provide 6 images for each of the left index finger, the right index finger, the left middle finger and the right middle finger. In total, 48 images from 4 fingers of each subject were collected. In total, the database contains 7,920 images from 660 different fingers [47].

4.1 Authentication at Feature Level:

Authentication using feature vector, obtained by a combination of Harris detector and SIFT descriptor, aims at classifying users on the basis of log likelihood value obtained. At feature level, every subject represent a model obtained using HMM. For every one subject, there is one representing model that makes the subject different from others. Every subject is represented by a model in the training phase. In training phase, all the training images and their feature vectors, particular to one subject, are used as data for the HMM. Using this data, a model is created for each subject. Once models for every subject have been obtained, now testing can be done using the testing images. Using every testing image, a feature vector is computed (data), which is then used to calculate the probability. Every model, λ comprises of few parameters like transition probability (A), emission probability (B), prior probabilities (π). Using these parameters and feature vector (data) obtained through feature extraction method for authentication, a probability (log likelihood) is calculated for all the models using HMM. The greater the likelihood for the specific subject, more it belongs to that subject. For example, in case of palmprint authentication using PolyU palmprint database, we have 346 subjects, so number of models will be 346. (λ_1 λ_2.....λ_{346}) . Corresponding to each model a probability is calculated ($P(O|\lambda)$).

Deciding the number of states (N) for each model is a big task. So, an iterative approach is followed in which the value of state is assumed to be n (empirically chosen value greater than 2) initially and increased with an increment of 2 at each iteration. This process is done until GAR (at 0.01 FAR) tends to converge approximately. For HMM model parameters (A, B, π) , a initial random guess is made and is then optimized according to the data using EM algorithm. The whole process can be summarized as follows:

1. For i = 1, N = *n*

2. Initialize N = N+2

3. Training phase: For every subject do the following:

- Take training images and apply Harris corner and SIFT descriptor to obtain data.
- Initialize HMM parameters randomly.
- Derive the HMM model.

4. Testing Phase: For every subject do the following:

- Take testing images and apply Harris corner and SIFT descriptor to obtain data.
- Calculate probability against every model derived above.
- The model with highest probability is taken as the best match for the test image.

5. Plot the ROC(Receiver Operating Characteristic).

- If (GARi-1 < GARi), then goto to step 2 for another iteration .
- If ($GAR_{i-1} \approx GAR_i$), then terminate.

4.2 Authentication at score level using fusion methods:

The authentication rate of any biometric system can be increased if multiple traits are used. The fusion is done here at score level, in which the scores from different traits are used and fused together using various fusion rules to analyse the variations in the authentication rates (comparing GAR). The data which is fed into the Hidden Markov Model for training and testing purposes is fused data , which is obtained by applying rules like sum rule, min rule, max rule etc. The features obtained using the feature detection methods are matched using cosine similarity or Euclidean distance. The scores are stored as genuine and imposter score vectors. These vectors are now used as data in Hidden Markov Model for authentication. Same as feature level, here also there are two phases. Training and Testing phase. In training phase, the two models are created $\lambda_{genuine}$ and $\lambda_{imposter}$. $\lambda_{genuine}$ is created using the genuine score vector and $\lambda_{imposter}$ is created using imposter genuine score vector. The classification includes both training and testing phases, data must be provided to each phase sufficiently. In training phase, a subset of scores is chosen from both genuine vector and imposter score vector for the same number of users. These subsets are then used to generate two models. In testing phase, the observation data is the score obtained by matching with the training images

stored in the database. For every subject, there is 1 genuine score and other scores are imposter scores. The probability is calculated using these scores against the two models developed $P(O|\ \lambda_{genuine}\ )$ and $P(O|\lambda_{imposter})$. An unlabeled biometric sequence will be classified as a genuine user if $P(O|\ \lambda_{genuine}\ ) > P(O|\lambda_{imposter})$. A count(count_FRR) is maintained for all those scores which belong to genuine class and are categorised into imposter class using the above said method to calculate False Rejected Ratio. Similarly, a count (count_FAR) is also maintained for the falsely accepted scores to calculate False Accept Ratio (FAR).

The steps can be listed as follows:

1. For i = 1, N = *n*

2. Initialize N = N+2

3. Training phase:

- Take training subsets of genuine and imposter scores from each trait.
- Apply the fusion rule (sum rule, min rule or max rule)
- Initialize HMM parameters randomly.
- *Log-likelihood* is computed.
- Next iteration, again *log-likelihood* is computed, if it decreases, increase the state.
- Derive the HMM models $\lambda_{genuine}$ and $\lambda_{imposter}$. using the fused data

4. Testing Phase: Take the remaining subset of scores from the genuine and imposter vectors and do the following:

- Initialize count_FAR = 0 and count_FRR = 0
- Take Genuine score data from all the traits and form the fused data
  - ✓ Calculate probability against both the models derived above.
  - ✓ The model with highest probability is taken as the best match for that score.
  - ✓ If $P(O|\ \lambda_{genuine}\ ) > P(O|\lambda_{imposter})$ it will be added to genuine class which is true. But if $P(O|\ \lambda_{genuine}\ ) < P(O|\lambda_{imposter})$ then it is false rejection and so increment count_FRR by 1.
- Take Imposter score data from all the traits and fuse them together to generate the fused data
  - ✓ Calculate probability against both the models derived above.

- ✓ The model with highest probability is taken as the best match for that score.
- ✓ If $P(O|\lambda_{genuine}) < P(O|\lambda_{imposter})$ it will be added to Imposter class which is true. But if $P(O|\lambda_{genuine}) > P(O|\lambda_{imposter})$ then it is false acceptance and so increment count_FAR by 1

2. Using the values of count_FAR , count_FRR and the total number of scores, GAR can be calculated.

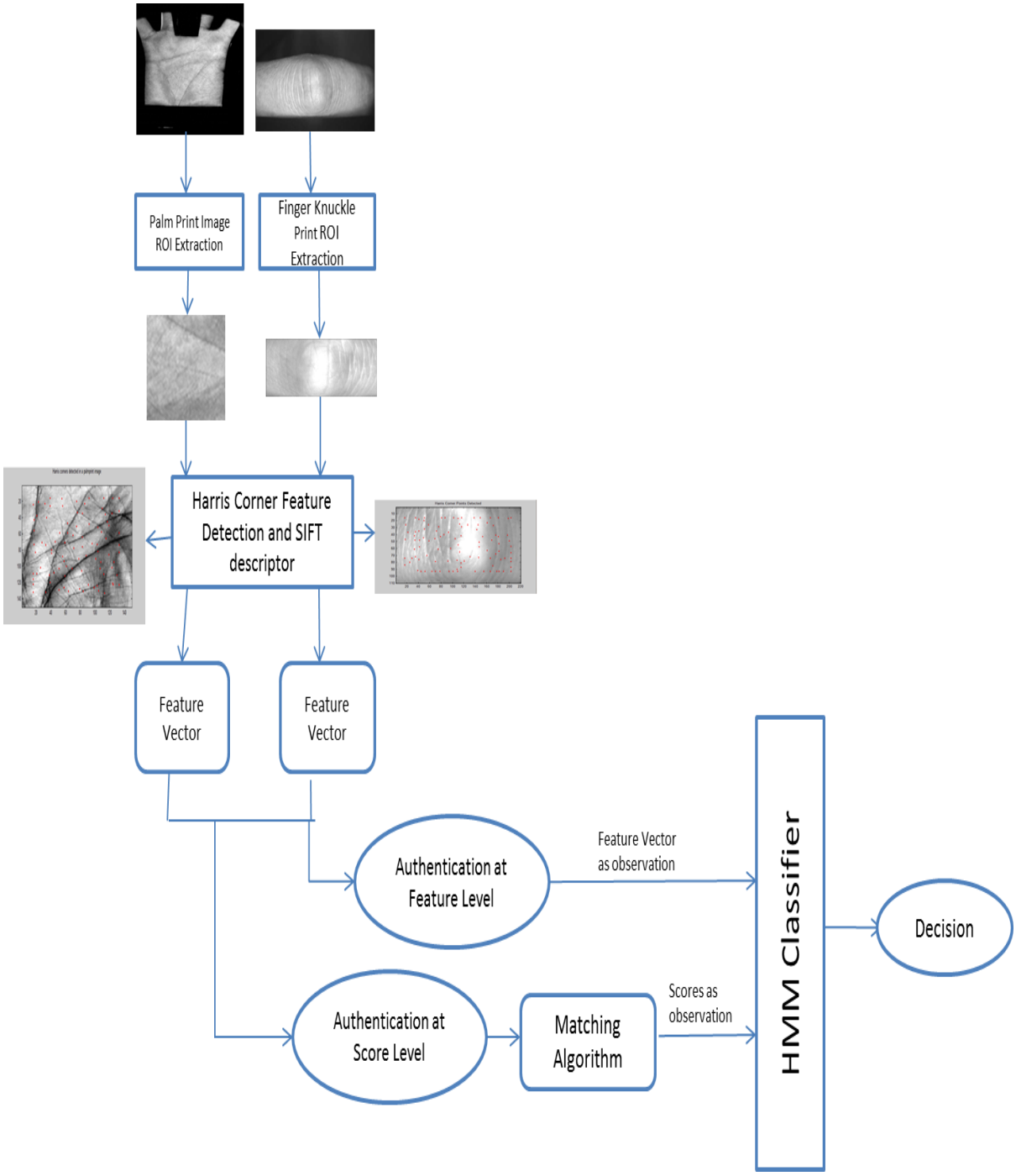

Figure 4.1: The flowchart for the proposed approach.

Chapter 5: Experimental Results

The proposed approach is used and the experimental results on the various databases are shown below. The ROC plots at feature level depict the highest authentication rate and the correct number of state for that particular database.

5.1 Results for authentication at feature level:

5.1.1 Experiment1: PolyU Palmprint Database

In the first experiment, PolyU palmprint database was used for the authentication process at feature level. The value of the states was empirically decided initially to 4. As the number of states increased an increase in GAR is also observed which converged at state equal to 14. The GAR at this state was 92.5%

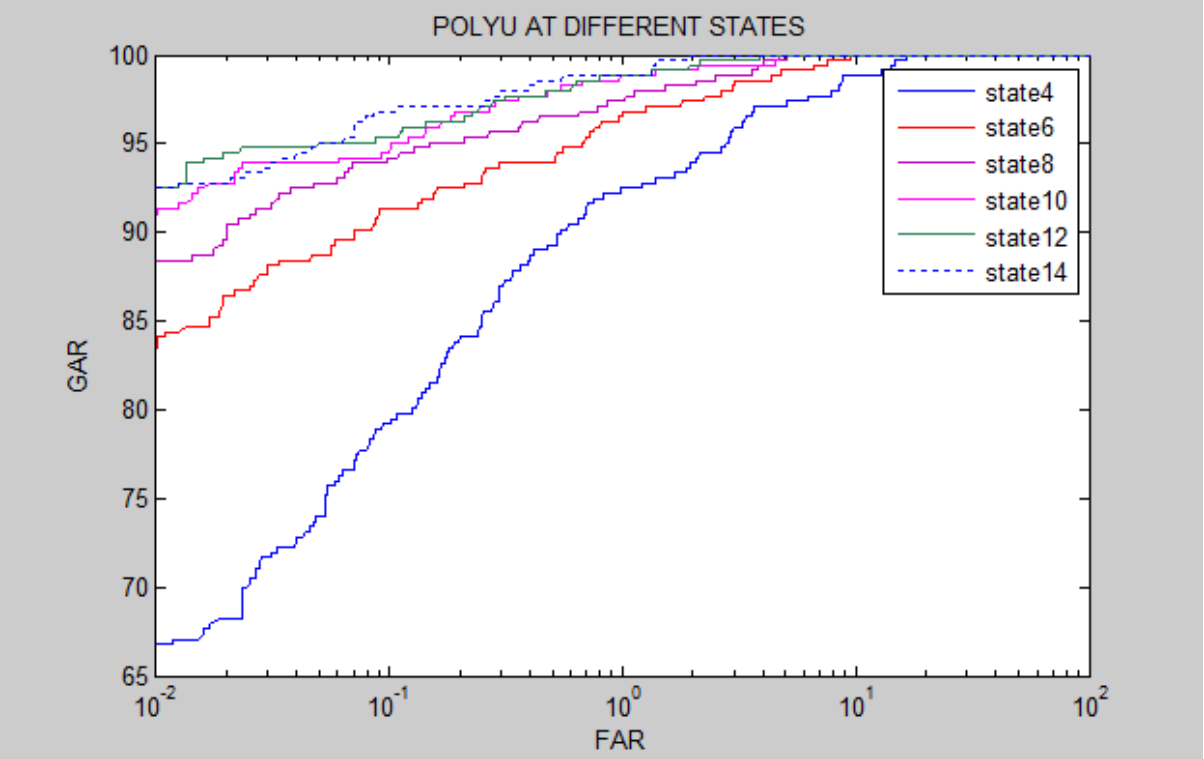

Figure 5.1: ROC for PolyU palmprint database at different states using HMM at feature level

Number of States	GAR at 0.01FAR
4	67
6	84
8	88
10	91
12	92
14	92.5

Table 1: GAR of PolyU palmprint database at different states using HMM at feature level

5.1.2 Experiment2: PolyU Gabor Convolved Palmprint Database:

In this experiment, GROI (Gabor convolved ROI) images are used for authentication with HMM classifier. It was surprising to note that the ROC curve declined for the GROI images. It was reported 65% GAR at 0.01 FAR against 92.5% for the ROI images. Here the number of states was assumed to be same as that in PolyU database for which highest GAR was reported.

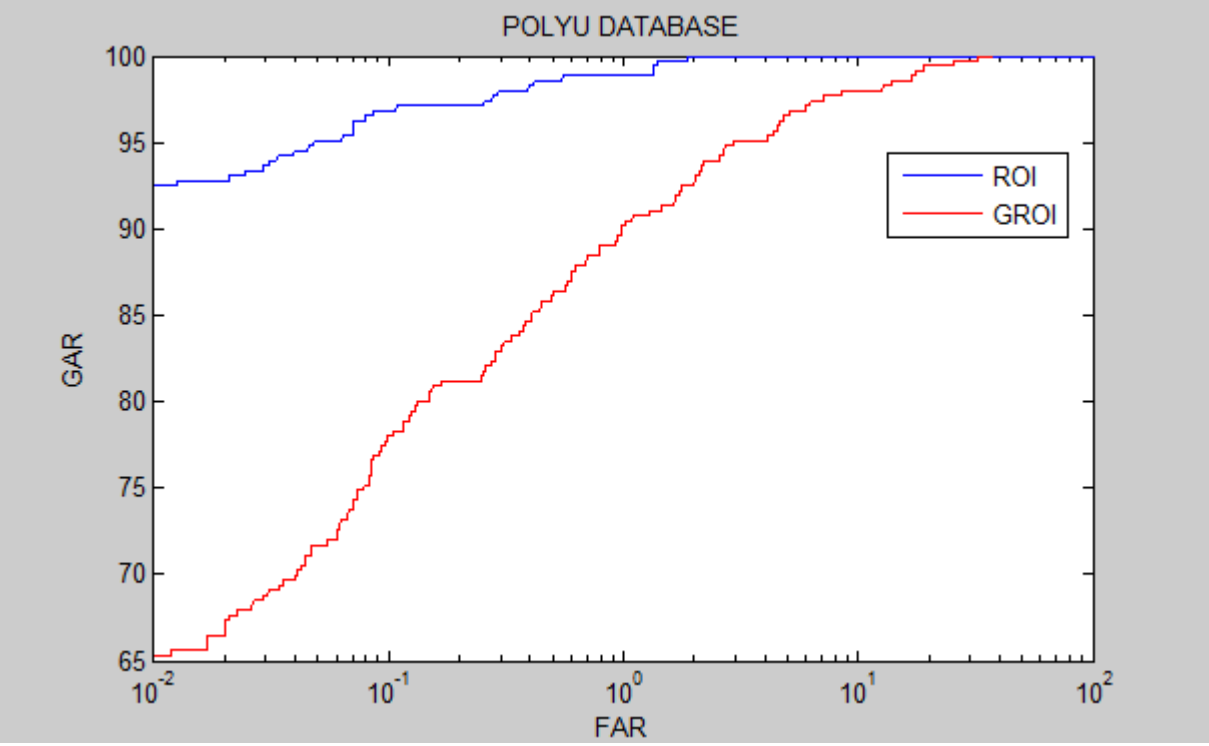

Figure 5.2: ROC for PolyU Palmprint database and Gabor Convolved PolyU Palmprint Database using HMM at feature level

5.1.3 Experiment 3: IITD Palmprint Database:

In this experiment IITD palmprint database was used and HMM was used for authentication process at feature level and number of states was estimated. Initially, the number of states was set at 6. The final GAR obtained was 73% at state 12.

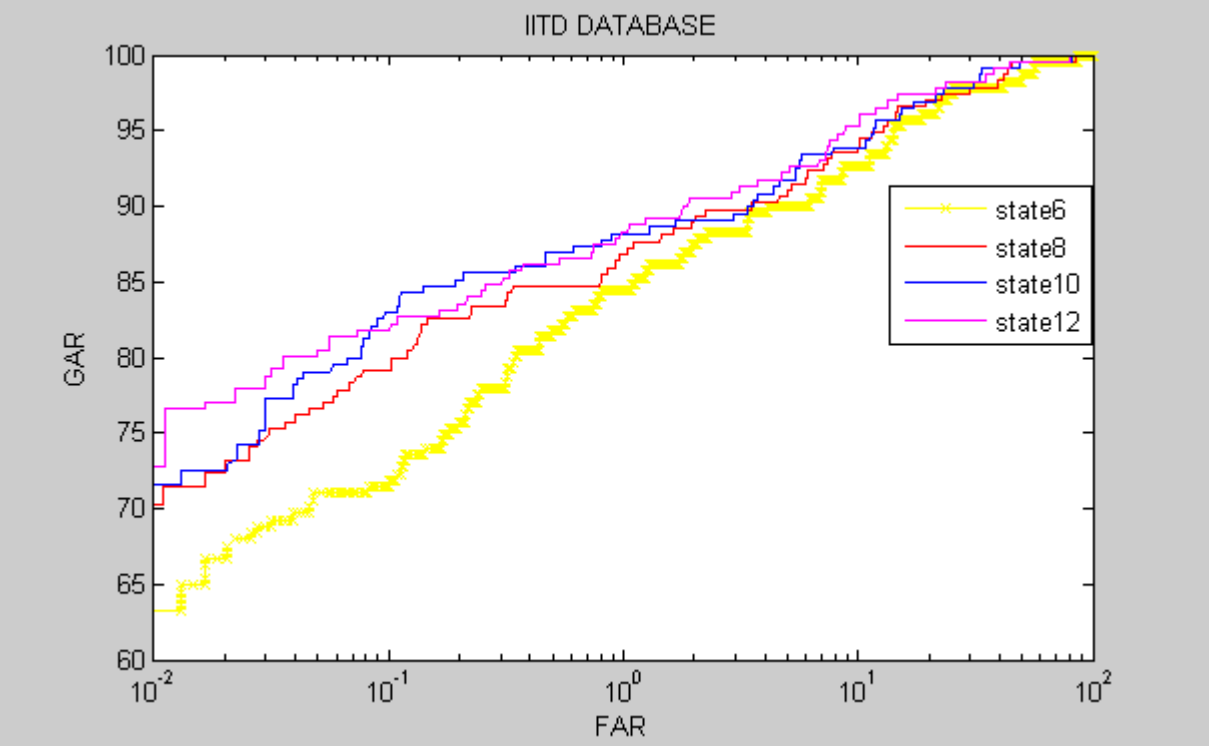

Figure 5.3: ROC for IITD Palmprint database at different states using HMM at feature level

Number of States	GAR at 0.01FAR
6	70
8	71
10	72.5
12	73

Table 2: GAR of IITD Palmprint database at different states using HMM at feature level

5.1.4 Experiment 4: PolyU Knuckle Print of Right Middle Finger:

In the fourth experiment, PolyU Knuckle print database was used for the authentication process at feature level. The feature vector was extracted using Harris Corner and SIFT. This feature vector is now used as the observation data. The value of the states of the models was empirically decided initially to 6. The number of states was incremented with the interval of 2. The GAR converged at state 16. The highest GAR reported at this state was 97%.

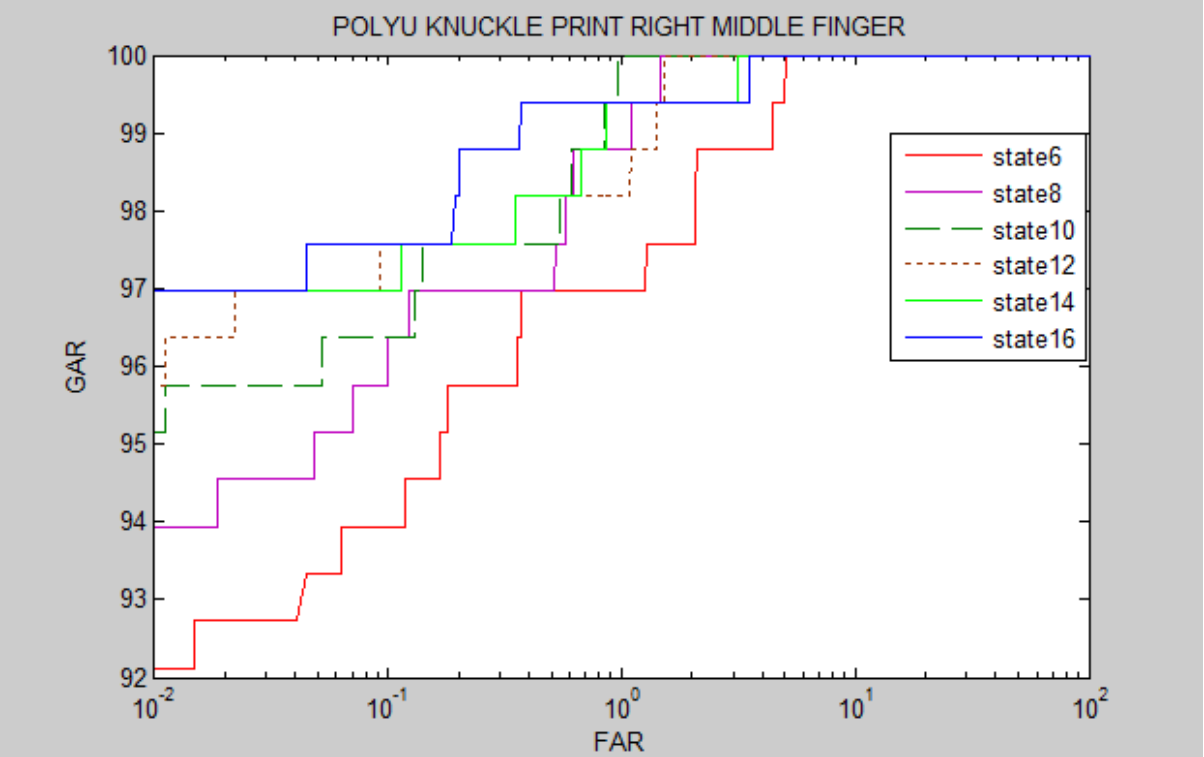

Figure 5.4: ROC for PolyU left index finger knuckle print database at different states using HMM at feature level

Number of States	GAR at 0.01FAR
6	92
8	94
10	95
12	96
14	97
16	97

Table 3: GAR of PolyU right middle finger knuckle print database at different states using HMM at feature level

5.1.5: Experiment 5: PolyU Knuckle Print of Left Index Finger

In this experiment, PolyU Left Index Finger Knuckle Print images were used. The features were extracted and described using SIFT descriptor. These feature vectors then resulted in calculation of log-likelihood genuine and imposter vectors which were used to plot the ROCs. Most surprising was that to see a great increase in GAR from state 6 to state 8. Here, GAR increased from 9 to 76 almost an increase of 8 times. Finally, the GAR converged at states 14 with value 89%.

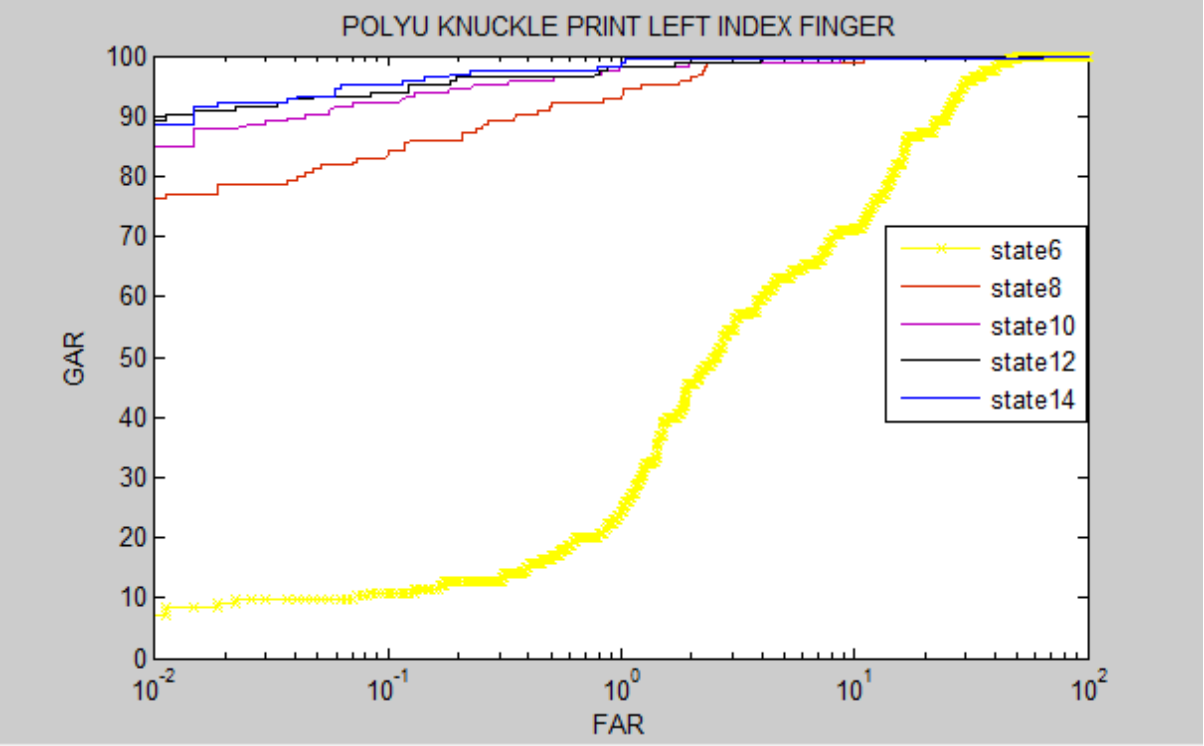

Figure 5.5: ROC for PolyU left index finger knuckle print database at different states using HMM at feature level

Number of States	GAR at 0.01FAR
6	10
8	76
10	85
12	88.5
14	89

Table 4: GAR of PolyU left index finger knuckle print database at different states using HMM at feature level

5.1.6 PolyU Finger Knuckle print of Right Index Finger`

In this experiment, PolyU Right Index Finger Knuckle Print images were used. The feature vectors resulted in calculation of log-likelihood genuine and imposter vectors which were used to plot the ROCs. A great increase in GAR from state 6 to state 8 was observed. Finally, the GAR converged at states 14 with value 92.5%

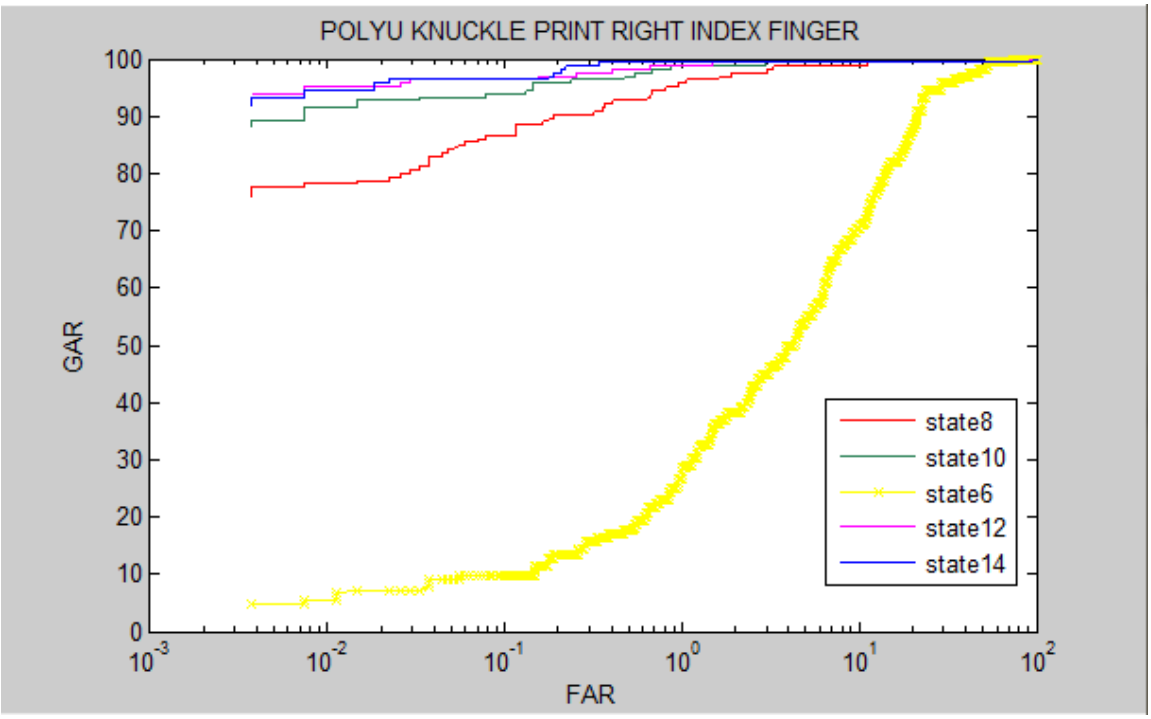

Figure 5.6: ROC for PolyU Right Index Finger knuckle print database at different states using HMM at feature level

Number of States	GAR at 0.01FAR
6	5
8	76
10	89
12	92
14	92.5

Table 5: GAR of PolyU right index finger knuckle print database at different states using HMM at feature level

5.1.7 PolyU Finger Knuckle print of Left Middle Finger`

In this experiment, PolyU Left Middle Finger Knuckle Print images were used. The features were extracted and described using SIFT descriptor. Finally, the GAR converged at states 14 with value 95.5%.

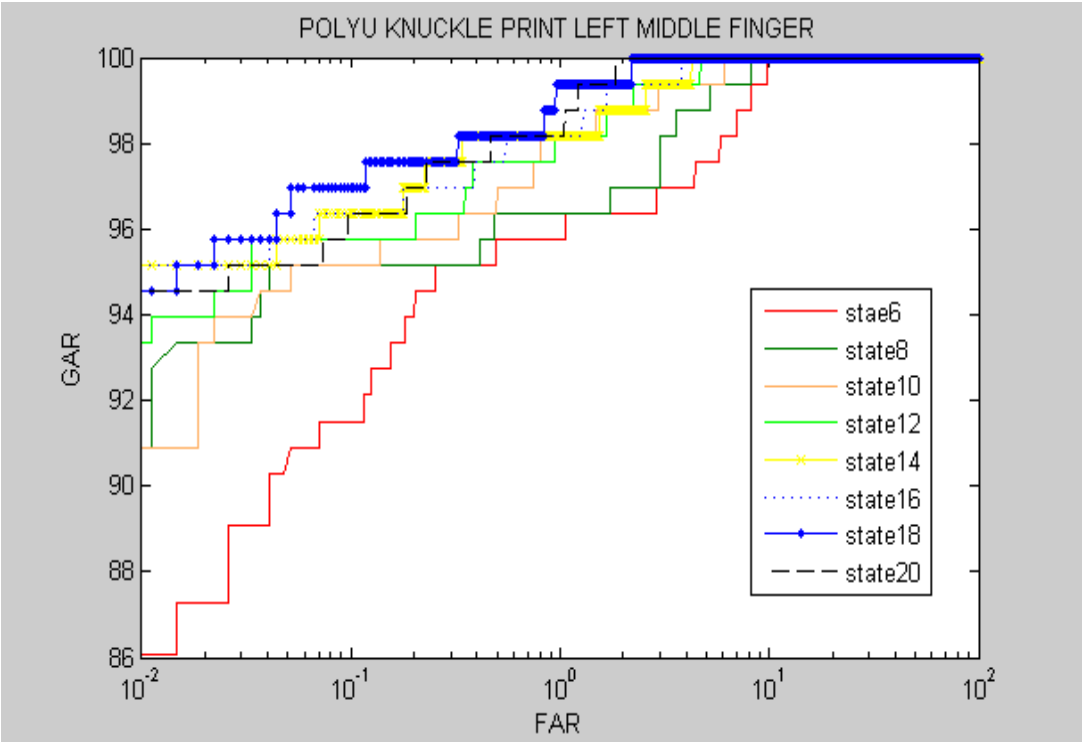

Figure 5.7: ROC for PolyU left middle finger knuckle print database at different states using HMM at feature level

Number of States	GAR at 0.01FAR
6	86
8	91
10	92
12	95
14	95.5

Table 6: GAR of PolyU left middle finger knuckle print database at different states using HMM at feature level

5.2 Results of Authentication at Score level using Fusion

At score level, the fusion of few score matrices is taken. The fusion rules used are: SUM, MIN, MAX and PRODUCT rule. A combination of Knuckle Finger Print score vectors and Palmprint score vectors is used to get the fused result. In this two sets of experiments were carried out. First involves a fusion of left index, left middle, right index and right middle finger knuckle genuine and imposters score vectors. In second, a fusion of left index, left middle, right index, right middle finger knuckle as well as palm genuine and imposter score vectors are used to get the fused score. All GAR values are computed at FAR = 0. The scores of 100 users were taken for the training purpose and 135 users were taken for the testing phase. The number of states were separately computed for genuine and imposter.

Modalities / Rules	Left index, Left middle, Right index and Right middle finger	Number of states for Genuine Scores	Number of states for Imposter Scores
SUM	**98.5**	**10**	**30**
MAX	**97.5**	**8**	**28**
MIN	**89**	**10**	**30**
PRODUCT	**85**	**12**	**30**

Table 7: GAR of PolyU finger knuckle using HMM at score level

Modalities / Rules	Left index, Left middle, Right index, Right middle finger and Palm	Number of states for Genuine Scores	Number of states for Imposter Scores
SUM	**99**	**12**	**30**
MAX	**98**	**8**	**28**
MIN	**93**	**14**	**32**
PRODUCT	**89**	**12**	**30**

Table 7: GAR of PolyU finger knuckle and PolyU palmprint database using HMM at score level

Chapter 6: Conclusion

The current works has focussed on extracting point features from the palmprint and finger knuckle print images using complex algorithms like Harris Corner Detection and SIFT descriptors and then classify them accordingly using Hidden Markov Model. In the proposed approach, feature vectors and score vectors are the observation vector. An iterative approach is proposed to calculate the suitable number of states. GMM along with unsupervised fitting algorithm is also used to calculate the correct number of components in each state and thus calculate the probability densities. These feature vectors are used as observation data for the HMM classifier. The observation is assumed to be consisting of some number of states. Each observation belongs to some states that are hidden.

At feature level, the observations were the feature vectors obtained after extracting features from the images. For PolyU Palmprint database, the correct number state was found to be giving a GAR of. For PolyU Right Middle Finger Knuckle and for PolyU Left middle Finger Knuckle, GAR was reported 97% and 95% with state value equal to 16 and 14 respectively. While in case of PolyU Left index and PolyU Right Index Finger Knuckle Print, GAR was quite low. It was 89% and 92.5% respectively with state value equal to 14. The experiments were also carried out with IITD palmprint database. Since it is a contactless database so the results were quite low as compared to contact based PolyU database. The GAR for IITD palmprint database was reported 73% with number of states 12.

Moreover, at feature level the GAR for Gabor convolved images was very less (65%) as compared to that obtained by using images that were not enhanced (92.5%). This shows the fact that HMM does not work well with Gabor convolved images as Gabor image matrix contains pixels with intensity values equal to 0. This makes it difficult for HMM to classify because HMM deals with numbers and statistics.

At score level, the data was fused using different fusion rules and appropriate number of states were calculated using the proposed iterative approach. Fusing palmprint and all the finger knuckle print databases gave high authentication results (GAR = 99%) using SUM-fusion rule. The results after fusing all the finger knuckle print images also reported a GAR of 98.5% and 97% with SUM and MAX fusion rules. The least authentication rates were recorded with MIN fusion rule. So, MIN rule cannot be considered as an efficient rule for giving high acquisition rates.

Thus, it can be concluded that palmprint database as well as knuckle print databases can be effectively used in a combination with complex algorithms like Harris Detector and HMM classifier to get high authentication rates. Moreover, it can also be concluded that the HMM classifier works efficiently with very complex data as well. Thus, personal authentication system using palm and knuckle print databases and classifying them at the score level, will be a very efficient authentication system.

References

1. D. Zhang, A. W.-K. Kong, J. You, and M. Wong, “Online palmprint identification”,IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 25, no. 9, pp. 1041–1050, Sep. 2003
2. W.Li, D. Zhang and Z.Xu, “Palmprint identification by Fourier transform,” International Journal on Pattern Recognition & Artificial Intelligence, vol. 16, no. 4, pp. 417-432, 2002.
3. Ross, Arun, and Anil K. Jain. "Multimodal biometrics: An overview."*Proceedings of 12th European Signal Processing Conference*. 2004.
4. Rabiner, Lawrence, and B. Juang. "An introduction to hidden Markov models."*ASSP Magazine, IEEE* 3.1 (1986): 4-16.
5. Lowe, David G. "Distinctive image features from scale-invariant keypoints."*International journal of computer vision* 60.2 (2004): 91-110.
6. C. Harris and M. Stephens, “A Combined Corner and Edge Detector”, In Proceedings of the Fourth Alvey Vision Conference, pp. 147-151, 1988
7. M. Figueiredo and A. K. Jain, “Unsupervised Learning of Finite Mixture Models,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 3, pp. 381–396, March 2002.
8. Ghahramani, Zoubin. "Unsupervised learning." *Advanced Lectures on Machine Learning*. Springer Berlin Heidelberg, 2004. 72-112.
9. Nandakumar, Karthik. *Integration of multiple cues in biometric systems*. Diss. Michigan State University, 2005.
10. Jain, Anil K., Arun Ross, and Salil Prabhakar. "An introduction to biometric recognition." *Circuits and Systems for Video Technology, IEEE Transactions on* 14.1 (2004): 4-20.
11. Qu, Zhong, and Zheng-yong Wang. "Research on preprocessing of palmprint image based on adaptive threshold and Euclidian distance." *Natural Computation (ICNC), 2010 Sixth International Conference on*. Vol. 8. IEEE, 2010.
12. Zhang, Lin, Lei Zhang, and David Zhang. "Finger-knuckle-print verification based on band-limited phase-only correlation." *Computer Analysis of Images and Patterns*. Springer Berlin Heidelberg, 2009.

13. Dale, Manisha P., Madhuri A. Joshi, and Neena Gilda. "Texture based palmprint identification using DCT features." *Advances in Pattern Recognition, 2009. ICAPR'09. Seventh International Conference* IEEE, 2009.
14. G. Lu, D. Zhang and K. Wang, "Palmprint recognition usingeigenpalm-like features", Pattern Recognition Letter, vol. 24, pp.1473–1477, 2003.
15. Sanchez-Reillo, Raul. "Hand geometry pattern recognition through gaussian mixture modelling." *Pattern Recognition, 2000. Proceedings. 15th International Conference on*. Vol. 2. IEEE, 2000.
16. E. Bigun, J. Bigun, B. Duc, S. Fischer, Expert conciliation for multimodal person authentication systems using Bayesian Statistics, in: First International Conference on AVBPA, Crans-Montana, Switzerland, 1997, pp. 291-300.
17. P. Verlinde, G. Cholet, Comparing decision fusion paradigms using k-NN based classifiers, decision trees and logistic regression in a multi-modal identity verification application, in: Second International Conference on AVBPA, Washington D.C., USA, 1999, pp. 188-193.
18. Zheng, Yufeng, and Adel Elmaghraby. "A brief survey on multispectral face recognition and multimodal score fusion." *Signal Processing and Information Technology (ISSPIT), 2011 IEEE International Symposium on*. IEEE, 2011.
19. Nandakumar, Karthik, et al. "Likelihood ratio-based biometric score fusion."*Pattern Analysis and Machine Intelligence, IEEE Transactions on* 30.2 (2008): 342-347.
20. Wu, Xiangqian, Kuanquan Wang, and David Zhang. "HMMs based palmprint identification." *Biometric Authentication*. Springer Berlin Heidelberg, 2004. 775-781.
21. Vuyyuru, Sampath K., et al. "Computer User Authentication using Hidden Markov Model through Keystroke Dynamics." *Manuscript submitted to ACM Transactions on Information and System Security* (2006).
22. Kulkarni, Mrs SS, and Mrs RD Rout. "Secure Biometrics: Finger Knuckle Print."
23. Forney Jr, G. David. "The viterbi algorithm." *Proceedings of the IEEE* 61.3 (1973): 268-278.
24. Harris, Chris, and Mike Stephens. "A combined corner and edge detector."*Alvey vision conference*. Vol. 15. 1988.
25. Ross, Arun A., and Rohin Govindarajan. "Feature level fusion of hand and face biometrics." *Defense and Security*. International Society for Optics and Photonics, 2005.

26. Csetverikov, Dmitrij. "Basic algorithms for digital image analysis." *Course, Institute of Informatics, Eotvos Lorand University, visual. ipan. sztaki. hu*(2003).
27. Lu, Xiaoguang. "Image analysis for face recognition." *personal notes, May* 5 (2003).
28. Lee, Lily, and W. Eric L. Grimson. "Gait analysis for recognition and classification." *Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on*. IEEE, 2002.
29. Simon-Zorita, Danilo, et al. "Minutiae extraction scheme for fingerprint recognition systems." *Image Processing, 2001. Proceedings. 2001 International Conference on*. Vol. 3. IEEE, 2001.
30. Wu, Chaohong, Zhixin Shi, and Venu Govindaraju. "Fingerprint image enhancement method using directional median filter." *Defense and Security*. International Society for Optics and Photonics, 2004.
31. Ross, Arun, and Anil Jain. "Information fusion in biometrics." *Pattern recognition letters* 24.13 (2003): 2115-2125.
32. Nandakumar, Karthik. *Multibiometric systems: Fusion strategies and template security*. ProQuest, 2008.
33. Faundez-Zanuy, Marcos. "Data fusion in biometrics." *Aerospace and Electronic Systems Magazine, IEEE* 20.1 (2005): 34-38.
34. Uludag, Umut. *Secure biometric systems*. Diss. Michigan State University, 2006.
35. Prabhakar, Salil, and Anil K. Jain. "Decision-level fusion in fingerprint verification." *Pattern Recognition* 35.4 (2002): 861-874.
36. Cremer, Frank, et al. "A comparison of decision-level sensor-fusion methods for anti-personnel landmine detection." *Information fusion* 2.3 (2001): 187-208.
37. Morales, Aythami, Miguel A. Ferrer, and Ajay Kumar. "Improved palmprint authentication using contactless imaging." *Biometrics: Theory Applications and Systems (BTAS), 2010 Fourth IEEE International Conference on*. IEEE, 2010.
38. Jain, Anil, Arun Ross, and Salil Prabhakar. "Fingerprint matching using minutiae and texture features." *Image Processing, 2001. Proceedings. 2001 International Conference on*. Vol. 3. IEEE, 2001.
39. D. Hall, *Mathematical Techniques in Multisensor Data Fusion*, Artech House, Norwood, MA., 1992.
40. Dumas, Bruno, et al. "Myidea-multimodal biometrics database, description of acquisition protocols." *In proc. of Third COST 275 Workshop (COST 275)*. 2005.

41. Iula, Antonio, Alessandro Savoia, and Giosuè Caliano. "Capacitive micro-fabricated ultrasonic transducers for biometric applications." *Microelectronic Engineering* 88.8 (2011): 2278-2280.
42. Frolova, Darya, and Denis Simakov. "Matching with invariant features." *The Weizmann Institute of Science, March* (2004).
43. Bilmes, Jeff A. "A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models."*International Computer Science Institute* 4.510 (1998): 126.
44. Manning C and Schotze H, "Foundations of Statistical Natural Language Processing," The MIT Press, Cambridge Massachusetts London, England [2002]
45. Forney Jr, G. David. "The viterbi algorithm." *Proceedings of the IEEE* 61.3 (1973): 268-278.
46. http://www4.comp.polyu.edu.hk/~csajaykr/myhome/papers/SMCC_2011.pdf
47. http://www4.comp.polyu.edu.hk/~biometrics/FKP.htm
48. http://www4.comp.polyu.edu.hk/~csajaykr/IITD/Database_Palm.htm