CHAPTER 3

PERFORMANCE ANALYSIS OF MULTIMODAL BIOMETRICS USING FINGERPRINT AND FINGER VEIN

3.1 INTRODUCTION

Multimodal biometrics is more advanced that it ensures (i) improved accuracy, (ii) secondary means of enrollment and verification or identification if sufficient data is not extracted from a biometric sample and (iii) can detect spoofing attempts through non-live data sources like fake fingers.

Higher accuracy and greater spoofing resistance are advantages of multimodal compared to unimodal biometrics. Multimodal biometrics use complementary information and make it difficult for intruders to simultaneously spoof various biometric traits of registered users. Also, non-universality is overcome as multiple traits ensure sufficient population coverage. Due to these advantages, multimodal biometrics systems are preferred over single modality though processing time, storage requirements and computational demands are higher.

3.2 FINGERPRINT FEATURES

Feature is a function of one or more measurements, each specifying an object’s quantifiable property and computed to quantify object’s significant characteristics. Features used presently are classified as:
General features: Application independent features like color, texture and shape. Depending on abstraction level, they are further divided into:

- **Pixel-level features**: Features calculated at every pixel, e.g. color, location.

- **Local features**: Features calculated on image band subdivision results on image segmentation or edge detection.

- **Global features**: Features calculated on entire image or image’s regular sub-area.

Domain-specific features: Application dependent features like human faces, fingerprints, and conceptual features which are usually low-level features domain specific synthesis.

Features are classified as low and high level features. Low level features are extracted from original images, while high-level feature extraction is low-level features dependent (Choras 2007).

Fingerprint biometrics identifies a user in a non-repudiated way. It enhances user convenience, reduces (even eliminates) credential management costs and provides user-specific provisioning. Biometric transactions are auditable and non-repudiated (Ayouband Rodriguez 2011). Hence, organizations adopt fingerprint biometrics for identity management.

A fingerprint has a pattern of interleaved ridges and valleys which flow smoothly in parallel and terminate or bifurcate. At a global level, a pattern exhibits many shapes called singularities classified into three types: loop, delta and whorl. At local level, ridges and valleys exhibit a specific shape called minutia. There are many types of minutiae, but practically only two types of minutiae - ridge ending and ridge bifurcation are considered.
Fingerprint features are divided into three levels (Zhao et al., 2010). Level 1 feature are defined by fingerprint ridge flow and general morphological information, e.g. ridge pattern type, ridge orientation field, and singular points. These are not unique to a finger and are used for indexing and fingerprint classification (whorl, left loop, right loop, and arch).

Level 2 features refer to discrete fingerprint ridges and fingerprint ridge events like minutiae. There are two prominent minutiae types, i.e. ridge endings and ridge bifurcations. Level 2 features are discriminative and stable. Level 3 features are fingerprint ridge dimensional features and include pores, dots, incipient ridges and ridge edge shapes.

Every finger epidermis ridge has tiny sweat pores and other permanent details. Pores are highly distinctive regarding number, position and shape. But, extracting pores is possible only in high-resolution fingerprint images (for e.g. 1000 DPI) and with very high image quality. Hence, this is not adopted by current automatic fingerprint identification systems (Wu 2007).

If a finger’s sensing area is small or finger placement on fingerprint sensor deviates from normal central contact, there might not be enough discriminating power for identification. This may be due to either discriminative regions not being included in image, or because detected minutiae number is not enough. Level 3 features increase Level 2 features discriminating capacity.

Various sensors are used for fingerprint extraction - optical, capacitive, ultrasound and thermal, which can collect a finger surface’s digital image. But, traditional live-scan methods were slow to adapt better capture areas required to digitize finger prints. Challenges for sensors to attain high-resolution finger print images are yet to be dealt with. A common approach,
which uses a capacitive sensor, determines pixel value based on measured capacitance.

This is possible due to an area of air (valley), with less capacitance value than palm area (ridge). Other fingerprint sensors capture images using high frequency ultrasound or optical devices, using prisms to detect change. Swiping action is needed in thermal sensors over surface to measure temperature difference to produce patterns. Ultrasound sensors use ultrasonic waves which are not user-friendly. They may damage to cells due to frequent exposure. So, capacitive sensors are the best bet.

### 3.3 FINGER VEIN FEATURES

Finger veins are good biometric patterns for personal identification regarding their advantages over current biometrics. Finger vein features matches the vascular pattern (blood vessel pattern) in an individual’s finger to previously obtained data. Finger veins as a biometric identifier have the following properties (Xi et al., 2013):

1. non-contact;
2. live-body identification;
3. high security; and
4. small device.

Finger vein recognition includes four steps: image capture, preprocessing, feature extraction and matching.
Finger–vein features have the following merits (Malik and Sharma 2013):

**Anti-counterfeit:** Finger veins underneath the skin make vein pattern duplication practically impossible.

**Active liveness:** Vein information vanishes with biological tissues losing liveness, making artificial veins unavailable in application.

**User friendliness:** Finger–vein images are captured non-invasively without contagion and unpleasant sensations.

Finger vein has ridge and valley lines. To show irregular shape in minutiae and singularity regions, ridge and valley lines have continuous and smooth change in most regions (Cui and Yang 2011). Finger vein minutiae are not as easy as fingerprint. Finger vein features are two-dimensional set of points. As finger vein thickness is susceptible to light and other factors, extracted finger vein lines are thick. It is necessary to thin them prior to feature extraction. Image thinning extracts a single pixel wide skeleton from image to maintain original topology. Junctions and end points are deemed minutiae extracted from finger vein images.

Performance measurement is important in biometrics. Four parameters used to measure performance are FRR (False Rejection Ratio) to recognize an impostor, FAR (False Acceptance Ratio) the ratio of impostor to be recognized as genuine, EER (Equal Error Rate) is when FAR and FRR are equal (less EER is better for system performance) and ROC (Receiver Operating Characteristic) is plot of FRR versus FAR (Razzak et al., 2011). Uniqueness and features reliability are two important factors that affect FAR and FRR.
3.4 METHODOLOGY

Figure 3.1 is the block diagram for the new methodology which are explained in detail in following sections

![Block diagram of new methodology](image)

3.4.1 Dataset

Five left index finger vein images from 100 subjects and five fingerprint images of left index finger from 100 subjects were experimented on. The fingervein images is obtained from the Hong Kong Polytechnic University (http://www4.comp.polyu.edu.hk/~csajaykr/fvdatabase.htm). The database consists of 6264 images from 156 subjects. Each subject provided six image samples from index and middle finger and one fingervein image and one finger texture image from left hand. All the images are stored in bitmap format. The sample index of the fingervein obtained is shown in Figure 3.2. The subject file names were given sequential numerical values and each of the subject was randomly selected using random seed.
The fingerprint images were obtained from Fingerprint Verification Competition 2004 (http://bias.csr.unibo.it/fvc2004/databases.asp). The dataset consists of a total of 120 fingers and 12 impressions per finger (1440 impressions). The sample fingerprints used for evaluation is shown in Figure 3.3. Matlab was extensively used for the implementation.
3.4.2 Wavelet Packet Transform for Feature Extraction

Wavelet packet method is a wavelet decomposition generalization ensuring wide signal analysis possibilities. In wavelet analysis, a signal is split into approximation and detail. Approximation is further split into a
second-level approximation and details the process. There are \( n+1 \) possible ways to decompose/encode signals for \( n \)-level decomposition (Amiri and Asadi 2009).

Wavelet packet analysis’ details and approximations split yield more ways to encode signals. For e.g. wavelet packet analysis permit signal \( S \) to be represented as \( A1+A6+D6+D3 \), which is impossible with ordinary wavelet analysis. Wavelet decomposition tree is part of a binary tree. Wavelet packet analysis is same as DWT, the only difference is that in addition to wavelet approximation component decomposition at each level, wavelet detail component decomposes ensuring own approximation/detail components as seen in Figure 3.4.

![Wavelet Packet decomposition tree](image)

**Figure 3.4 Wavelet Packet decomposition tree**

Wavelet packet tree’s components are viewed as filtered components with filter bandwidth decreasing with increased decomposition. The entire tree is viewed as a filter bank. Wavelet packet components time resolution is good at tree top, but it is at the expense of poor frequency resolution while at the bottom wavelet packet analysis ensures frequency resolution of decomposed component with high frequency, content increases. So, wavelet packet analysis ensures better frequency resolution control in signal decomposition (Abhijeet 2004). Wavelet packet is represented as a
function, \( \psi \) where ‘i’ is modulation parameter, ‘j’ dilation parameter and ‘k’ translation parameter

\[
\psi_{j,k}(t) = 2^\frac{j}{2} \psi(2^{-i}t - k)
\]  

(3.1)

where \( i = 1, 2 \ldots j_n \) and ‘n’ is wavelet packet tree decomposition level.

Wavelet packet offers a complex and flexible analysis, and represents a generalization of multi-resolution decomposition. Approximation component is decomposed in WT whereas in WPT, approximations and detailed components are decomposed (Chopra and Gupta 2011). The second stage, Quantization/Thresholding focused on selecting a value that satisfies HVS constraints for better visual quality and increased CR. The entropy encoder stage reduces overall number of bits needed to represent data set. It removes redundancy in repetitive bit pattern in quantizer output.

Figure 3.5 Fingervein Coefficient value for sample

WPT is a Dyadic Wavelet Transform (DWT) generalization offering a rich decomposition structures set. Wavelet packet decomposition is achieved when filter bank is iterated over frequency bands at all levels. The
final decomposition structure will be a subset of a full tree, chosen by best basis selection algorithm (Oktem 2000).

![Figure 3.6 The Tree Decomposition of the sample Fingervein](image)

The fingervein coefficient value achieved using 2D wavelet is depicted in Figure 3.5. The tree decomposition of the sample fingervein is shown in Figure 3.6 for the third level.

### 3.4.3 Gabor Features for Feature Extraction

Gabor filters are formed by modulating complex sinusoids through a Gaussian function. J.G Daugman (1980, ‘88, ‘90) showed that receptive fields of orientation receptive neurons in a (cat’s) brain looked similar to Gabor functions. As with Gabor filters, a brain uses complete, non-orthogonal functions.

Dennis Gabor introduced Gabor filter. Multiplication of a cosine/sine (even/odd) wave with Gaussian windows is a one-dimensional Gabor filter as follows,
\[ g_x(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-x^2}{2\sigma^2}} \cos(2\pi w_o x) \]

\[ g_o(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-x^2}{2\sigma^2}} \sin(2\pi w_o x) \]

(3.2)

Where \( w_o \) defines centre frequency (frequency where filter yields greatest response) and \( \sigma \) spread of Gaussian window (Derpanis 2007).

Gabor filter is got by modulating a sinusoid with Gaussian. A 1D sinusoid is modulated with Gaussian for 1D signals which responds to some frequency, but only in signal’s localized part. Let \( g(x,y,\theta,\phi) \) be function defining a Gabor filter centered at origin with \( \theta \) as spatial frequency and \( \phi \) as orientation. A Gabor filter is defined as

\[ g(x,y,\theta,\phi) = \exp\left(-\frac{x^2 + y^2}{\sigma^2}\right) \exp(2\pi i (x \cos\phi + y \sin\phi)) \]

(3.3)

It is shown that \( \sigma \), Gaussian kernel standard deviation depends on spatial frequency to be measured, i.e. \( \theta \).

A Gabor filter-based method directly extracts fingerprint features from grey-level images without pre-processing. Gabor filter-based features were successfully applied to face recognition, texture segmentation, handwriting recognition and fingerprint enhancement due to its characteristics specially frequency and orientation representations. Gabor filter-based features were directly extracted from grey-level fingerprint images.

Images are converted to gray scale using rg2bgray command in Matlab before processing. Gabor filters and fingerprint feature extraction: General 2D Gabor filter form is defined by
\[
    h(x, y, \theta_k, f, \sigma_x, \sigma_y) = \exp \left[ -\frac{1}{2} \left( \frac{x^2_{\theta_k}}{\sigma_x^2} + \frac{y^2_{\theta_k}}{\sigma_y^2} \right) \right] \\
    \times \exp \left( i2\pi fx_{\theta_k} \right)
\] (3.4)

Where \( x_{\theta_k} = x\cos\theta_k + y\sin\theta_k \) and \( y_{\theta_k} = -x\sin\theta_k + y\cos\theta_k \), \( f \) is frequency of sinusoidal plane wave, \( \theta_k \) is Gabor filter orientation and \( \sigma_x \) and \( \sigma_y \) standard deviations of Gaussian envelope along \( x \) and \( y \) axes, respectively.

To analyse Gabor filter regarding even-symmetric and odd-symmetric.

Parameters of Gabor are shown in Table 3.1.

**Table 3.1 Parameters of Gabor Filter**

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wavelength</td>
<td>8</td>
</tr>
<tr>
<td>Orientation (s)</td>
<td>0</td>
</tr>
<tr>
<td>Phase offset (s)</td>
<td>0, 90</td>
</tr>
<tr>
<td>Aspects Ratio</td>
<td>0.5</td>
</tr>
<tr>
<td>Number of Orientations</td>
<td>4</td>
</tr>
</tbody>
</table>

Parallel ridges and valleys configurations with well-defined orientation and frequency in Finger Vein image ensure information to remove undesired noise. Sinusoidal-shaped ridges and valleys waves, fluctuate slowly in local constant orientation. So, a band pass filter tunes corresponding frequency. Orientation efficiently also removes undesired noise, preserving true ridge and valley structures. Gabor filters have frequency-selective and orientation-selective properties and optimal joint resolution in spatial and frequency domains and so it is appropriate to use Gabor filters as band pass filters for noise removal and true ridge and valley structures preservation (Yu et al., 2009).
In spatial domain, a circular 2-D Gabor filter has a general form (Kong et al., 2003):

\[
G(x, y, \theta, u, \sigma) = \frac{1}{2\pi\sigma^2} \exp \left\{ \frac{-x^2 + y^2}{2\sigma^2} \right\} \exp \{2\pi i (ux \cos \theta + uy \sin \theta)\}
\]  

(3.5)

Where \( i = -1 \); \( u \) is frequency of sinusoidal wave; \( q \) controls orientation of function and \( s \) is standard deviation of Gaussian envelope.

![Figure 3.7 The Gabor orientation obtained for a sample fingerprint; a. 0 degree gabor orientation, b. 45 degree orientation, c. 90 degree orientation](image)

The gabor features extracted for first image shown in figure 3.3 is seen in Figure 3.7. The 0 degree gabor orientation, 45 degree orientation, and 90 degree orientation is depicted.
3.4.4 Principal Component Analysis for Feature Selection

PCA is a dimensionality reduction and feature extraction technique. It tries to locate original feature space’s lower dimensionality linear subspace where new features have large variance called dimensionality reduction where a vector with original data and N-dimensional is reduced to compressed vector which is M-dimensional, where M<N. A vector is coded to a vector with reduced dimension. Vector is stored, transmitted or processed, resulting in vector that is decoded back to vector. The last vector is a result approximation attained by storing, transmitting or processing vector (Jolliffe 2005).

\[
\bar{c} = \overline{Q} \bar{x}
\]

Diagram’s encoder must perform a linear operation, using a matrix \( \overline{Q} \):

\[
\bar{c} = \overline{Q} \bar{x}
\]

Diagram’s encoder must ensure a linear operation, using a multiplied by matrix columns

\[
\overline{Q} : \hat{x} = \bar{c}^T \overline{Q} \rightarrow \hat{x} = \sum_{i=1}^{M} c_i q_i
\]

\[(3.6)\]

PCA uses a linear transformation to ensure a simplified data set retaining original dataset characteristics. Assuming that original data matrix has d dimensions and n observations it is ordered that dimensionality be
reduced to a k dimensional subspace (Deegalla and Bostrom 2006). This is given by

\[ Y = E^T X \]

Here \( E_{d \times k} \) is projection matrix having k Eigen vectors equivalent to k highest Eigen values, and where \( X_{d \times n} \) is a mean centered data matrix. Generally, PCA’s is to re-express original dataset in new basis hoping that new basis filter out noise present in data and reveal part of structure underlying data (Okal 2011).

PCA strength for data analysis is due to efficient computational mechanism and due to being understood and also due to applicability. PCA transforms initial data set represented by vector samples to new vector samples set with derived dimensions. The following is a description: a set of n-dimensional vector samples \( X = \{x_1, x_2, x_3, ..., x_m\} \) ought to be transformed into additional set \( Y = \{y_1, y_2, ..., y_m\} \) of same dimensionality, but y-s have properties where most information content is stored in first few dimensions. Hence, data set is reduced to smaller dimensions with reduced information loss (Novakovic and Rankov 1841).

Eigen vectors are ranked according to original data variation they account for. The first few transformed attributes account for most data set variation but are retained, with remainders being discarded. PCA, an unsupervised method does not use information embodied in class variable. As PCA returns original features linear combinations, original features meaning is not preserved. There are extensions to conventional PCA over years. PCA algorithm is as follows:

Traditional PCA allows linear dimensionality reduction, but when data has complicated structures that cannot be simplified in linear subspace, it
becomes invalid. But, kernel PCA permits conventional PCA generalization to nonlinear dimensionality reduction

Recover basis:

Calculate $XX^T = \sum_{i=1}^{t} x_i x_i^T$ and let $U = \text{eigenvectors of } XX^T$ corresponding to the top $d$ eigenvectors

Encode training data:

$Y = U^T X$ where $Y$ is a $d \times t$ matrix of encodings of the original data.

Reconstruct training data:

$\hat{X} = UY = UU^T X$

Encode test examples:

$y = U^T x$ where $y$ is a $d$-dimensional encoding of $x$

Reconstruct test example:

$\hat{x} = Uy = UU^T x$

Kernel PCA was introduced (Honkela et al., 2004) as a nonlinear PCA generalization the aim being to map data points from input space $\mathbb{R}^n$ to high-dimensional (possibly infinite-dimensional) feature space $F$

$$\Phi = \mathbb{R}^n \rightarrow \mathcal{F}$$ (3.7)

and perform PCA in $\mathcal{F}$. Space $\mathcal{F}$ and also mapping $\Phi$ is complicated. Employing the kernel trick, kernel PCA avoids using $\Phi$: PCA in $\mathcal{F}$ is formulated that only $\mathcal{F}$’s inner product is needed. This is seen as a nonlinear kernel function

$$\mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$$

$$(x, y) \rightarrow k(x, y),$$ (3.8)

which calculates real number for each vector pair from input space.
Simple PCA models are linear variability, efficient in high-dimensional data. But such data sets are nonlinear. Then high-dimensional data lies on or near a nonlinear manifold. So, PCA cannot model data variability. Kernel PCA finds principal components nonlinearly related to input space by performing PCA in space by nonlinear mapping, where low-dimensional latent structure is found.

Consider a feature space $H$ so that:

$$\phi : x \rightarrow H$$

$$x \mapsto \phi(x)$$

The objective of kernel PCA is,

$$\min \sum \| \phi(x_i) - U_q U_q^T \phi(x_i) \|$$

The solution is found by SVD.

$$\phi(X) = U \sum V^T$$

### 3.4.5 Classifiers

#### 3.4.5.1 Naïve Bayes classifier

Naïve Bayes classifiers are Bayes theorem based statistical classifiers (McCallum and Nigam 1998) using a probabilistic approach to predict a data’s class, by matching data to class with highest posterior probability. Following algorithms are used in Naïve Bayes:

$$P(C_j|V) = \frac{P(V|C_j)P(C_j)}{P(V)} \quad (3.9)$$
Where \( v = (v_1, \ldots, v_n) \) is document represented in \( n \)-dimensional attribute vector and \( c_1, \ldots, c_m \) represents \( m \) class, but is computationally expensive to compute. \( P(V|c_i) \) to reduce computation, hence, naïve assumption of class conditional independence is made. Thus,

\[
P(V|c_i) = \prod_{i=1}^{n} P(x_i|c_i)
\]

(3.10)

### 3.4.5.2 k Nearest Neighbor (KNN)

k-NN classifier is based on the premise that vector space model is similar for same documents. Training documents are indexed, each being associated with corresponding label. When a test document is submitted, it is a query and it retrieves from training set, documents similar to test document. The latter class label is assigned based on k-NN distribution. Class label is refined by weights addition. So, higher accuracy is ensured by tuning. k-NN method is simple and easy to implement (Kulkarni et al., 1998; Timofeev 2004).

\[
p(x) = \frac{k}{NV}
\]

Similarly probability density function \( p(x|H_i) \) of observation \( x \) conditioned to hypothesis \( H_i \) is approximated. Let us assume \( N_i \) is number of patterns associated with hypothesis \( H_i , i=1 \ldots C \), so that \( N_1 + \cdots + NC = N \).

### 3.4.5.3 Radial Basis Function (RBF) classifier

A Radial Basis Function (RBF) classifier is a three layer feed-forward network of an input layer, a middle layer and an output layer. Every input neuron corresponds to input vector \( x \) component. Middle layer has \( n \)
neurons and one biased neuron. Every input neuron is connected to middle layer neurons except to the biased one.

**RBF** is Gaussian curve based. It takes a parameter to determine center (mean) value of function used as desired value. RBF is a real-valued function whose significance depends on distance from origin, so that (Karlik and Olgac 2010),

\[ g(x) = g\left(\| x \|\right) \]

or alternatively on distance from some other point \( c \), called a center, so that

\[ g(x, c) = g\left(\| x - c \|\right) \]

RBFs sums approximate given functions which are interpreted as a simple neural network. RBF normally builds up function form approximations

\[ y(x) = \sum_{i=1}^{N} w_i g\left(\| x - c_i \|\right) \]

where approximating function \( y(x) \) is represented as sum of \( N \) RBFs, each being associated with a diverse center \( c_i \), and weighted by suitable coefficient \( w_i \). Weights \( w_i \) is estimated using linear least squares matrix, as approximating function is linear in weights.

Each middle layer neuron computes a kernel function (activation function) which is usually the following Gaussian function

\[ y_j = \begin{cases} \exp\left( -\frac{x - c_j}{2\sigma_i^2} \right) & i = 1, 2, \ldots, n \\ 1 & i = 0 \ (bias \ neuron) \end{cases} \]

(3.11)
Where $c_i$ and $s_i$ are the center and width of $i$-th neuron in middle layer, respectively. $k$ denotes Euclidean distance (Hwang and Bang 1997). Weight vector between input layer and $i$-th middle layer neuron correspond to center $c_i$ in equation (1). And in an RBFN net input to $i$-th middle layer neuron is $\|x - c_i\|$ rather than $x \cdot c_i$ the kernel function quickly decreases if width $s_i$ is small, and slowly if large. Output layer includes $m$ neurons corresponding to problem’s possible classes and is connected to middle layer. Every output layer neuron computes a linear weighted sum of outputs of middle layer as follows

$$Z_j = \sum_{i=0}^{n} y_i w_{ij}, \quad j = 1, 2, \ldots, m$$

Where $w_{ij}$ is weight between $i$-th middle layer neuron and $j$-th output layer neuron.

### 3.5 EXPERIMENTAL SETUP

Fingerprint and finger vein based Multimodal biometrics are considered. For finger vein images, energy coefficients are collected using wavelet packet tree. Gabor features extraction from fingerprint is by Gabor filter with orientation of 0, 15, 45, 60 and 75 degrees using parameters shown in figure 3.1. Both obtained features are normalized with min max normalization and fused with concatenation. Feature selection is through PCA and kernel PCA. Classification is through kNN, Naive Bayes and RBF Neural Network Classifier.

Step 1: Extract features from finger vein using Wavelet Packet tree

Step 2: Extract features from fingerprint using Gabor Filter

Step 3: Normalize the features with min-max normalization and fuse
Step 4: Select the features using PCA and Kernel PCA

Step 5: Classify the features using k-NN, Naive Bayes and RBF

3.6 RESULT AND DISCUSSION

The results obtained are shown from Figure 3.9 - 3.12.

<table>
<thead>
<tr>
<th>PCA based features</th>
<th>Recognition rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of features</td>
<td>K-NN</td>
</tr>
<tr>
<td>20</td>
<td>84</td>
</tr>
<tr>
<td>40</td>
<td>86.4</td>
</tr>
<tr>
<td>60</td>
<td>88.4</td>
</tr>
<tr>
<td>80</td>
<td>88.6</td>
</tr>
<tr>
<td>100</td>
<td>89</td>
</tr>
<tr>
<td>120</td>
<td>89</td>
</tr>
<tr>
<td>140</td>
<td>89</td>
</tr>
</tbody>
</table>

From Figure 3.9 it is shown that the proposed PCA based feature RBFNN classifier has higher recognition rate of 93.4%. 

Figure 3.9 Recognition rate - PCA based features
Table 3.3 Recognition rate based on kernel PCA

<table>
<thead>
<tr>
<th>PCA based features</th>
<th>Recognition rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of features</td>
<td>K-NN</td>
</tr>
<tr>
<td></td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>40</td>
</tr>
<tr>
<td></td>
<td>60</td>
</tr>
<tr>
<td></td>
<td>80</td>
</tr>
<tr>
<td></td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>120</td>
</tr>
<tr>
<td></td>
<td>140</td>
</tr>
</tbody>
</table>

From Figure 3.10 it is shown that the proposed Kernel PCA based feature RBFNN classifier has higher recognition rate of 95.8%. 

Figure 3.10 Recognition rate - Kernel PCA based features
Figure 3.11 ROC for Kernel PCA features

Figure 3.11 shows the false acceptance rate of ROC kernel-PCA. This ROC is the low level of the false Acceptance Rate to Genuine Acceptance Rate for different classifiers KNN, RBF, Naive Bayes.

Figure 3.12 ROC for PCA features
Figure 3.12 shows the false acceptance rate of ROC – PCA. This Figure represents the high ROC curve of the false Acceptance Rate to Genuine Acceptance Rate for different classifiers KNN, RBF, Naïve Bayes. ROC curves are popular with information content measurement of various imaging systems. Also, ROC curves judge various statistical methods discrimination ability that combine various clues and test results, for predictive purposes (e.g., to determine if a patient needs hospitalization or will benefit from treatment). In most cases, ROC curves are plotted and evaluated qualitatively with little attention being paid to statistical characteristics for various reasons. It can be seen that the GAR and FAR intersect at high values indicating the efficiency of the proposed technique.

ROC displays a marker or test’s discriminatory capacity. If $D = 0$ denotes controls and $D = 1$ denotes cases and assumes without generality loss that larger values of $Y$ are indicative of a subject being a case. ROC curve for a marker, $Y$, is a plot of true positive rate $TPR(c) = P [Y \geq c|D = 1]$ versus false positive rate $FPR(c) = P [Y \geq c|D = 0]$ for threshold criterion ‘$Y \geq c$’ where $c$ varies from $-\infty$ to $\infty$. It is a monotone increasing function in unit square tied down at boundary points (0,0) and (1,1). A perfect classifier has a ROC curve rising steeply along left axis to point (FPR=0, TPR=1), while an uninformative marker has a ROC curve diagonal 45° line A ROC curve’s key attributes are: (i) it does not depend on raw measurement units for $Y$ and is invariant to monotone increasing transformations of $Y$; (ii) it ensures a common scale to compare different markers performances; and (iii) it displays a range of performance levels achieved by varying threshold. ROC curve estimation is through two steps: (1) Estimating reference cumulative distribution function (CDF), $F$, using controls; and calculating corresponding standardized marker values for cases, and (2) estimating cumulative distribution of cases’ standardized marker values.
3.7 SUMMARY

Multimodal biometrics is popular due to performance and security. This study reveals multimodal biometrics systems related issues. Combining various biometric traits increases system performance. This study considers fingerprint and finger vein based multimodal biometrics. Gabor features extraction from fingerprint is by using Gabor filter with orientation of 0, 15, 45, 60 and 75 degrees. Energy coefficients are obtained using wavelet packet tree for finger vein images. Features are normalized through min max normalization and fused via concatenation. Feature selection is through PCA and kernel PCA. Classification is by kNN, Naïve Bayes and RBF Neural Network Classifiers. Results revealed that the new PCA based feature RBFNN classifier had 93.4% higher recognition rates while Kernel PCA based feature RBFNN classifier had 95.8% higher recognition rate.