Volume 2, No. 9, September 2011

Journal of Global Research in Computer Science

ISSN-2229-371X

TECHNICAL NOTE

Available Online at www.jgrcs.info

STATISTICAL BIOMETRIC METHODS

Sushma Jaiswal*[1], Dr. Sarita Singh Bhadauria[2] and Dr.Rakesh Singh Jadon
*[1]Lecturer, S.O.S. in Computer Science,
Pt.Ravishankar Shukla University,Raipur(C.G.),
jaiswal1302@gmail.com
[2]Professor & Head,Department of Electornics Engineering, Madhav Institute of Technology & Science,Gwalior(M.P.)
[3]Professor & Head,Department of Computer Applications, Madhav Institute of Technology & Science,Gwalior(M.P.)

Abstract: Biometric systems have been researched intensively by many organization. It overcomes the conventional security systems by identify "who you are". This paper discusses the current image based biometric systems. It first gives some information about why biometric is needed and what should people look for in biometric systems. Several popular image based biometric systems have been examined in this paper. The technique used in each system for data acquisitions, feature extraction and classifiers are briefly discussed. The biometric systems included are face, fingerprint, hand geometry, hand vein, iris, retina and signature; here also describe the statistical approaches of biometrics. The paper concludes by examining the benefits of multi-modal biometric systems, it is found that there is no one good biometric systems each have its advantages and disadvantages and the performance of each biometric system is summarized.

INTRODUCTION

As technology advances and information and intellectual properties are wanted by many unauthorized personnel. As a result, many organizations have being searching ways for more secure authentication methods for user access. Furthermore, security has always been an important concern to many people. It is soon realized by many, that traditional security and identification are not sufficient enough, people need to find a new authentic system in the face of new technological reality [1].

Conventional security and identification systems are either knowledge based – like a social security number or a password, or token based – such as keys, ID cards. The conventional systems can be easily breached by others, ID cards and passwords can be lost, stolen or can be duplicated. In other words, it is not unique and not necessary represent the rightful user. Therefore, biometric systems are under intensive research for this particular reason.

What is Biometric:

Humans recognize each other according to their various characteristics for ages. People recognize others by their face when they meet and by their voice during conversation. These are part of biometric identification used naturally by people in their daily life.

Biometrics relies on "something you are or you do", on one of any number of unique characteristics that you can't lose or forget. It is an identity verification of living, human individuals based on physiological and behavioral characteristics. In general, biometric system is not easily duplicated and unique to each individual. It is a step forwards from identify something you have and something you know, to something you are [2].

General biometric system:

Biometrics uses physical characteristics, defined as the things we are and personal traits, it can consists of following [1],

Table 1. Biometric characteristics.

Physical characteristics	Personal traits
• chemical composition of body odor • facial features and thermal emissions • features of the eye - retina and iris • fingerprints • hand geometry • skin pores • wrist/hand veins	• handwritten signature • keystrokes or typing • voiceprint

Same as many recognition systems, a general biometric system can consists of following sections, data collection, transmission, signal processing storage and decision [3], see Figure 1. It can considered that each section function independently, and errors can be introduced at each point in an additive way.

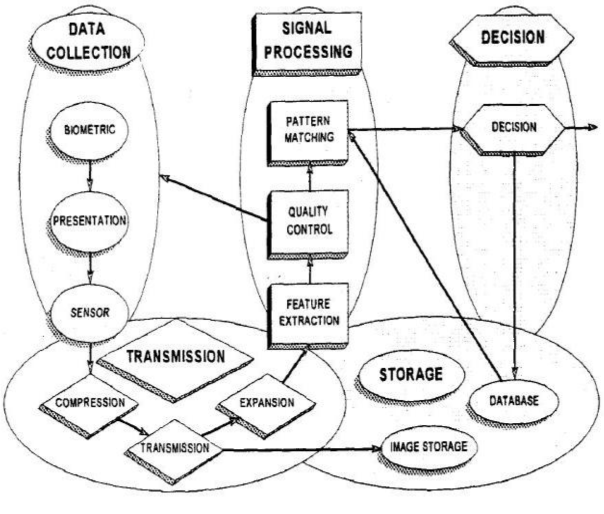


Figure 1. Generalized biometric system.

Data collection consists of sensors to obtain the raw biometric of the subject, and can output one or multidimensional signal. Usually, data are obtained in a normalized fashion, fingerprints are moderately pressed and rotation is minimized, faces are obtained in frontal or profiled view, etc. Data storage is usually separated from point of access, therefore the data have to be transmitted or distributed via a communication channel. Due to bandwidth, data compression may be required. The signal processing module takes the original biometric data and converts it into feature vectors. Depend on the applications, raw data might be stored as well as the obtained feature vectors. The decision subsystem compares the measured feature vectors with the storage data using the implemented system decision policy. If measures indicate a close relationship between the feature vector and compared template, a match is declared [3].

False matching and false non-matching error can occur, although for different systems error equation varied, a general equation can be developed [3][4]. Let M be the number of independ biometric measures the probability of false match FMR_{SR} against any single record can be given by,

$$FMR_{SR} = \prod_{j=1}^{M} FMR_j(\tau_j)$$

Where $FMR_j(\tau_j)$ equal single comparison false match rate for the j^{th} biometric and threshold τ. The probability for not making any false match in comparison in multiple records can be expressed as,

$$1 - FMR_{SYS} = (1 - FMR)^{PN}$$

Where FMR_{SYS} is the system false match rate, and N and P is number of records and percentage of the database to be searched respectively. For the single record false non-match rate,

$$1 - FNMR_{SR} = \prod_{j=1}^{M} [1 - FMR_j(\tau_j)]$$

More commonly used biometric system reliability indexes are FRR (False Reject Rate) which is the statistical probability that the system fails to recognize an enrolled person and FAR (False Accept Rate) which is the statistical probability that an imposter is recognized as an enrolled person. FRR and FAR are inversely dependent on each other and as within modern biometric systems identification:

$$FAR \in [0.0001\,\%,\ 0.1\%]$$

$$FAR \in [0.0001\,\%,\ 5\%]$$

They are very reliable, promising, universal and tampering resistant.

Biometric Techniques:

There are many biometric systems based on different characteristics and different part of the human body. However, people should look for the following in their biometrics systems [5],

a. Universality - which means that each person should have the characteristic
b. Uniqueness - which indicates that no two persons should be the same in terms of the characteristic
c. Permanence - which means that the characteristic should not be changeable
d. Collectability - which indicates that the characteristic can be measured quantitatively

From the above, various image based biometric techniques has been intensively studied. This paper will discuss the following techniques, face, fingerprints, hand geometry, hand veins, iris, retina and signature.

Face:

Face recognition technology (FRT) has applications in many wide ranges of fields, including commercial and law enforcement applications. This can be separate into two major categories. First is a static matching, example such as passport, credit cards, photo ID’s, driver’s licenses, etc. Second is real-time matching, such as surveillance video, airport control, etc. In the psychophysical and neuroscientific aspect, they have concerned on other research field which enlighten engineers to design algorithms and systems for machine recognition of human faces. A general face recognition problem can be, given still or video images of a scene to identify one or more persons in the scene using a stored database of faces [6]. With additional information such as race, age and gender can help to reduce the search. Recognition of the face is a relatively cheap and straightforward method. Identification based on acquiring a digital image on the target person and analyzing and extracting the facial characteristics for the comparison with the database.

Karhunen-Loeve (KL) expansion for the representation and recognition of faces is said to generate a lot of interest. The local descriptors are derived from regions that contain the eyes, mouth, nose, etc., using approaches such as deformable templates or eigen-expansion. Singular value decomposition (SVD) is described as deterministic counterpart of KL transform. After feature extraction, recognition is done, early approach such as feature point distance or nearest neighbor rule is used, and later eigenpictures using eigenfaces and Euclidean distance approach is examined. Other methods using HyperBF network a neural network approach, dynamic link architecture, and Gabor wavelet decomposition methods are also discussed in [6].

A back-propagation neural network can be trained to recognize face images. However, a simple network can be very complex and difficult to train. A typical image recognition network requires $N = m$ x n input neurons, one for each of the pixels in an m x n image. These are mapped to a number of hidden-layer neurons, p [7]. These in turn map to n output neurons, at least one of which is expected to fire on matching a particular face in the database. The hidden layer is considered to be a feature vector.

Eigenfaces is an application of principal component analysis (PCA) of an n-dimensional matrix. Start with a preprocessed image $I(x,y)$, which can be considered as vector of dimension N^2. An ensemble of images then maps to a collection of points in this huge space. The idea is to find a small set of faces (eigenfaces) that can approximately represent any point in the face space as a linear combination. Each of the eigenfaces is of dimension N x N, also can be interpreted as an image [7]. An image can be reduced to an eigenvector $\vec{B} = b_i$ which is the set of best-fit coefficients of an eigenface expansion. Eigenvector is then used to compare each of those in a database through distance matching, such as Cartesian distance.

Gabor wavelet is another widely used face recognition approach, it can be described by the equation,

$$\psi_{\vec{k},\sigma}(\vec{x}) = \exp\left(-\frac{k^2\,|\vec{x}|^2}{2\sigma^2}\right)\exp(i\vec{k}\vec{x})$$

where *k* is the oscillating frequency of the wavelet, and the direction of the oscillation. *σ* is the rate at which the wavelet collapses to zero as one moves from its center outward. The main idea is to describe an arbitrary two-dimensional image function *I*(*x*,*y*) as a linear combination of a set of wavelets. The *x*, *y* plane is first subdivided into a grid of non-overlapping regions. At each grid point, the local image is decomposed into a set of wavelets chosen to represent a range of frequencies, directions and extents that "best" characterize that region [7]. By limiting *k* to a few values, the resulting coefficients become almost invariant to translation, scale and angle. The finite wavelet set at a particular point forms a feature vector called a *jet*, which characterize the image. Also with elastically distorted grid, best match between two images can be obtained [7].

HOW FACIAL RECOGNITION SYSTEMS WORK

A facial recognition system consist of cameras that capture images of people who pose or simply walk by, and a software that matches those pictures. Following are the details of how these systems work, in general.

Basic Steps of a Facial Identification Process:

Although facial recognition methods may vary, but they generally involve a series of steps that serve to capture, analyze and compare your face to a database of stored images. These basic steps are:

Detection: When the system is attached to a video surveillance system, the recognition software keeps searching the field of view of a video camera for faces. For each head-like shape it detects, the system processes it.

Alignment: Once a face is detected, the system determines the head's position, size and pose. A face needs to be turned at least **35 degrees** toward the camera for the system to register it.

Normalization: The image of the head is scaled and rotated so that it can be registered and mapped into an appropriate size and pose. Normalization is performed regardless of the head's location and distance from the camera. Light does not impact the normalization process.

Representation: The system translates the facial data into a unique code. This coding process allows for easier comparison of the newly acquired facial data to stored facial data.

Matching: The newly acquired facial data is compared to the stored data and (ideally) linked to at least one stored facial representation.

What facial data the system takes?

Assuming that the system has detected a head and has done with the alignment and normalization processes, what facial data does it take into consideration before it translates that data into a unique code? Different systems may analyze a face that they have captured differently. Let's look at Visonics's way because it's one of the leading companies specialized in developing facial recognition systems. Visionics' facial recognition system, FaceIt, measures a face according to its peaks and valleys, which are called nodal points. The tip of the nose, the depth of the eye sockets, distance between eyes, width of nose, cheekbones, jaw line, and chin are some examples of nodal points, and "there are about 80 nodal points on a human face". The software concentrates on the inner region of the face called "golden triangle", which runs from temple to temple and just over the lip; this is the most stable area because even if you grow beard, put on classes, put on weight or age, that region tends not to be affected. FaceIt only needs from 14 to 22 nodal points to complete a recognition process, according to Kevin Bonsor's article "How Facial Recognition Systems work." This software takes the measurements and creates a numerical code called a faceprint, which is made up of a string of numbers and it represents a face (as step #4 above).

How is the matching process done?

The faceprints stored in the database represents the people (or criminal) we're looking for and will be compared against the ones of the people passing in front of the cameras to find a match. "The FaceIt system can match multiple faceprints at a rate of 60 million per minute from memory or 15 million per minute from hard disk." As comparisons are made, the system assigns a value to the comparison using a scale of one to 10. If a score is above predetermined threshold, a match is declared. The operator then views the two photos that have been declared a match to be certain that the computer is accurate.

Fingerprints:

One of the oldest biometric techniques is the fingerprint identification. Fingerprints were used as a means of positively identifying a person as an author of the document and are used in law enforcement. Fingerprint recognition has a lot of advantages, a fingerprint is compact, unique for every person, and stable over the lifetime. A predominate approach to fingerprint technique is the uses of minutiae [8], see Figure 2.

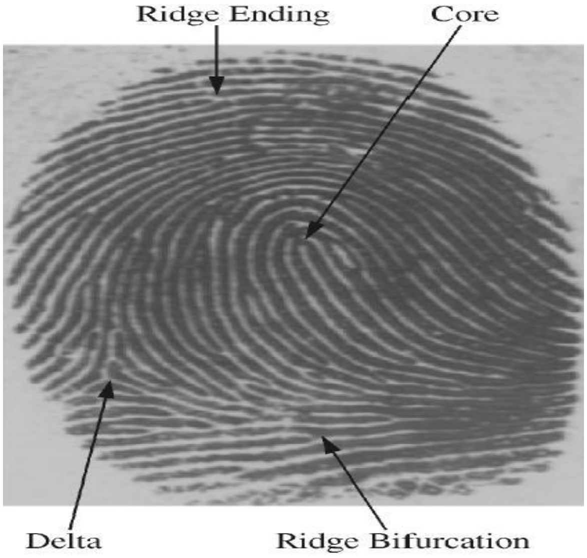


Figure 2. Minutiae, ridge endings and ridge bifurcations.

The traditional fingerprints are obtained by placing inked fingertip on paper, now compact solid state sensors are used. The solid state sensors can obtain patterns at 300 x 300 pixels at 500 dpi, and an optical sensor can have image size of 480 x 508 pixels at 500 dpi [9]. A typical algorithm for fingerprint feature extraction contains four stages, see in Figure 3.

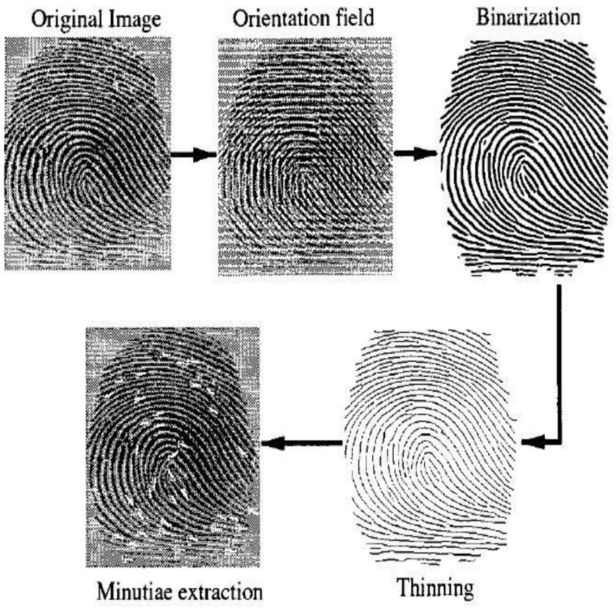


Figure 3. Typical fingerprint feature extraction algorithm.

The feature extraction first binarizes the ridges in a fingerprint image using masks that are capable of adaptively accentuating local maximum gray-level values along a direction normal to ridge direction [10]. Minutiae are determined as points that have one neighbor or more than two neighbors in skeletonized image [10].

Feature extraction approach differs between many papers, one simple minutiae extraction can be by applying the following filter, where resulting of 1 means ending, 2 a ridge, and 3 a bifurcation [8].

$$Minutiae\,Filter = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Or one might have the following filter [11], where $R_1 = R_9$

$$Minutiae\,Filter = \begin{array}{|c|c|c|} \hline R_1 & R_2 & R_3 \\ \hline R_8 & M & R_4 \\ \hline R_7 & R_6 & R_5 \\ \hline \end{array}$$

$\sum_{k=1}^{8} |R(k+1) - R(k)| = 2$, pixel *M* is a end point

$\sum_{k=1}^{8} |R(k+1) - R(k)| = 6$, pixel *M* is a bifurcation

However more complicated feature extraction such as [9], [10] applied Gabor filters. [9] uses a bank of 8 Gabor filter with same frequency,0.1 pix^{-1}, but different orientations (0° to 157.5° in steps of 22.5°). The frequency is chosen based on average inter-ridge distance in fingerprint, which is ~10 pixels. Therefore, there are 8 feature values for each cell in tessalation, and are concatenate to form 81 x 8 feature vector. In [10] the frequency is set to average ridge frequency (1/*K*), where *K* is the average inter-ridge distance. The Gabor filter parameters δ_x and δ_y are set to 4.0, and orientation is tuned to 0°. This is due to the extracted region is in the direction of minutiae. In general the result can be seen in Figure 4.

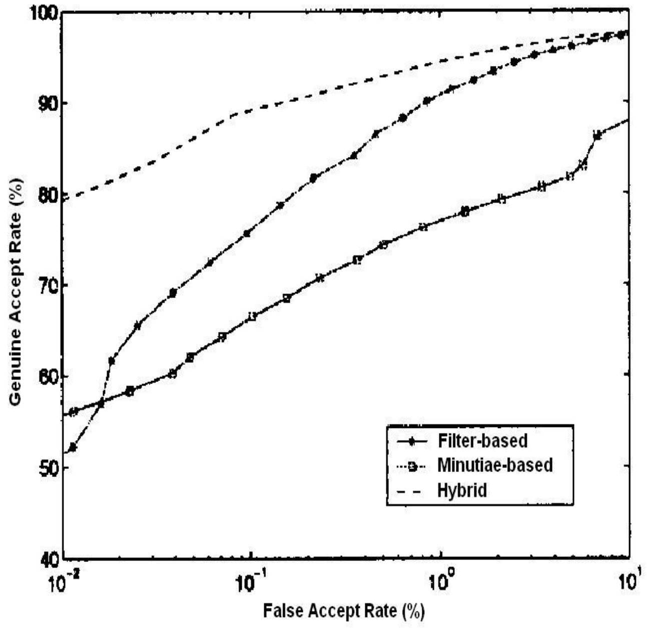


Figure 4. The ROC curve comparison.

Other enhancement algorithm such as preprocessing, mathematic algorithm and etc, have been discussed by [12], [13] and [14].

Finger print Recognition:

Among all the Biometric techniques, fingerprint based identification is the oldest method which has been successfully used in numerous applications. Every person’s fingerprint is unique and is a feature that stays with the person throughout his/her life. This makes the fingerprint the most reliable kind of personal identification because it cannot be forgotten, misplaced, or stolen. Fingerprint authorization is potentially the most affordable and convenient method of verifying a person's identity. This fact has been utilized for restricting the access to individuals in high security areas.

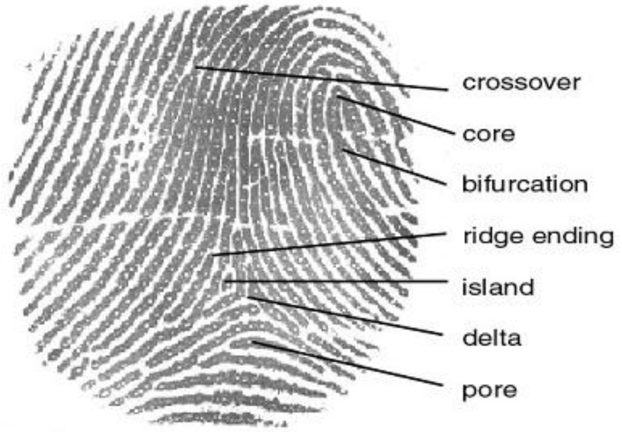
The Basics of Fingerprint Identification:

The skin on the inside surfaces of our hands, fingers, feet, and toes is “ridged” or covered with concentric raised patterns. These ridges are called friction ridges and they serve the useful function of making it easier to grasp and hold onto objects and surfaces without slippage. It is the many differences in the way friction ridges are patterned, broken, and forked which make ridged skin areas, including fingerprints, unique.

Global Versus Local Features:

We make use of two types of fingerprint characteristics for use in identification of individuals: Global Features and Local Features. Global Features are those characteristics that you can see with the naked eye. Global Features include:

a. Basic Ridge Patterns
b. Pattern Area
c. Core Area
d. Delta
e. Type Lines
f. Ridge Count



The Local Features are also known as Minutia Points. They are the tiny, unique characteristics of fingerprint ridges that are used for positive identification. It is possible for two or more individuals to have identical global features but still have different and unique fingerprints because they have local features - minutia points - that are different from those of others.

Fingerprint Scanning:

Fingerprint scanning is the acquisition and recognition of a person's fingerprint characteristics for identification purposes. This allows the recognition of a person through quantifiable physiological characteristics that verify the identity of an individual.

There are basically two different types of finger-scanning technology that make this possible. One is an optical method, which starts with a visual image of a finger. The other was a semiconductor-generated electric field to image a finger.

There are different ways to identify fingerprints. They include traditional police methods of matching minutiae, straight pattern matching, moiré fringe patterns and ultra sonic.

Fingerprint Matching:

Fingerprint matching techniques can be placed into two categories: minutiae-based and correlation based. Minutiae-based techniques first find minutiae points and then map their relative placement on the finger. However, there are some difficulties when using this approach. It is difficult to extract the minutiae points accurately when the fingerprint is of low quality. Also this method does not take into account the global pattern of ridges and furrows. The correlation-based method is able to overcome some of the difficulties of the minutiae-based approach. However, it has some of its own shortcomings. Correlation-based techniques require the precise location of a registration point and are affected by image translation and rotation.

Fingerprint Classification:

Large volumes of fingerprints are collected and stored everyday in a wide range of applications including forensics, access control, and driver license registration. An automatic recognition of people based on fingerprints requires that the input fingerprint be matched with a large number of fingerprints in a database To reduce the search time and computational complexity, it is desirable to classify these fingerprints in an accurate and consistent manner so that the input fingerprint is required to be matched only with a subset of the fingerprints in the database.

Fingerprint classification is a technique to assign a fingerprint into one of the several pre-specified types already established in the literature which can provide an indexing mechanism. Fingerprint classification can be viewed as a coarse level matching of the fingerprints. An input fingerprint is first matched at a coarse level to one of the pre-specified types and then, at a finer level, it is compared to the subset of the database containing that type of fingerprints only. We have developed an algorithm to classify fingerprints into five classes, namely, whorl, right loop, left loop, arch, and tented arch.

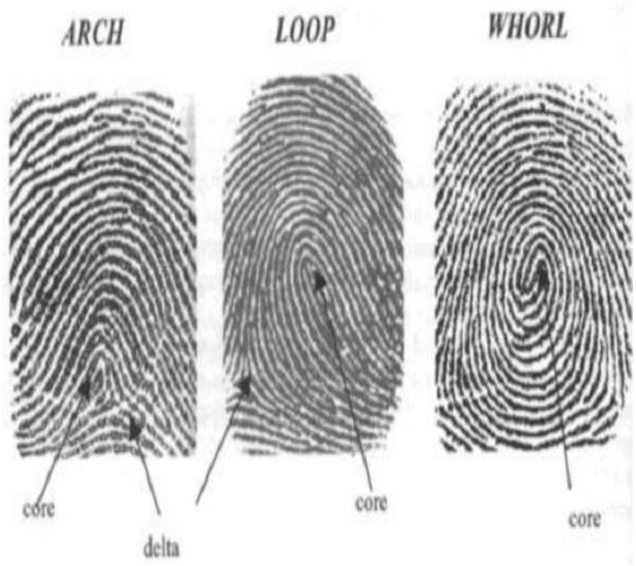


Image Capture:

There are two approaches for capturing the fingerprint image for matching: minutia matching and global pattern matching. Minutia matching is a more microscopic approach that analyzes the features of the fingerprint, such as the location and direction of the ridges, for matching. The only problem with this approach is that it is difficult to extract the minutiae points accurately if the fingerprint is in some way distorted. The more macroscopic approach is global pattern matching where the flow of the ridges is compared at all locations between a pair of fingerprint images; however, this can be affected by the direction that the image is rotated.

Scanners:

a. Optical Scanner - captures a fingerprint image using a light source refracted through a prism
b. Thermal Scanner - very small sensor that produces a larger image of the finger and is contrast-independent
c. Capacitive Scanner - uses light to illuminate a finger placed on a glass surface and records the reflection of this light with a solid-state camera

Each of these devices use light to measure the ridges and non-ridges, take an original fingerprint image, capture the minutia points and create an identifying template from the minutia points.

Image Processing:

Following the image capture, image processing is performed to achieve a definitive match on the individual. At this stage, image features are detected and enhanced for verification against the stored minutia file. Image enhancement is used to reduce any distortion of the fingerprint caused by dirt, cuts, scars, sweat and dry skin.

Image Verification:

At the verification stage, the image of the fingerprint is compared against the authorized user's minutia file to determine a match and grant access to the individual.

Potential Issues:

Although fingerprint recognition is the cheapest form of biometric security available and is widely accepted by public and law enforcement communities as reliable identification, its disadvantage is that it requires close physical contact with scanning device. Some other issues includes

a. Privacy
b. False Rejection
c. False Acceptance
d. Accuracy

a) Privacy: Comparison and storage of unique biological traits makes some individuals feel that their privacy is being invaded. Many associate fingerprint scanning with the fingerprinting of alleged criminals and are therefore hesitant to accept this technology.
b) False Rejection Rate (FRR): False rejection occurs when a registered user does not gain access to the system. This person has then been falsely rejected from access.
c) False Acceptance Rate (FAR): False acceptance is when an unauthorized user gains access to a biometrically protected system.
d) Accuracy: Although fingerprints are unique to an individual, there are instances where a fingerprint may become distorted and authorization will not be granted to the user. As discussed above in image processing, dirt, cuts, scars, sweat and dry skin can cause fingerprint distortion.

Hand Geometry:

Apart from face and fingerprints, hands are another major biometric of human being. Several hand parameters can be used for person identification,

a. hand shape and geometry
b. blood vessel patterns
c. palm line patterns

Hand geometry are consider to achieve medium level of security, it have several advantages [15].

a. Medium cost, only needs a platform and a low/medium resolution CCD camera.
b. It uses low-computational cost algorithms, which lead to fast results.
c. Low template size: from 9 to 25 bytes, this reduces the storage needs.
d. Very easy and attractive to users: leading to a nearly null user rejection.
e. Lack of relation to police, justice, and criminal records.

One of the prototype designs for this biometric system can be seen in Figure 5,

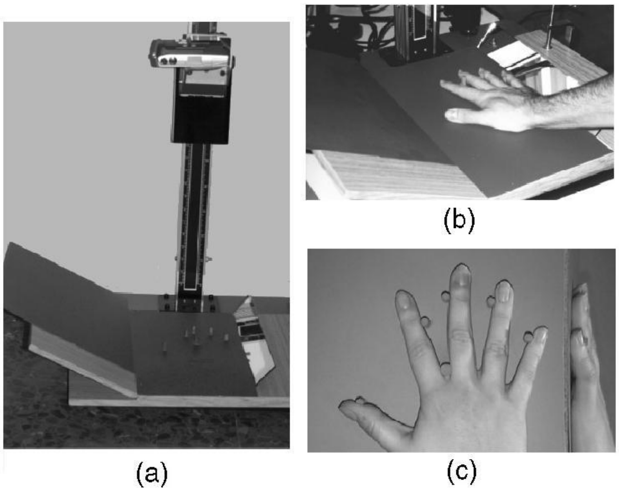


Figure 5. Prototype design a) platform and camera, b) placement of user's hand, c) photograph taken

The image obtained from the CCD camera is a 640 x 480 pixels color photograph in JPEG format [15]. Not only the view of the palm is taken, but also a lateral view is obtained with the side mirror. To extract features, the image is first convert into black and white, and spurious pixels are also removed at this point. Rotation and resizing of image are also done to eliminate variations caused by position of camera. This is follow by Sobel edge detection to extract contours of the hand [15]. The measurements for feature extractions consists of following [15], refer to Figure 6,

Table 2. Measurements for feature extraction.

Widths	Each of the four fingers is measured in different heights, avoiding the pressure points (w11-w44). The width of the palm (w0) is also measured and the interfinger distance at point P1, P2 and P3, vertical and horizontal coordinates.
Heights	The middle finger, the little finger, and the palm (h1, h2, h3).
Deviations	The distance between a middle point of the finger and the straight line, $deviation= P_{12}^{x} - \left(\frac{P_{14}^{x} - P_{1}^{x}}{P_{14}^{y} - P_{1}^{y}}\right)(P_{12}^{y} - P_{1}^{y})$
Angles	Between interfinger point and horizontal.

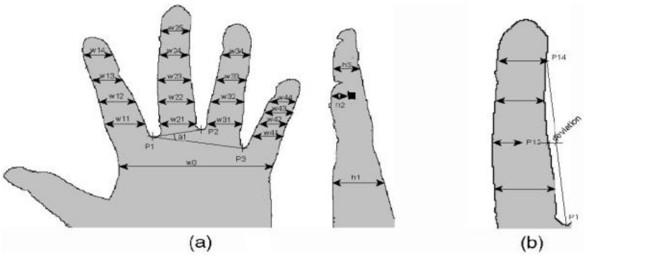


Figure 6. a) Location of measurement point for feature extraction, b) details of the deviation measurement.

In total 31 features are extracted, several classifier algorithm have been discussed in [15]. These are, Euclidean distance, Hamming distance, Gaussian mixture models (GMMs), and Radial basis function neural networks (RBF). The Euclidean distance performs using the following equation,

$$d = \sqrt{\sum_{i=1}^{L} (x_i - t_i)^2}$$

where L is the dimension of the feature vector, x_i is the i^{th} component of the sample feature vector, and t_i is the i^{th}

component of the template feature vector. Hamming distance measure the difference between numbers of components that differ in value. Assume that feature follow a Gaussian distribution, both mean and standard deviation of the samples are obtained, size of the template increase from 25 to 50 bytes.

$$d(x_i,t_i^m) = \# i \{ \in \{1,\ldots,L\} / \left| x_i - t_i^m \right| > t_i^v$$

where t_i^m is the mean for the i^{th} component, and t_i^v the factor of the standard deviation for the i^{th} component. GMMs technique uses an approach between statistical methods and the neural networks [15]. The probability density of a sample belonging to a class *u* is,

$$p(\vec{x}/u) = \sum_{i=1}^{M} \frac{c_i}{(2\pi)^{L/2} \left| \sum_i \right|^{1/2}} \exp\left\{ -\frac{1}{2} (\vec{x} - \vec{u}_i)^T \sum_i^{-1} (\vec{x} - \vec{u}_i) \right\}$$

c_i being the weights of each of the Gaussian models, u_i the mean vector of each model, $\sum_i$ the covariance matrix of each model, *M* the number of models, and *L* the dimension of feature vectors. RBF consists of two layers, one is base on a radial basis function, such as Gaussian distribution the send is a linear layer.

It is found from [15], that GMM give the best result is both classification (about 96 percent success) and verification with a higher computational cost and template size [15], [16]. Performance improves with increasing enrollment size, except Euclidean distance and RBFs. The Equal Error Rate (FAR = FRR), remains similar in each technique for the different feature vector sizes.

Hand Veins:

Not like fingerprint, hand shape and iris/retina biometric systems, hand veins have advantages over contamination issues and will not pose discomfort to the user [17]. The process algorithm consists of image acquisition unit, processing units and recognition module [17], [18]. Under the visible light, vein structure is not always easily seen; it depends on factors such as age, levels of subcutaneous fat, ambient temperature and humidity, physical activity and hand positions, not to mention hairs and scars. [17] proposed using conventional CCD fitted with IR cold source for imaging acquisition. IR emits wavelength of 880 nm ± 25 nm, provide better contrast than ordinary tungsten filament bulbs. Preferably a IR filter is inserted to eliminate any visible light reaches CCD sensor [17]. Below shows a proposed hand vein acquisition device.

The segmentation of the vein pattern consists of procedure several numbers of processes,

Table 3. Thermographic image procedures

Attenuate impulse noise and enhance contrast	Moving average is applied
Determine the domain of the hand	Morphological gradient is used to separate background
Reduce the domain	Morphological openings, closings and erosion are applied
Remove hair, shin pores and other noise	Max and min of independent opening and closing using linear structuring elements are applied
Normalize the background	Brightness surface is subtracted, leaving only the vein structure and background
Threshold out the vein pattern	Morphological gradient is applied to obtain a threshold value that separates the vein and background
Remove artifacts, fill holes	Binary alternating sequential filter, is used to remove threshold artifacts and fill holes in vein structure
Thin the patter down to its medial axis	Modified Zhang and Suen algorithm is used
Prune the medial axis	Automatic pruning algorithm is used

It is noted that significant horizontal positional noise during docking for different registration process [17]. The proposed matching approach compares medial axis and coding algorithm, *constrained sequential correlation*. It is a variation on the traditional correlation methods used for template matching. The reference or library signature is first dilated by a hexagonal structuring element. The test signature is then superimposed on this reference buffer and the percentage of pixels contained within the buffer determined. Due to horizontal translation error, test signature is sequentially translated horizontally and compared against the reference buffer [17]. The horizontal translation is limit to ± 30 pixels. The highest match percentage is said to be the forward similarity, where reverse similarity is obtained by dilate the test signature and reference signature is sequentially correlated until the maximum measure is obtained. By setting the forward and reverse minimum percentage to 75% and 60% respectively, the resultant FRR is 7.5% and FAR is 0%. If forward percentage is lowered to 70%, FRR improved down to 5% and FAR remains the same [17].

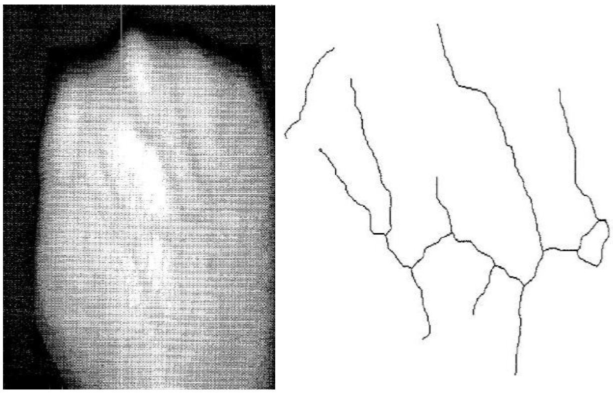


Figure 7. Left is original captured image, right is vein structure after prune the medial axis

Conventional method uses low pass filter follow by high pass filter, after that threshold is applied with bilinear interpolation and modified median filter to obtain hand vein in region of interest (ROI) [18]. The Gaussian low pass filter is a 3 x 3 spatial filter with equation,

$$Z(5)=\sum_{i=1}^{9} W(i)Z(i)$$

$$\textit{Filtered image}=\begin{array}{|c|c|c|}\hline Z_1 & Z_2 & Z_3 \\ \hline Z_4 & Z_5 & Z_6 \\ \hline Z_7 & Z_8 & Z_9 \\ \hline \end{array}\times\begin{array}{|c|c|c|}\hline W_1 & W_2 & W_3 \\ \hline W_4 & W_5 & W_6 \\ \hline W_7 & W_8 & W_9 \\ \hline \end{array}$$

Due to heavy computation load [18] introduce a way of enhancing the algorithm. Both the coefficients for the Gaussian low pass filter and the low pass filter are designed to have 7-tap CSD (canonical signed digit) codes at the maximum. Also, for the normalization, the decimation method is used. It is said that, CSD code is an effective code for designing a FIR filter without a multiplier. The proposed preprocessing algorithm follows the same steps as the conventional method, except that the coefficients for each filter are made of CSD codes. The general CSD code is equal to,

$$W_j=\sum_{i=1}^{M_j} S_i 2^{-i}$$

where j = 1, 2, …, 121, $S_i \in$ {-1, 0, 1} and M is an integer. Instead a 3 x 3 Gaussian low pass filter, a 11 x 11 spatial filter is applied instead. Is it found that Gaussian filter is 94.88% reliable relative to their experiment, and maximum of 0.001% *FAR* can be obtained by varying the threshold level [18].

Iris:

Another biometric non-invasive system is the use of color ring around the pupil on the surface of the eye. Iris contains unique texture and is complex enough to be used as a biometric signature. Compared with other biometric features such as face and fingerprint, iris patterns are more stable and reliable. It is unique to people and stable with age [19]. Figure 8, shows a typical example of an iris and extracted texture image.

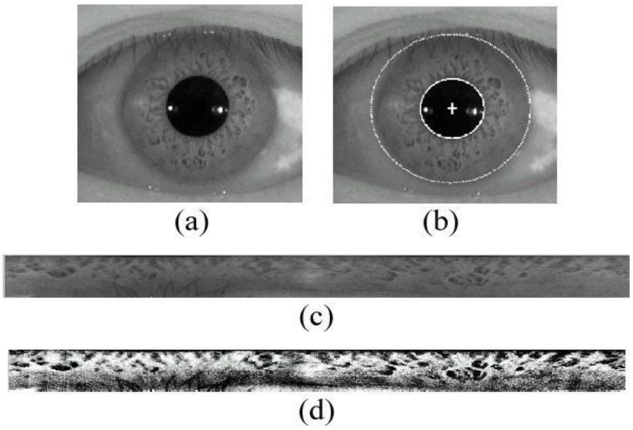


Figure 8. (a) Iris image (b) iris localization (c) unwrapped texture image (d) texture image after enhancement

Iris is highly randomized and its suitability as an exceptionally accurate biometric derives from its [20],

a. extremely data-rich physical structure
b. genetic independence, no two eyes are the same
c. stability over time
d. physical protection by a transparent window (the cornea) that does not inhibit external view ability

There are wide range of extraction and encoding methods, such as, Daugman Method, multi-channel Gabor filtering, Dyadic wavelet transfmor [21], etc. Also, iris code is calculated using circular bands that have been adjusted to conform to the iris and pupil boundaries. Daugman is the first method to describe the extraction and encoding process [22]. The system contains eight circular bands and generates 512-byte iris code, see a)
b)
Figure [20].

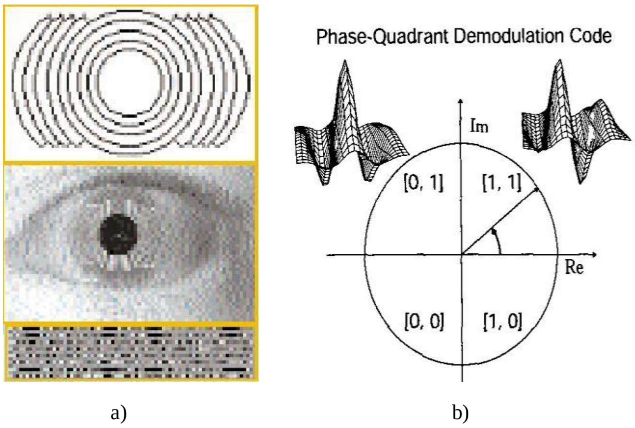


Figure 9. a) Dougman system, top 8 circular band, bottom iris code b) demodulation code

After boundaries have been located, any occluding eyelids detected, and reflections or eyelashes excluded, the isolated iris is mapped to size-invariant coordinates and demodulated to extract its phase information using quadrature 2D Gabor wavelets [22]. A given area of the iris is projected onto complex-valued 2D Gabor wavelet using,

$$h_{\{\mathrm{Re,Im}\}}=\mathrm{sgn}_{\{\mathrm{Re,Im}\}}\int_{\rho}\int_{\phi} I(\rho,\phi)e^{-iw(\theta_0-\phi)}e^{-(r_0-\rho)^2/\alpha^2}e^{-(\theta_0-\phi)^2/\beta^2}\rho\, d\rho\, d\phi$$

where $h_{\{\mathrm{Re,Im}\}}$ can be regarded as a complex-valued bit whose real and imaginary parts are either 1 or 0 (sgn) depending on the sign of the 2D integral. $I(\rho,\phi)$ is the raw iris image in a dimensionless polar coordinate system that is size- and translation-invariant, and which also corrects for pupil dilation. α and β are the multi-scale 2D wavelet size parameters, spanning a 8-fold range from 0.15mm to 1.2mm on the iris, and w is wavelet frequency spanning 3 octaves in inverse proportion to β. (r_0,θ_0) represent the polar coordinates of each region of iris for which the phasor coordinates $h_{\{\mathrm{Re,Im}\}}$ are computed [22]. 2,048 such phase bits (256 bytes) are computed for each iris and equal amount of masking bits are computed to signify any region is obscured by eyelids, eyelash, specular reflections, boundary artifacts or poor signal-to-noise ratio. Hamming distance is used to measure the similarity between any two irises, whose two phase code bit vectors are denoted {*codeA*, *codeB*} and mask bit vectors are {*maskA*, *maskB*} with Boolean operation [22],

$$HD=\frac{\left\|(codeA\otimes codeB)\cap maskA\cap maskB\right\|}{\left\|maskA\cap maskB\right\|}$$

For two identical iris codes, the HD is zero; for two perfectly unmatched iris codes, the HD is 1. For different irises, the average HD is about 0.5 [20]. The observed mean HD was $p = 0.499$ with standard deviation $\sigma = 0.317$, which is close fit to theoretical values [22]. Generally, an HD threshold of 0.32 can reliably differentiate authentic users from impostors [20].

An alternative approach to this iris system can be the use of multi-channel Gabor filtering and wavelet transform [19]. The boundaries can be taken by two circles, usually not co-centric. Compared with the other part of the eye, the pupil is much darker, therefore, inner boundary between the pupil and the iris is determined by means of thresholding. The outer boundary is determined by maximizing changes of the perimeter-normalized sum of gray level values along the circle [19]. Due to size of pupil can be varied, it is normalized to a rectangular block of a fixed size. Local histogram equalization is also performed to reduce the effect of non-uniform illumination, see Figure 8. The multi-channel Gabor filtering technique involves of cortical channels, each cortical channel is modeled by a pair of Gabor filters opposite symmetry to each other.

$$h_e(x,y) = g(x,y) \cdot \cos[2\pi f(x\cos\theta + y\sin\theta)]$$

$$h_o(x,y) = g(x,y) \cdot \sin[2\pi f(x\cos\theta + y\sin\theta)]$$

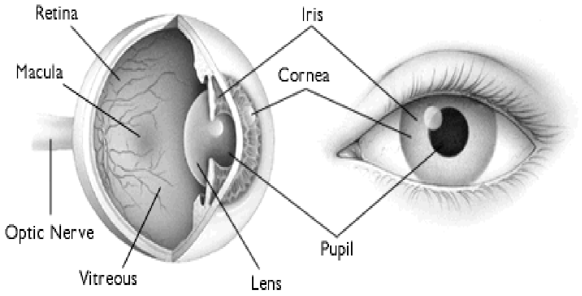
where $g(x,y)$ is a 2D Guassian function, f and θ are the central frequency and orientation. The central frequencies used in [19] are 2, 4, 8, 16, 32 and 64 cycles/degree. For each central frequency f, filtering is performed at $\theta = 0°$, 45°, 90° and 135°. Which produces 24 output images (4 for each frequency), from which the iris features are extracted. These features are the mean and the standard deviation of each output image. Therefore, 48 features per input image are calculated, and all 48 features are used for testing.

A 2D wavelet transform can be treated as two separate 1-D wavelet transforms [19]. A set of sub-images at different resolution level are obtained after applying wavelet transform. The mean and variance of each wavelet sub-image are extracted as texture features. Only five low resolution levels, excluding the coarsest level, are used. This makes the 26 extracted features robust in a noisy environment [19]. Weighted Euclidean Distance is used as classifier,

$$WED(k) = \sum_{i=1}^{N} \frac{(f_i - f_i^{(k)})^2}{(\delta_i^{(k)})^2},$$

where f_i denotes the i^{th} feature of the unknown iris, $f_i^{(k)}$ and $\delta_i^{(k)}$ denots the i^{th} feature and its standard deviation of iris k, N is the total number of features extracted from a single iris. It is found that, a classification rate of 93.8% was obtained when either all the 48 features were used or features at $f = 2$, 4, 8, 16, 32 were used. And the wavelet transform can obtained an accuracy of 82.5% [19]. Other methods such as Circular Symmetric Filters [23] can obtain correct classification rate of 93.2% to 99.85%.

Iris recognition combines computer vision, pattern recognition, statistics, and the human-machine interface. The purpose is real-time, high confidence recognition of a person's identity by mathematical analysis of the random patterns that are visible within the iris of an eye from some distance. Because the iris is a protected internal organ whose random texture is stable throughout life, it can serve as a kind of living passport or a living password that one need not remember but one always carries along. The iris's random patterns are unique to each individual — a human "bar code" or living passport. No two irises are alike. Each person has a distinct pattern of filaments, pits and striations in the colored rings surrounding the pupil of each eye. This pattern is stable throughout life. Unlike fingerprints, iris "prints" are not subject to environmental damage. An internal organ, the iris is protected by the cornea. Because these structures are transparent, the iris can be easily identified with a high degree of certainty up to three feet away.



Retina:

A retina-based biometric involves analyzing the pattern of blood vessels captured by using a low-intensity light source through an optical coupler to scan the unique patterns in the back of the eye [2]. Retina is not directly visible and so a coherent infrared light source is necessary to illuminate the retina. The infrared energy is absorbed faster by blood vessels in the retina than by the surrounding tissue. Retinal scanning can be quite accurate but does require the user to look into a receptacle and focus on a given point. However it is not convenient if wearing glasses or if one concerned about a close contact with the reading device [2]. A most important drawback of the retina scan is its intrusiveness.

The light source must be directed through the cornea of the eye, and operation of the retina scanner is not easy. However, in healthy individuals, the vascular pattern in the retina does not change over the course of an individual's life [24]. Although retina scan is more susceptible to some diseases than the iris scan, but such diseases are relatively rare. Due to its inherent properties of not user-friendly and expansive, it is rarely used today. A typical retinal scanned image is shown in Figure 7.

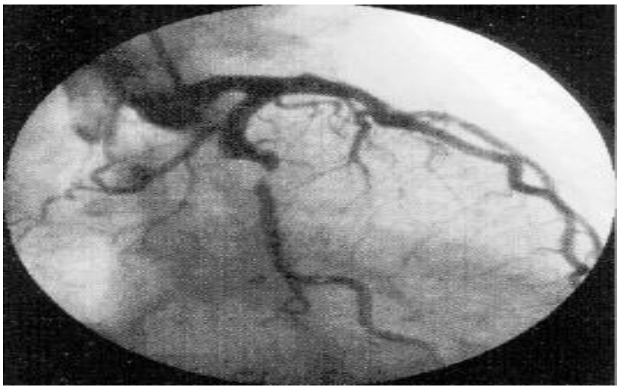


Figure 7. Retinal scanned image

Paper [25] propose a general framework of adaptive local thresholding using a verification-based multithreshold probing scheme. It is assumed that, given a binary image B_T resulting from some threshold T, decision can be made if any region in B_T can be accepted as an object by means of a classification procedure. A pixel with intensity lower than or equal to T is marked as a vessel candidate and all other pixels as background. Vessels are considered to be curvilinear structures in B_T, i.e., lines or curves with some limited width [25]. The approach to vessel detection in B_T consists of three steps: 1) perform an Euclidean distance transform on B_T to obtain a distance map, 2) prune the vessel candidates by distance map retain only center line pixels of curvilinear bands, 3) reconstruct the curvilinear bands from their center line pixels. The reconstructed curvilinear bands give that part of the vessel network that is made visible by the particular threshold T [25].

Fast algorithm for Euclidean distance transform is applied. For each candidate vessel point, the resulting distance map contains the distance to its nearest background pixel and the position of that background pixel [25]. The pruning operation uses two measures, ϕ and d, to quantify the likelihood of a vessel candidate being a center line pixel of a curvilinear band of limited width, see Figure 8.

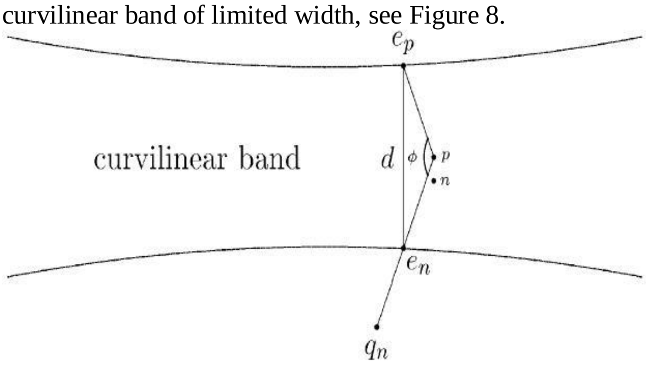


Figure 8. Quantities for testing curvilinear bands.

where p and n represent a vessel candidate and one of the eight neighbors from its neighborhood N_p, respectively, e_p and e_n are their corresponding nearest background pixel. The two measures are defined by,

$$\phi = \max_{n \in N_p} angle(\overline{p}\overline{e}_p, \overline{p}\overline{e}_n) = \max_{n \in N_p} \frac{180}{\pi} \cdot \arccos \frac{\overline{p}\overline{e}_p \cdot \overline{p}\overline{e}_n}{\left\|\overline{p}\overline{e}_p\right\| \cdot \left\|\overline{p}\overline{e}_n\right\|}$$

$$d = \max_{n \in N_p} \left\|\overline{e}_p \overline{e}_n\right\|$$

The overall improvement result can be seen in Figure 9 below,

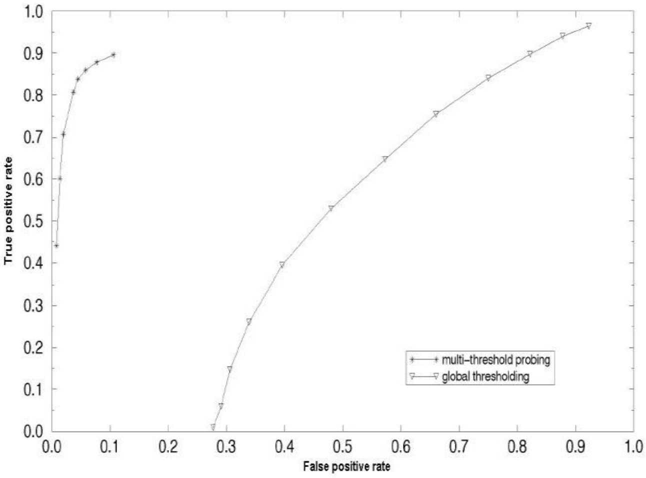


Figure 9. Proposed approach versus global thresholding.

Signature:

Signature differ from above mentioned biometric system, it is a trait that characterize single individual. Signature verification analyzes the way a user signs his or her name. This biometric system can be put into two categories, on-line and off-line methods. On-line methods take consideration of signing features such as speed, velocity, rhythm and pressure are as important as the finished signature's static shape [26]. Where as, off-line classification methods are having signature signed on a sheet and scanned. People are used to signatures as a means of transaction-related identity verification, and most would see nothing unusual in extending this to encompass biometrics. Signature verification devices are reasonably accurate in operation and obviously lend themselves to applications where a signature is an accepted identifier [2]. Various kinds of devices are used to capture the signature dynamics, such as the traditional tablets or special purpose devices. Special pens are able to capture movements in all 3 dimensions. Tablets are used to capture 2D coordinates and the pressure, but it has two significant disadvantages. Usually the resulting digitalized signature looks different from the usual user signature, and sometimes while signing the user does not see what has been written so far. This is a considerable drawback for many (unexperienced) users.

A proposed off-line classification method to compensate the less information is raised by [26]. The proposed method utilizes Hidden Markov Models (HMM) as the classifiers. Before, HMM is applied, scanned signature image have to go through the following,

a. Noise filtering, to remove the noise including noise added by scan process.
b. Correcting the inclination of the sheet in the scanner.
c. Binarization of the graphic.
d. Center the signature image.
e. Skeletonization or thinning algorithm.

Feature extraction is then performed, first try to obtained the starting point (more on the left and more below). Then code the direction using the direction matrix, see Figure 10, the obtained direction vector indicates the direction of the next pixel signature. When come to a crossing point, the straight direction is followed and this point is returned after the straight direction line is fished. The direction vector usually have 300 elements [26].

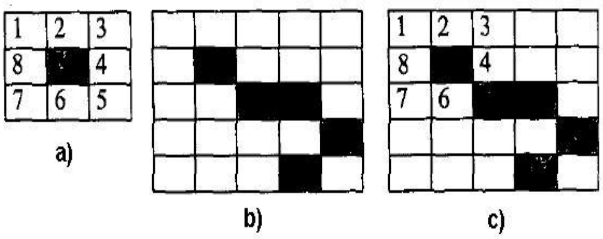


Figure 10. a) Directional matrix b) Signature c) Apply matrix to obtain direction vector = [5 4 5 7]

In recognition stage of an input or signature vector sequence *X*, each HMM model λ_i,
$i = \{1,2,\ldots,M\}$, with *M* equal to the number of different signatures, estimates the “a posteriori” probabilities $P(X \mid \lambda_i)$, and the input sequence *X* is assigned to the *j* signature which provides the maximum score (maxnet),

$$X \in \lambda_i \quad if \quad j = \arg\max_{i=1,2,\ldots,M} P(X \mid \lambda_i)$$

The resultant system decrease greatly when the number of signature increases. The recognition and verification rates are for 30 signatures are 76.6% and 92% respectively.

On-line verification signature verification methods can be further divided into two groups: direct methods (using the raw functions of time) and indirect methods (using parameters) [27]. With direct methods, the signature is stored as a discrete function to be compared to a standard from the same writer, previously computed during an enrolment stage. Such methods simplify data acquisition but comparison can become a hard task. For indirect methods, it requires a lot of effort preparing data to be processed, but the comparison is quite simple and efficient [27]. One direct method system, mentioned in [27], relies on three pseudo-distance measures (shape, motion and writing pressure) derived from coordinate and writing pressure functions through the application of a technique known as Dynamic Time Warping (DTW). It is reported to have over 90% success rate. Another approach is the use of Fast Fourier Transform as an alternative to time warping. It is suggested that working in the frequency domain would eliminate the need to worry about temporal misalignments between the functions to be compared. It is concluded that the FFT can be useful as a method for the selection of features for signature verification [27].

Alternative approach could be wavelet base method, where the signature to be tested is collected from an electronic pad as two functions in time ($x(t)$,$y(t)$). It is numerically processed to generate numbers that represent the distance between it and a reference signature (standard), computed in a previous enrolment stage. The numerical treatment includes resampling to a uniform mesh, correction of elementary distortions between curves (such as spurious displacements and rotations), applying wavelet transforms to produce features and finally nonlinear comparison in time (Dynamic Time Warping).

The decomposition of the functions $x(t)$ and $y(t)$ with wavelet transform generates approximations and details like those showed in Figure 11 to an original example of $x(t)$ [27].

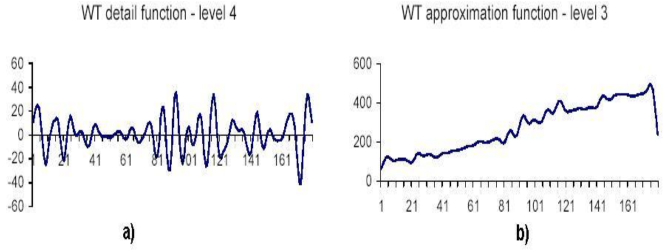


Figure 11. Function $x(t)$ after wavelet transform.

Each zero-crossing of the detail curve at the 4th level of resolution (this level was chosen empirically, by trial and error), three parameters are extracted: its abscissa, the integral between consecutive zero-crossings,

$$vi_k = \int_{ZC_{k-1}}^{ZC_k} WD4(t)dt$$

and the corresponding amplitude to the same abscissa in the approximation function at 3rd level,

$$va_k = WA3(zc_k)$$

As it has been demonstrated that this information suffices to a complete reconstruction of the nontransformed curve [27]. Before measuring distance, it is necessary to identify a suitable correspondence between zerocrossings, which is accomplished with the Dynamic Time Warping (DTW) algorithm. It consists of a linear programming technique, in which the time axis of the reference curve is fixed, while the time axis of the test curve is nonlinearly adjusted, so as to minimize the norm of the global distance between the curves [27]. It is found that the Dynamic Time Warping algorithm on features extracted with the application of wavelet transforms, is suitable to on-line signature verification. Furthermore, it is only with the inclusion of wavelet transform that proposed system can prevent trained forgeries to be accepted (0% FAR).

Multiple Biometric:

In practice, a biometric characteristic that satisfies the requirements mentioned in section image based biometric techniques may not always be feasible for a practical biometric system. In a practical biometric system, there are a number of other issues which should be considered, including [28],

a. Performance, which refers to the achievable identification accuracy, speed, robustness, the resource requirements to achieve the desired identification accuracy and speed, as well as operational or environmental factors that affect the identification accuracy and speed.
b. Acceptability, which indicates the extent to which people are willing to accept a particular biometrics in their daily life.
c. Circumvention, which reflects how easy it is to fool the system by fraudulent methods.

Also, single biometric system has some limitations, such as noisy data, limited degrees of freedom [29]. In searching for a better more reliable and cheaper solution, fusion techniques have been examined by many researches, which also known as multi-modal biometrics. This can address the problem of non-universality due to wider coverage, and provide anti-spoofing measures by making it difficult for intruder to “steal” multiple biometric traits [29]. Commonly used classifier combination schemes such as the product rule, sum rule, min rule, max rule, media rule and the majority rule were derived from a common theoretical framework under different assumptions by using different approximations [30]. In [29] it is discussed that different threshold or weights can be given to different user, to reduce the importance of less reliable biometric traits. It is found by doing this, FRR can be improved. As well it can reduce the failure to enroll problem by assigning smaller weights to those noisy biometrics. Also, in [28], the proposed integration of face and fingerprints overcomes the limitations of both face-recognition systems and fingerprint-verification systems.

The decision-fusion scheme formulated in the system enables performance improvement by integrating multiple cues with different confidence measures, with FRR of 9.8% and FAR of 0.001%. Other fusion techniques have been mentioned in [30], these are Bayes theory, clustering algorithms such as fuzzy K-means, fuzzy vector quantization and median radial basis function. Also vector machines using polynomial kernels and Bayesian classifiers (also used by [31] for multisensor fusion) are said to outperform Fisher's linear discriminant [30]. Not only fusion between biometric, fusions within a same biometric systems using different expert can also improve the overall performance, such as the fusion of multiple experts in face recognition [32] and [33].

Statistical Approaches:

Principal Components Analysis

Part 1: Introduction

A common problem in data processing is that large amounts of data are expensive to *transmit*, *store* or *process*. For transmitting we need high bandwidth, for storing large storage space and for processing we need complex computer systems to reduce the long processing time. To reduce the amount of data, would mean a reduction in expenses. But simply throwing away part of the data would result in a loss of information, which could be important. In so called *random data*, like for example data from images, sounds or other samples, there is however a difference in how important each part of data is to the information which is stored in the data. By leaving out the part of data which is the least valuable to the information, we reach a reduction of the amount of data.

Principal Components Analysis (PCA) is used to compress data in such a way that the least information is lost. It does so by truncating data and thereby leaving out the data which is of the least importance to the information stored in the data. This PCA process is called *dimensionality reduction*, because a vector $\bar{x}$ which contains the original data and is N-dimensional is reduced to a compressed vector $\bar{c}$ which is M-dimensional, where M<N. The question that is answered by PCA is: how can we map vector $\bar{x}$ into a vector $\bar{c}$ with a smaller dimension, but where the information contained in $\bar{x}$ is more or less equal to the information stored in $\bar{c}$?

So, which linear operation should be performed on vector $\bar{x}$ to transform it to vector $\bar{c}$, where information $\bar{x}$ ~ information $\bar{c}$, and dim $\bar{c}$ < dim $\bar{x}$?

We can visualise the above in the following diagram. A vector $\bar{x}$ is coded into a vector $\bar{c}$ with a reduced dimension. Vector $\bar{c}$ is then stored, transmitted or processed, which results in vector $\bar{c}'$, which can be decoded back to a vector $\tilde{\bar{x}}'$. This last vector is an approximation of the result which would have been attained by storing, transmitting or processing vector $\bar{x}$.

$$\xrightarrow{\bar{x}} \boxed{\text{encoder}} \xrightarrow{\bar{c}} \boxed{\text{store / transmit / process } \bar{c}}$$

$$\xrightarrow{\bar{c}'} \boxed{\text{decoder}} \xrightarrow{\tilde{\bar{x}}'}$$

The encoder in this diagram should perform a linear operation, using a matrix $\bar{\bar{Q}}$:

$$\bar{c} = \bar{\bar{Q}}\bar{x}$$

The decoder is also a linear operation, which can be written as a sum of the vector elements of $\bar{c}$ multiplied by the columns of matrix $\bar{\bar{Q}}$: **(I don't think this is correct; it should be x=Qc, instead of ctQt)**

$$\tilde{\bar{x}} = \bar{c}^T \bar{\bar{Q}}^T \rightarrow \tilde{\bar{x}} = \sum_{i=1}^{M} c_i \bar{q}_i$$

How can we find a linear operation so that the difference between $\bar{x}$ and $\tilde{\bar{x}}$ is minimal, but we attain a considerable reduction of $\bar{x}$ when mapped into $\bar{c}$? The solution to this problem is to find a $\bar{c}_{optimal}$ which minimises the error between $\bar{x}$ and $\tilde{\bar{x}}$:

$$\min E\left[\left\|\bar{x} - \tilde{\bar{x}}\right\|^2\right] = \min E\left[\sum_{i=1}^{N}(x_i - \tilde{x}_i)^2\right] = \min E\left[\sum_{i=1}^{N}(x_i - \left(\sum_{j=1}^{N} c_j \bar{q}_j\right)_i)^2\right]$$

In the next paragraph we will proof that the best data to omit from data vector $\bar{x}$ with a given *correlation matrix R*, is the data with the smallest variance. This is the data determined by the smallest *eigenvalues* of the matrix R. Before we provide this proof, we will recall some aspects from linear algebra, which are needed to understand the proof.

Linear Algebra:

We will recall briefly some aspects of linear algebra about Hermitian matrices and correlation matrices.

Hermitian Matrices:

A matrix $\bar{\bar{R}}$ is Hermitian if it is equal to its conjugate transpose. For a real matrix, this means that the matrix is symmetric:

$$\bar{\bar{R}} = \bar{\bar{R}}^T \rightarrow R_{ij} = R_{ji}$$

Furthermore it holds that:

$$\bar{x}^T \bar{\bar{R}} \bar{y} = \bar{y}^T \bar{\bar{R}} \bar{x} \quad \forall \bar{x}, \bar{y}$$

If the Hermitian matrix is non-linear and has dimension n-by-n, then it has n *eigenvalues*. The eigenvalues of $\bar{\bar{R}}$ denoted by λ_i can be found in the calculation of:

$$\bar{\bar{R}} \bar{q}_i = \lambda_i \bar{q}_i$$

In this equation the $\bar{q}_i$ are the N *eigenvectors* of $\bar{\bar{R}}$. These eigenvectors are orthogonal, which means that:

$$\bar{q}_i^{\,T} \bar{q}_j = \delta_{ij} = \begin{cases} 0 & if \ i \neq j \\ 1 & if \ i = j \end{cases}$$

Any Hermitian matrix can be written as the product of its eigenvectors and eigenvalues in the following way: $\overline{\overline{R}} = \overline{\overline{Q}}^T \overline{\overline{\Lambda}}\,\overline{\overline{Q}}$ with $\overline{\overline{Q}}$ = the matrix with the eigenvectors of $\overline{\overline{R}}$ as its columns, and $\overline{\overline{\Lambda}}$ = the empty matrix except for its diagonal, which holds the eigenvalues of $\overline{\overline{R}}$.

A last property of Hermitian matrices is that the sum of its diagonal values equals to the sum of its eigenvalues:

$$\sum_{i=1}^{N} R_{ii} = \sum_{i=1}^{N} \lambda_i$$

Correlation Matrices :

A correlation matrix $\overline{\overline{R}}$ of a random vector $\overline{x}$ is the expectation of the outer product of the vector $\overline{x}$ with itself:

$$\overline{\overline{R}} = E\left[\overline{x}^T \overline{x}\right]$$

The correlation matrix is Hermitian.

Proof of PCA:

With the knowledge from linear algebra, we now can proof that PCA results in the best compression with the minimal loss of information. We use the following considerations in the proof:

The first consideration is:

$$E\left[\overline{x}^T \overline{x}\right] = E\left[\sum_{i=1}^{N} x_i^2\right] = \sum_{i=1}^{N} E\left[x_i^2\right] = \sum_{i=1}^{N} R_{ii} \quad (I)$$

The second consideration, with $c_i = \overline{x}^T \overline{q}_i$:

$$E\left[c_i c_j\right] = E\left[\overline{x}^T \overline{q}_i \overline{q}_j^T \overline{x}\right]$$
$$= E\left[\overline{q}_j^T \overline{x}\overline{x}^T \overline{q}_i\right]$$
$$= \overline{q}_j^T E\left[\overline{x}\overline{x}^T\right] \overline{q}_i$$
$$= \overline{q}_j^T \overline{\overline{R}}\,\overline{q}_i$$
$$= \overline{q}_j^T \lambda_i \overline{q}_i$$
$$= \lambda_i \overline{q}_j^T \overline{q}_i$$
$$= \lambda_i \delta_{ij} \quad (II)$$

The difference between two random vectors **with the same correlation matrix (!?)** now is:

$$E\left[\left\|\overline{x} - \tilde{\overline{x}}\right\|^2\right] = E\left[\left(\overline{x} - \sum_{i=1}^{N} c_i \overline{q}_i\right)^2\right]$$
$$= E\left[\left(\overline{x} - \sum_{i=1}^{N} c_i \overline{q}_i\right)^T \left(\overline{x} - \sum_{i=1}^{N} c_i \overline{q}_i\right)\right]$$
$$= E\left[\overline{x}^T \overline{x}\right] - 2\sum_{i=1}^{N} E\left[c_i \overline{x}^T \overline{q}_i\right] + \sum_{j=1}^{N}\sum_{i=1}^{N} E\left[c_i c_j \overline{q}_i^T \overline{q}_j\right] \xrightarrow{(I)\,\&\,(II)}$$
$$= \sum_{i=1}^{N} R_{ii} - 2\sum_{i=1}^{N} E\left[c_i^2\right] + \sum_{j=1}^{N}\sum_{i=1}^{N} \lambda_i \delta_{ij}$$
$$= \sum_{i=1}^{N} R_{ii} - 2\sum_{i=1}^{N} \lambda_i + \sum_{j=1}^{N}\sum_{i=1}^{N} \lambda_i \delta_{ij}$$
$$= \sum_{i=1}^{N} R_{ii} - 2\sum_{i=1}^{N} \lambda_i + \sum_{i=1}^{N} \lambda_i$$
$$= \sum_{i=1}^{N} R_{ii} - \sum_{i=1}^{N} \lambda_i$$

Since the matrix R is Hermitian this error equals 0, because for such a Hermitian matrix it holds that:

$$\sum_{i=1}^{N} R_{ii} = \sum_{i=1}^{N} \lambda_i$$

Now we have proven that a measure for the difference or *error* between $\overline{x}$ and $\tilde{\overline{x}}$ is given by:

$$E\left[\left\|\overline{x} - \tilde{\overline{x}}\right\|^2\right] = \sum_{i=1}^{N} R_{ii} - \sum_{i=1}^{N} \lambda_i$$

We may rewrite $\overline{x}$ as a combination of eigenvectors:

$$\overline{x} = \sum_{i=1}^{M} c_i \overline{q}_i + \sum_{i=M+1}^{N} c_i \overline{q}_i$$

If we want to compress $\overline{x}$ the next question is, which terms can we best leave out?

The error by leaving out terms is expressed by:

$$\sum_{i=1}^{N} R_{ii} - \sum_{i=1}^{N} \lambda_i = \sum_{i=1}^{N} R_{ii} - \sum_{i=1}^{M} \lambda_i - \sum_{i=M+1}^{N} \lambda_i$$

Clearly, when we use the smallest eigenvalues λ_i for the part we leave out (the terms numbered M+1 until N), we obtain the smallest error.

PCA in six Steps:

Given a random vector $\overline{x}$ of dimension N and its correlation matrix $\overline{\overline{R}}$ we can reduce its dimension to M (with M<N) by Principal Components Analysis in six steps:

a. Find the eigenvectors $\overline{\overline{Q}}$ and eigenvalues λ_i of correlation matrix $\overline{\overline{R}}$:

$$\overline{\overline{R}}\,\overline{q}_i = \lambda_i \overline{q}_i$$

b. Arrange the eigenvalues in decreasing order:

$$\lambda_1 > \lambda_2 > .. > \lambda_M > .. > \lambda_N$$

c. Pick up the eigenvectors which belong to the first M largest eigenvalues.

d. Calculate compressed vector $\bar{c}$ by $c_i = \bar{x}^T \bar{q}_i$ for i = 1, .., M
e. Use vector $\bar{c}$ for storage, transmission, process, etc.
f. Decode the resulting vector $\bar{c}'$ into N-dimensional vector $\tilde{\bar{x}}'$ using the eigenvector matrix $\overline{\overline{Q}}$.

$$\tilde{\bar{x}}' = \sum_{i=1}^{M} c_i \bar{q}_i$$

Part 2: Generating a Correlation Matrix:

To use Principal Components Analysis we need to have a correlation matrix, which defines the similarity between different input vectors. To obtain a correlation matrix $\overline{\overline{R}}$, we construct one by means of observations of different input vectors. We examine for example K different images for constructing matrix $\overline{\overline{R}}$ for a PCA of images. We note $\bar{x}^{(k)}$ as being the k-th observed image.

We use the following empirical approximation of $\overline{\overline{R}}$:

$$\tilde{R}_{ij} = \frac{1}{K} \sum_{k=1}^{K} x_i^{(k)} x_j^{(k)}$$

The more observations are made, the better the approximation $\tilde{\overline{\overline{R}}}$ of $\overline{\overline{R}}$ gets. Instead of matrix $\overline{\overline{R}}$ we use matrix $\tilde{\overline{\overline{R}}}$ in the PCA calculations.

Part 3: Kernel Method:

To determine the eigenvectors of correlation matrix $\overline{\overline{R}}$, we have to construct the matrix $\overline{\overline{R}}$ by calculating the outer product of vector $\bar{x}$. In most applications of PCA, this vector $\bar{x}$ is very large, as it represents the data which is to be compressed. The complexity of the calculations are high, namely $O(N^3)$. There is a way in which we can reduce this complexity to $O(K^3)$, where K is much smaller than N, when we use the limited number of observations of vectors $\bar{x}$ to construct the needed eigenvectors for PCA.

Remember that:

Since $\tilde{\overline{\overline{R}}}$ is constructed from different vectors $\bar{x}^{(k)}$, the eigenvalues of $\tilde{\overline{\overline{R}}}$ are in the space which is spanned by $\bar{x}^{(k)}$:

$$\tilde{\bar{q}}_i \in span\{\bar{x}^{(k)}, k = 1,.., K\}$$

Therefore:

$$\tilde{\bar{q}}_i = \sum_{k=1}^{K} \alpha_i^{(k)} \bar{x}^{(k)}$$

We here provide the reduction in complexity. This the so called PCA Kernel method, because it uses the Kernel matrix U, which elements are given by:

$$U_{ij} = \bar{x}^{(i)T} \bar{x}^{(j)}$$

$$\tilde{\overline{\overline{R}}} \tilde{\bar{q}}_i = \tilde{\lambda}_i \tilde{\bar{q}}_i \rightarrow$$

$$\tilde{\overline{\overline{R}}} \sum_{k=1}^{K} \alpha_i^{(k)} \bar{x}^{(k)} = \sum_{k=1}^{K} \tilde{\lambda}_i \alpha_i^{(k)} \bar{x}^{(k)} \rightarrow$$

$$\frac{1}{K} \sum_{j=1}^{K} \bar{x}_j^{(k)} \bar{x}_j^{(k)T} \sum_{k=1}^{K} \alpha_i^{(k)} \bar{x}^{(k)} = \sum_{k=1}^{K} \tilde{\lambda}_i \alpha_i^{(k)} \bar{x}^{(k)} \rightarrow$$

$$\sum_{k=1}^{K} \alpha_i^{(k)} \sum_{j=1}^{K} \bar{x}_j^{(k)} \bar{x}_j^{(k)T} \bar{x}^{(k)} = K \sum_{k=1}^{K} \tilde{\lambda}_i \alpha_i^{(k)} \bar{x}^{(k)} \rightarrow$$

$$\sum_{k=1}^{K} \alpha_i^{(k)} \sum_{j=1}^{K} \bar{x}_i^{(k)T} \bar{x}_j^{(k)} \bar{x}_j^{(k)T} \bar{x}^{(k)} = K \sum_{k=1}^{K} \tilde{\lambda}_i \alpha_i^{(k)} \bar{x}_i^{(k)T} \bar{x}^{(k)} \rightarrow$$

$$\sum_{k=1}^{K} \alpha_i^{(k)} \sum_{j=1}^{K} U_{ij} U_{jk} = K \tilde{\lambda}_i \sum_{k=1}^{K} \alpha_i^{(k)} U_{ik} \rightarrow$$

$$\overline{\overline{U}}^2 \bar{\alpha}_i = K \tilde{\lambda}_i \overline{\overline{U}} \bar{\alpha}_i \rightarrow$$

$$\overline{\overline{U}} \bar{\alpha}_i = K \tilde{\lambda}_i \bar{\alpha}_i$$

This last equation can be recognised as the eigenvalue problem. The $\bar{\alpha}_i$ vectors are in fact the eigenvectors of this problem. We can find them by solving the equation.

Now given that $\tilde{\bar{q}}_i = \sum_{k=1}^{K} \alpha_i^{(k)} \bar{x}^{(k)}$ we can determine the principal components:

$$c_i = \bar{x}^T \tilde{\bar{q}}_i$$

$$= \bar{x}^T \sum_{k=1}^{K} \alpha_i^{(k)} \bar{x}^{(k)}$$

$$= \sum_{k=1}^{K} \alpha_i^{(k)} \bar{x}^T \bar{x}^{(k)}$$

The construction of these principal components in this manner, can be thought of as a neural network. The network has K input neurons, each with N input weights. The output of the network is vector c, which is the compressed version of input vector x.

To decode vector c back into x, we perform the following operation:

$$\tilde{\bar{x}} = \sum_{i=1}^{M} c_i \tilde{\bar{q}}_i$$

The advantage of the Kernel method is that a great reduction in computational complexity is achieved.

Part 4: Hebbian Learning:

A second way to find out the needed eigenvectors used for PCA, is to use another type of neural network. This is the so called Hebbian-based PCA. It is a feedforward, single layer network with N inputs, where N is the dimension of vector x, and M outputs, where M is the dimension of compressed vector c.

(picture of Hebbian neural network)

To learn the network, different vectors x are presented as an input vector. For the k-th vector x, denoted by x(k), we compute the following calculations, for m=1,2,...,M and i=1,2,...,N:

$$c_m(k) = \sum_{i=1}^{N} w_{mi}(k) x_i(k)$$

$$\Delta w_{mi}(k) = \eta\left[c_m(k)x_i(k) - c_m(k)\sum_{j=1}^{m} w_{ji}(k)c_j(k)\right]$$

Here at time k, $c_m(k)$ is the m-th element of vector c, $w_{mi}(k)$ is the weight of the connection between neuron m and the i-th element of input vector x and $\Delta w_{mi}(k)$ denotes the changing of the weights due to network learning. The learning-rate parameter η is used to set the speed at which the network is learning. A higher rate of learning causes however a larger inaccuracy in the final result. The result we are interested in are the weights w_{mi}. We will not give the proof here, but Kuhn-Tucker proved that using this Hebbian-based neural network, the weights of the network converge to the PCA eigenvectors of the correlation matrix belonging to vector x.

$$\lim_{k\to\infty} \overline{w}_i(k) = \overline{q}_i$$

Thus instead of calculating the complete correlation matrix R, we only determine its first M eigenvectors, which in turn can be used to do a PCA. We use the formula above to determine the compressed vector c for a given data vector x, and we can reconstruct vector x afterwards by calculating:

$$\tilde{\overline{x}} = \sum_{i=1}^{M} c_i \tilde{\overline{q}}_i = \sum_{i=1}^{M} c_i \overline{w}_i$$

Clearly by avoiding the calculation of the complete correlation matrix R, we can gain enormous computational savings, when only the first M eigenvectors are to be determined, where M is substantially smaller than N.

Part 5: The Difference Between Kernel Method and Hebbian Learning:

(draft version)

The next question is: which of the two methods is better in determining the eigenvectors used for PCA? And how complex are the calculations?

Part 6: Conclusion:

(Draft Version)

We have seen how we can use PCA to compress large amounts of data into smaller parts, in such a way that a minimum of information is lost.

Fisher face:

Fisherface was suggested by Peter N. Belhumeur, Joao P. Hespanha and David J. Kriegman of Yale Univeristy in 1997. This approach is similar to eigenface approach, which makes use of projection of image into a face space, with improvements on insensitive to large variation in lighting and facial expression.

Eigenface method uses PCA for dimensionality reduction, which yields projection directions that maximize the total scatter across all classes of images. This projection is best for reconstruction of images from a low dimensional basis. However, this method doesn't make use of between-class scatter. The projection may not be optimal from discrimination for different classes. Let the total scatter matrix S_T is defined as

$$S_T = \sum_{k=1}^{N}\left(T_k - \psi\right)\left(T_k - \psi\right)^T$$

The projection W_{opt} is chosen to maximize the determinant of the total scatter matrix of the projection sample, i.e.

$$W_{opt} = \arg\max_W \left|W^T S_T W\right|$$

$= [w_1, w_2, \ldots\ldots, w_m]$

where $\{w_i|\ i=1,2\ldots\ldots,m\}$ is the set of n–dimensional eigenvectors of S_T corresponding to the m largest eigenvalues.

Fisherface method uses Fisher's Linear Discriminant (FLD) by R.A. Fisher. This projection maximizes the ratio of between-class scatter to that of within-class scatter. The idea is that it tries to "shape" the scatter in order to make it more reliable for classification. Let the between-class scatter matrix be defined as

$$S_B = \sum_{l=1}^{C} N_l(\psi_l - \psi)(\psi_l - \psi)^T$$

and the within-class scatter matrix be defined as

$$S_W = \sum_{l=1}^{C} \sum_{T_k \in T} (T_l - \psi_l)(T_l - \psi_l)^T$$

where ψ_i is the mean image of class $T_{i.}$ The optimal projection W_{opt} is chosen as the matrix with orthonormal columns, which maximizes the ratio of the determinant of the between-class scatter matrix of the projected samples to the determinant of the within-class scatter matrix of the projected samples, i.e.

$$W_{opt} = \arg\max_W \frac{\left|W^T S_B W\right|}{\left|W^T S_W W\right|}$$

$= [w_1, w_2, \ldots\ldots, w_m]$

Besides, this method projects away variation in lighting and facial expression while maintaining discriminability. For lighting variation, the variation due to lighting is reduced by discarding the three most significant principal components. This is because the first three principal components contribute the lighting variations. This results in better performance under variable lighting conditions. For facial expression variation, we can divided the training images into classes based on the facial expression. Take glasses recognition as an example, the training set can be divided into two main classes: "wearing glasses" and "not wearing glasses". With this set of training data, Fisherface can correctly recognized people even he is wearing glasses. Therefore, Fisherface works well with variation in lighting and facial expression.

Here we present some advantages and disadvantages:

Advantages:

Robust against noise and occlusion, Robust against illumination, scaling,orientation and translation when face is correctly normalized, Robust against facial expressions, glasses, facial hair, makeup etc., Can handle high resolution images efficiently, Can handle very low resolution images, Fast recognition/Low computational cost.

Disadvantages:

Removes neighbourhood relationships between pixels, Sensitive to faulty normalization,Sensitive to perspective, viewing angle and head rotation (can be improved using fisher light-fields), Does not handle small training sets well, Slow training/High computational cost (with large databases).

SMPCA:

solves all the roots of the characteristic polynomial equation derived from the sample covariance matrix, referred to as eigenvalues and find their corresponding eigenvectors which will be used to generate all principal components simultaneously.

PGPCA:

Does not solve all the roots of the characteristic equations as in SMPCA. It only locates the maximum Eigenvalue for each of the sample covariance matrices formed by a sequence of reduced subspaces so that principal components can be generated one at a time progressively.

SCPCA:

Does not solve characteristic polynomial equation. This technique uses a learning algorithm to generate the PCs. A random initial vector is used to produce a projection vector and the same process is repeated to generate successive PCs.

PRPCA:

Uses a custom designed initialization algorithm to appropriate set of initial projection vectors for the PCA. The PCs are prioritized by the projection vectors. An example of such an PCA. The new version divides the image into equal size sub images on which PCA is applied resulting in the formation of individual eigenfaces for each sub image [Chein – I Chang and Wei-min Liu(2007)]. Even though PCA produces good results, it is computationally very complex with increase in database size. A new PCA based algorithm using geometry and symmetry of the faces which extracts features using fast Fuzzy.

LDA (Linear Discriminant Analysis) and Evaluation:

The objective of LDA is to find the subspace that best discriminates different face classes by maximizing between class scatter, while minimizing the within-class scatter. The eigenvectors chosen by LDA provide the best separation among the class distributions, while PCA selects eigenvectors which provide best representation of the overall sample distribution. To illustrate the difference, Fig. 2.3 shows the first projection vector chosen by PCA and LDA for a two class problem. The eigenvectors for LDA can be obtained by computing the eigenvectors of $S_w^{-1}S_b$.

Here, S_b and S_W are the between-class and with in-class scatter matrices of training samples and are defined as:

$$S_W = \sum_{i=1}^{C} \sum_{x_k \in C_i} (x_k - m_i)(x_k - m_i)^T$$

$$S_b = \sum_{i=1}^{C} n_i (m_i - m)(m_i - m)^T$$

where m_i is the mean face for i^{th} class and n_i is the number of training samples in i^{th} class. LDA subspace is spanned by a set of vectors W , which maximizes the criterion, J, defined as:

$$j = \frac{tr(S_b)}{tr(S_W)}$$

W can be constructed by the eigenvectors of $S_w^{-1}S_b$. In most of the image processing applications, the number of training samples is usually less than the dimension of the sample space. This leads to the so-called small-sample-size (SSS) problem due to the singularity of the within-class scatter matrix.

a. DCV (Discriminative Common Vectors Approach): DCV “small sample size problem" of LDA by optimizing a variant of Fisher's criterion. It searches for the optimal projection vectors in the null space of within class scatter S_w, satisfying the criterion,

So, to find the optimal projection vectors in the null space of S_w, it projects the face samples onto the null space of S_w to generate common vectors for each class and then obtain the projection vectors by performing PCA on common vectors. A new set of vectors, called as discriminative common vectors, are obtained by projecting face samples on the projection vectors. Thus each class is represented by a single discriminative common vector. Among two algorithms to extract the discriminant common vectors for representing each person in the training set of face database, one algorithm uses within-class scatter matrix of the samples in the training set while the other uses the subspace methods and the Gram-Schmidt orthogonalization procedures to obtain the discriminative common vectors. These discriminative common vectors are used for classification of new faces.

b. Probabilistic Eigenspace Method: Probabilistic subspace method models intra-personal and extra-personal variations to classify the face intensity difference Δ as intra-personal variation (Ω_I) for the same class and extra-personal variation (Ω_E) for different classes.

c. 2D-PCA (Two Dimensional Principal Component Analysis): The 2DPCA technique (Yang, Zhang et al. [2004]) is based on 2D image matrices instead of 1D vectors. The image covariance matrix is constructed directly from the original image.

d. ***Hidden Markov Models (HMM):*** The use of hidden Markov models is a powerful statistical technique that has been applied to many subject areas, from predicting political crises to the reconstruction of DNA and the recognition of speech. The September 1964 issue of Scientific American illustrated a Markov chain by showing two containers with numbered balls in them. Numbered slips of paper associated with the balls were drawn repeatedly, with replacement, from a hat. The ball associated with the number drawn was transferred to the other container than the one it was in. Initially all the balls were in the first container, and gradually this declined exponentially until it contained only half of the balls. This modelled the physical process of allowing two separate chambers, containing a gas at different levels of pressure to be connected. One basic feature with the Markov process is that it involves probability. In addition to a random event the final result also depends on some kind of system memory, described by the number of balls in the first container.

A hidden Markov model consists of two interrelated processes. First an underlying, unobservable Markov chain with a finite number of states (N), a state transition probability matrix (A) and an initial state probability distribution (?). Transition probability is the probability that the system will change its state from one turn to the next. Second a set of probability density functions (B) associated with each state.

Eigenface:

Eigenface was suggested by Alex. P. Pentland and Matthew A. Turk of MIT in 1991. The main idea of eigenface is to get the features in mathematical sense instead of physical face feature by using mathematical transform for recognition.

There are two phases for face recognition using eigenfaces. The first phase is the training phase. In this phase, a large group of individual faces is acted as the training set. These training images should be a good representation of all the faces that one might encounter. The size, orientation and light intensity should be standardized. For example, all images are of size 128 x 128 pixels and all are frontal faces. Each of the images in the training set is represented by a vector of size *N* by *N*, with *N* representing the size of the image. With the training images, a set of eigen-vectors is found by using Principal Component Analysis (PCA).

The basic idea of PCA is to take advantages of the redundancy existing in the training set for representing the set in a more compact way. Using PCA, we can represent an image using *M* eigenvectors where *M* is the number of eigenvector used. ($M << N^2$). As *M* is much smaller than N^2, comparison between vectors would be efficient.

PCA is done by first finding the average face ψ by averaging the training set images $\{T_1, T_2, \ldots\ldots T_M\}$ with T_i representing each of the vector in the set. Then we form a matrix A = $\{\varphi_1, \varphi_2, \ldots\ldots\varphi_M\}$ with column vector $\varphi_i = T_i - \psi$, which is the difference vector of the train images and the average face. We can then get the covariance matrix $C = AA^T$ and the eigenvector and the associated eigenvalues of *C*.

After the eigenvectors have been calculated, the eigenvalues of each eigenvector are sorted. These vectors are known as eigenfaces. The eigenfaces with the largest number of eigenvalues are chosen. These *M'* (where *M'<M*) eigenfaces are considered the best eigenvector to represent a face. The span of the *M'* eigenfaces are called face space. Figure 2 below shown a few of low order eigenfaces used for projection.

Second phase of this algorithm is recognition phase. In this phase, a new image is obtained. To recognize this image, we first subtracted the image by the average face ψ. Then we calculate the dot product of the input vectors with the eigenfaces. This makes a projection of the input image onto the face space. Similarly, we make projections of the training image onto the face space. Figure 3 shows the projection of image onto the face space, which appears as the point in the plane. The euclidean distances of point of the input image with the points of training set are then computed. The training set image with minimum distance from the input image should be the best match. However, there maybe cases that the input image is not in the training set. This would still find a best match of the input image, but this best match is not the correct one. Therefore, we can set a distance threshold for the recognition by trail and error until a satisfactory one is found. When the minimum distance found is larger than the threshold, we can regard the input image is not in the training set.

In the experiment the effects of varying lighting, size and head orientation were investigated using a database of 2500 images. Experiment result shows that eigenface approach reach 96% correct classification averaged over lighting variation, 85% correct averaged over orientation variation and 64% correct averaged over size variation.

Elastic Bunch Graph Matching:

Elastic Bunch Graph Matching was suggested by Laurenz Wiskott, Jean-Marc Fellous, Norbert Kruger and Christoph von der Malsburg of University of Southern California in 1999. This approach takes into account the human facial features and is totally different to Eigenface and Fisherface. It uses elastic bunch graph to automatically locate the fiducial points on the face (eyes, nose, mouth etc) and recognize the face according to these face features.

The representation of facial feature is based on Gabor wavelet transform. Gabor wavelets are biologically motivated convolution kernels in the shape of plane waves restricted by a Gaussian envelope function. We use the Gabor wavelet because it can extract the human face feature well. The family of Gabor kernels

$$\varphi j(\vec{x}) = \frac{k_j^{\,2}}{\sigma^2}\exp\left(-\frac{k_j^{\,2}x^2}{2\sigma^2}\right)\left[\exp(i\vec{k}_j\vec{x}) - \exp\left(\frac{-\sigma^2}{2}\right)\right]$$

in the shape of plane waves with wave vector $\vec{k}_j$, restricted by a Guassian envelope function. We employ a discrete set of 5 different frequencies, index *v = 0, 1,…,7* and 8 orientations, index $\mu = 0, 1,\ldots,7$

$$\vec{k}_j = \begin{pmatrix} k_{jx} \\ k_{jy} \end{pmatrix} = \begin{pmatrix} k_v\cos\varphi_\mu \\ k_v\cos\varphi_\mu \end{pmatrix}, k_v = 2^{-\frac{v+2}{2}}\pi, \varphi_\mu = \mu\frac{\pi}{8}$$

with index $j = \mu + 8v$, and $\sigma = 2\pi$.

Gabor wavelet transformation is done by convolution of the image with the 40 Gabor filters shown in figure 5 above. A jet describes a small patch of gray values in an image $T(\vec{x})$ around a given pixel $\vec{x} = (x,y)$. A jet *J* is defined as the set {*Ji*} of 40 complex coefficients obtained for one image point. It can be written as

$$J_i = a_j\exp(i\phi j)$$

with magnitudes $a_{j(\vec{x})}$, which slowly vary with position, and phase $\phi_{j(\vec{x})}$, which rotate at a rate approximately determined by the spatial frequency or wave vector $\vec{k}_j$ of the kernels. Figure 6 below shows a convolution is made between the original image and the Gabor wavelets. The set of 40 coefficients obtained for one image point is referred as a jet. A collection of this jets, together with the relative location of the jets form an image graph in the right.

$$S_a(J, J') = \frac{\sum_j a_j a'_j}{\sqrt{\sum_j a_j{}^2 \sum_j a'_j{}^2}}$$

However, jets taken from image points only a few pixels apart from each other have very different coefficients due to phase rotation. This may decrease the accuracy of matching. Therefore, we have another method to compare the jets. This method takes into account the phase difference in comparison, the phase similarity function

$$S_a(J, J') = \frac{\sum_j a_j a'_j \cos(\phi_j - \phi'_j - \vec{d}\vec{k}_j)}{\sqrt{\sum_j a_j{}^2 \sum_j a'_j{}^2}},$$

Using this phase function, the phase difference ($\phi_j - \phi'_j$) is compensated by the displacement $\vec{d}$, which is estimated using Taylor expansion. The displacement estimation could be done using the disparity estimation.

To represent a face, we need to build an image graph from a set of fiducial points like the pupils, the corner of the mouth, the tip of the nose, the top and bottom of ears, etc. A labeled graph *G* representing a face consists of *N* nodes on the fiducial points at position $\vec{x}_n$, *n = 1, ...,N* and *E* edges between them. An image graph is shown in right side of Figure 6, which looks like a grid. For this image graph, 9 fiducial points are used as nodes.

For an automatic face recognition system, it has to locate the fiducial point and build the image graph from an input image automatically. This can be done by matching the input image with a stack like general representation of faces, Face Bunch Graph (FBG). A FBG consists of bunches, which are sets of jets of wide range variation of appearance of a face. Figure 8 shows a face bunch graph. There are set of jets in a node (a bunch) to represent a fiducial point, each with different variations. For example, the eye bunch may consist of jets of open eye, closed eye, male and female eye. With the variations, people with different facial expression could be matched accordingly.

In order to accurately and efficiently locate the fiducial points of an image, two types of FBG are used at two different stages. At normalization stage, a face position is found from an image, a FBG of 30 different models are used. At graph extraction stage, fiducial points are accurately found to build an image graph of the image. This requires FBG of larger size including 70 different models to match accurately.

For the matching between an input graph and the FBG, a function called graph similarity is employed. This function depends on the jet similarity mentioned before and the distortion of the image grid relative to the FBG grid. For an image graph g^I with nodes *n = 1,…,N* and edges *e = 1,…,E* and an FBG B with model graphs *m = 1,…,M* . The similarity is defined as

$$S_B(g^I, B) = \frac{1}{N}\sum_n \max_m(S_\phi(J_n^I, J_n^{B_m})) - \frac{\lambda}{E}\sum_e \frac{(\Delta\vec{x}_e^I - \Delta\vec{x}_e^B)^2}{(\Delta\vec{x}_e^B)^2}$$

where λ determines the relative importance of jets and metric structure. J_n are the jets at nodes *n*, and $\Delta\vec{x}_e$ are the distance vectors used as labels at edge *e*.

In order to extract the image graph from an image, two main steps of matching are needed. The first step is to find the location of a face from the image by using the smaller size FBG. This step is further divided into 3 sub-steps. The first one is to find the approximate face position. The second one is to refine the position and size of the grid found. The last sub-step is to further refine the size of the grid and find the aspect ratio of the face, i.e. the grid. We could then accurately locate the position of a face in the image after applying these steps. After that, step two is performed to find the local distortion of the grid. This helps us finding the fiducial points inside the grid accurately with the use of larger size FBG.

To recognize a image, we simply compare the image graph to all modal graph and pick the one with the highest similarity value. The similarity function is an average over the similarities between pairs of corresponding jets. If g^I is the image graph, g^M is the modal graph, and node $n_{n'}$ is the modal graph corresponds to node *n'* in the image graph, the define graph similarity is

$$S_g(gI, gM) = \frac{1}{N'}\sum_{n'} S_a(J^I_{n'}, J^M_{n_{n'}})$$

where the sum runs only over the *N'* nodes in the image graph with a corresponding node in the modal graph.

NEURAL NETWORK MODELS FOR TREE CONSTRUCTION

The aim in multivariate decision tree construction is to find a way to determine the coefficients of the attributes. In this chapter, we will discuss how to determine these set of coefficients by using methods based on neural networks.

When neural networks are used in decision trees, at each node of the decision tree, there is a neural network trained with its corresponding data. Once the weights of the neural networks are found, they can be used to classify a test example.

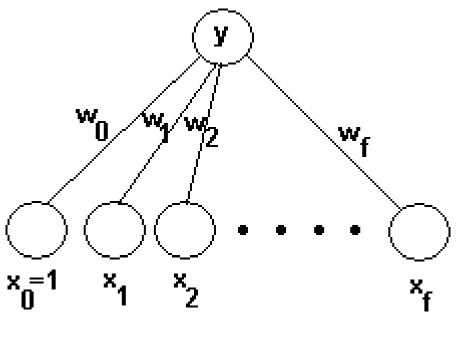


Figure15 Linear perceptron model for multivariate decision trees

A model for a linear perceptron is shown in Figure 0.1 (Bishop, 1996). $x_0, x_1, x_2 \ldots x_f$, form the input layer of the neural network whereas the node on top is the output node of the neural network denoted by y. The output of the neural network is binary. w_i's are the weights of the input neurons. The weight of $x_0 = 1.0$ is w_0, which will be used as a threshold unit (Corresponding to c of CART). y is computed as follows:

$$y = sigmoid(\sum_{i=0}^{f} w_i x_i) = \frac{1}{1+e^{-\sum_{i=0}^{f} w_i x_i}}$$

The sigmoid function gives a value between 0 and 1.

In order to train the linear perceptron, gradient-descent is used. This algorithm is used to train a single-output linear perceptron to differentiate between two disjoint groups of classes, which are the classes in the left branch C_L and the classes in the right branch C_R. The desired output for an instance is 1 when its class belongs to the group C_L and 0 when its class belongs to the group C_R. If there are only two classes present in that node, one class belongs to C_L and other to C_R. Otherwise we must select an appropriate partition for the classes available.

Finding the best split with neural network algorithms is done as a nested optimization problem. In the inner optimization problem, the gradient-descent algorithm is used to minimize the mean-square error and so find a good split as defined by w_i for the given two distinct groups of classes C_L and C_R. In the outer optimization problem, we find the best split of classes into the two groups, C_L and C_R.

Training Neural Networks:

The inner optimization problem will be solved by using neural networks at each node of the decision tree. For this purpose we have used three different kinds of neural network models. These are linear perceptrons, multilayer perceptrons and a hybrid combination.

Linear Perceptron Model:

A linear perceptron neural network model is shown in Figure above. The training of this network is done by gradient-descent algorithm. Let (x^t, d^t) denote the training data, where x^t is the instance t and d^t is the desired output for that instance, which is 1 for left child and 0 for right child. y^t is the real output found by the formula:

$$y^t = sigmoid(\sum_{i=1}^{f} w_i x_i^t + w_0)$$

A stochastic gradient algorithm for minimizing the mean-square error criterion is used.

$$E = \frac{1}{n}\sum_{t=1}^{n}(d^t - y^t)^2$$

At each epoch of training, all instances are passed one at a time in random order. While passing the coefficients of the input layer, w_i and threshold unit w_0 are updated by the given formulas. In these formulae η stands for learning rate. Learning rate is started as 0.3 / f and decreased to 10^{-5} by multiplying at each epoch with a constant. α stands for momentum rate (0.7 in our application). It is used to update w_i by multiplying with the previous update value as:

$\Delta w_i^t = \eta(d^t - y^t) y^t (1- y^t) x_i^t + \alpha \Delta w_i^{t-1}$, $i=0,\ldots,n$, $x_0=+1$
(0.4)

$w_i^{t+1} = w_i^t + \Delta w_i^t$

The training of linear perceptron takes O($e * n * (f+1)$) where e is the number of epochs to train the network, n is the number of instances and $(f+1)$ is the number of inputs (+threshold), which is the number of weights to update.

Multilayer Perceptron Model:

A multilayer perceptron model for decision making at a node is shown in Figure 0.1.

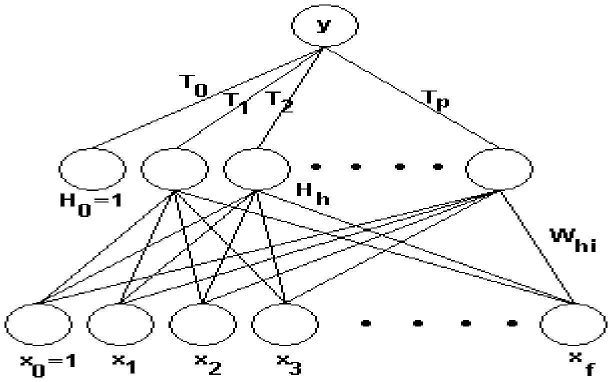


figure 16Multilayer perceptron model with one hidden layer

There is a hidden layer between input and output layers which makes y a nonlinear function of x. $x_0, x_1, x_2 \ldots x_f$, form the input layer, $H_0, H_1, \ldots H_m$ form the hidden layer and y denotes the output of the neural network. For the weights connecting the layers, w_{hi} connects input neuron i to hidden neuron h and T_h connects hidden neuron h to the output layer. x_0 and h_0 denote the bias units for the input and hidden layers respectively. The number of units in the hidden layer is taken as half of the number of features, for suitable dimensionality reduction from f to 1.

At instance t the real output y^t and the hidden layer output H_h^t are found as:

$$H_h^t = sigmoid(\sum_{i=0}^{f} w_{hi} x_i^t + w_{h0}) \qquad (0.6)$$

$$y^t = sigmoid(\sum_{h=0}^{m} T_h H_h^t + H_0) \qquad (0.7)$$

The update rules are different from single layer perceptron. There are two layers so there are two update rules, one for updating the w_{hi} and the other for updating T_h. Learning rate is started as 0.3 / f and decreased as explained in linear perceptron model. The update rules are:

$\Delta T_h^t = \eta (d^t - y^t) y^t (1- y^t)H_h^t + \alpha \Delta T_h^{t-1}$, $h=0,\ldots,m$, $H_0=+1$
(0.8)

$T_h^{t+1} = T_h^t + \Delta T_h^t$

$\Delta w_{hi}^t = \eta (d^t - y^t) y^t (1- y^t)T_h^t H_h^t (1 - H_h^t) x_i^t + \alpha \Delta T_h^{t-1}$
(0.10)

where $h=0,\ldots,m$ and $i=0,\ldots,n$ $H_0=+1$ $x_0=+1$

$w_{hi}^{t+1} = w_{hi}^t + \Delta w_{hi}^t$

The training of the multilayer perceptron takes O($e * N * f^2$) where e is the number of epochs to train the network, n is the number of instances and f is the number of features. It is f^2 because there are $m * f$ updates for the weights in the first layer and m updates in the second layer. m equals to f / 2 as explained before. So the total number of updates is f^2 / 2 + f / 2 which is O(f^2).

Multilayer perceptron models differ from other multivariate models in its nonlinear nature. With this model one can have nonlinear split at a decision node. In Figure 0.2, an example nonlinear split is shown to solve the problem *Choosing Car.*

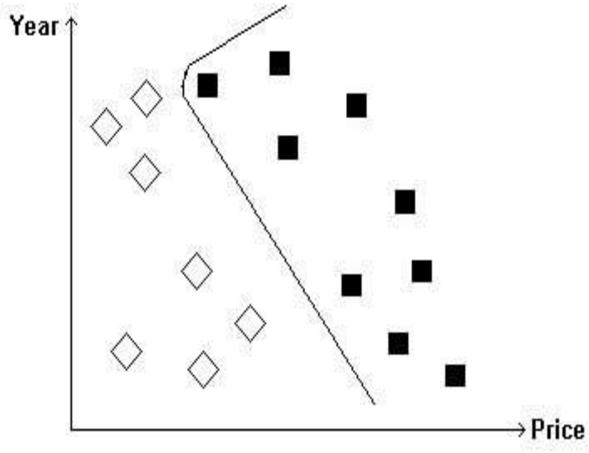


Figure17 A nonlinear split to *Choosing Car* problem

The Hybrid Model:

If we compare the complexity of the decision at a node, we see that the univariate methods are the simplest methods as they test only one feature. Linear methods, which take a linear combination of features, are more complex and nonlinear methods such as multilayer perceptrons are the most complex. But the aim of the decision trees is to find a way of representation of the decision that is as simple as possible and as accurate as possible. So we should not use always nonlinear methods, which have the additional disadvantage that they are not easily interpretable.

Therefore, it seems better to find a way of combining linear and nonlinear methods in a hybrid model. In this model, at each node we train both a linear and nonlinear perceptron and use a statistical test to check if there is a significant performance difference between the two. If the performance of the multilayer perceptron is better then the performance of the linear perceptron with a confidence level of %95, the multilayer perceptron is chosen to have a nonlinear decision node, else linear perceptron is chosen to have a linear decision node.

COMBINATION OF FEATURES EXTRACTED FROM MULTIPLE SOURCES WITH FUZZY REASONING

Tan et.al. Define Fuzzy reasoning is an inference process that is capable of processing data in a way similar to human decision-making. By employing linguistic variables, fuzzy rules provide a high level and efficient interface for building a system with human knowledge. In a fuzzy inference system, the implication rule is represented in a fuzzy relation, and the inferred conclusion is obtained by applying the compositional rule of inference to the fuzzy implication relation. These two properties allow fuzzy reasoning capable of making reasonable inference even when the conditions of an implication rule are only partially satisfied.

The problem in image data fusion based on the decomposition and feature extraction model can be expressed as follows: Given a number of L images $I_j(j = 1,\ldots, L)$ representing different data on the observed scene, a feature extraction operation is made to extract a set of features out of them. The value that associates i^{th} feature $F_i(i = 1,\ldots, N)$ extracted from the j^{th} image I_j is expressed as M_{ij}. At this processing stage, the image data input spaces are transformed to a feature space. The measures of M_{ij} are then combined to make decisions regarding the images to be recognized based on the fuzzy structural description of objects. Note that each object has a set of rules that describe the object in generic features and different objects have their respective fuzzy rule sets.

This process can be formulated as: For an object k, a decision $C_k = F(M_{11}, M_{12},\ldots, M_{ij},\ldots, M_{LN})$ that yields the degree of an object being detected is given by evaluating its fuzzy description rules. An object is said to be recognized when the feature measures obtained match a set of fuzzy description rules of an object. Employment of information from multiple images is achieved in both the antecedent clauses of fuzzy rules and the defuzzification stage. The Mamdani fuzzy reasoning model [21] may be employed, which consists of the following linguistic rules that describe a mapping from $U_1 \times U_2 \times\ldots\times U_r$ to W:
R_i : IF x_1 is A_{i1} and… and x_r is A_{ir} THEN y is C_i

where $x_j(j = 1, 2, \ldots, r)$ are the input variables, y is the output variable, and A_{ij} and C_i are fuzzy sets for x_j and y respectively. Given inputs of the form:
x_1 is A'_1, x_2 is $A'_2,\ldots, x_r$ is A'_r

where $A'_1, A'_2,\ldots, A'_r$ are fuzzy subsets of $U_1 \times U_2 \times\ldots\times U_r$, the contribution of rule R_i to the output of Mamdani model is a fuzzy set whose membership function is computed by

$$\mu_{c'_i}(y) = (\alpha_{i1} \wedge \alpha_{i2} \wedge \ldots \wedge \alpha_{ir}) \wedge \mu_{c_i}(y)$$

where α_i is the matching degree (i.e., firing strength) of rule R_i, and α_{ij} is the matching degree between x_j and R_i's condition about x_j:

$$\alpha_{ij} = \sup_{x_j} \{\mu_{A'_j}(x_j) \wedge \mu_{A_{ij}}(x_j)$$

where $\wedge$ denotes the “min” operator. The final output of the model is the aggregation output from all rules using the max operator:

$$\mu_c(y) = \max \{\mu_{c'_1(y)}, \mu_{c'_2(y)},\ldots, \mu_{c'_L(y)}$$

Note that the output C is a fuzzy set, which can be defuzzified into a crisp output using a defuzzification method, such as the center-of-area (COA) approach [21].
For example, a face may be recognized based on the detection of left eye, right eye, and mouth at the proper positions of an image. A typical rule to combine the feature extraction

CONCLUSION

Depend on application different biometric systems will be more suited than others. It is known that there is no one best biometric technology, where different applications require different biometrics [2]. Some will be more reliable in exchange for cost and vise versa.

Proper design and implementation of the biometric system can indeed increase the overall security. Furthermore, multiple biometric fusions can be done to obtain a relative cheaper reliable solution. The imaged base biometric utilize many similar functions such as Gabor filters and wavelet transforms. Image based can be combined with other biometrics to give more realible results such as liveliness (ECG biometric) or thermal imaging or Gait based biometric systems. A summary of comparison of biometrics is shown in table below [2],

Table 4. Comparison of biometrics systems

	Ease of use	Error incidence	Accuracy	User acceptance	Required security level	Long-term stability
Fingerprint	High	Dryness, dirt	High	Medium	High	High
Hand Geometry	High	Hand injury, age	High	Medium	Medium	Medium
Iris	Medium	Poor Lighting	Very High	Medium	Very High	High
Retina	Low	Glasses	Very High	Medium	High	High
Signature	High	Changing signatures	High	Very high	Medium	Medium
Face	Medium	Lighting, age, hair, glasses	High	Medium	Medium	Medium

REFERENCES

[1]. Woodward, J.D. 1997. Biometrics: privacy's foe or privacy's friend? Proceedings of the IEEE, 85 (9), Sep, 1480 -1492.

[2]. Liu, S., Silverman, M. 2001. A practical guide to biometric security technology. IT Professional, 3 (1) , Jan/Feb, 27 -32

[3]. Wayman, J.L. 1997. A generalized biometric identification system model. Signals, Systems & Computers, 1997. Conference Record of the Thirty-First Asilomar Conference, 1, 2-5 Nov, 291-295.

[4]. Wayman, J.L. 1999. Error rate equations for the general biometric system. IEEE Robotics & Automation Magazine, 6 (1), Mar, 35 -48.

[5]. Fernando L. Podio, “Biometrics - technologies for highly secure personal authentication”, http://csrc.nist.gov/publications/nistbul/itl05-2001.txt

[6]. Chellappa, R., Wilson, C.L., Sirohey, S. 1995. “Human and machine recognition of faces: a survey,” Proceedings of the IEEE, 83 (5), May, 705 -741.

[7]. Barrett, W.A. 1997. A survey of face recognition algorithms and testing results. Signals, Systems & Computers, 1997. Conference Record of the Thirty-First Asilomar Conference, 1, 2-5 Nov, 301 -305.

[8]. Wayman, J.L. 2002. Digital signal processing in biometric identification: a review. Image Processing. 2002. Proceedings. 2002 International Conference, 1, I-37 -I-40

[9]. Jain, A., Ross, A., Prabhakar, S. 2001. Fingerprint matching using minutiae and texture features. Image Processing, 2001. Proceedings. 2001 International Conference, 3, 282 -285.

[10]. Prabhakar, S., Jain, A.K., Jianguo Wang, Pankanti, S., Bolle, R. 2000. Minutia verification and classification for fingerprint matching. Pattern Recognition, 2000. Proceedings. 15th International Conference, 1, 25 -29.

[11]. Huvanandana, S., Changick Kim, Jenq-Neng Hwang. 2000. Reliable and fast fingerprint identification for security applications. Image Processing, 2000. Proceedings. 2000 International Conference, 2, 503 -506.

[12]. Connell, J.H., Ratha, N.K., Bolle, R.M. 2002. Fingerprint image enhancement using weak models. Image Processing. 2002. Proceedings. 2002 International Conference, 1, I-45 -I-48.

[13]. Emiroglu, I., Akhan, M.B. 1997. Pre-processing of fingerprint images. Security and Detection, 1997. ECOS 97., European Conference, 28-30 Apr, 147 -151.

[14]. Xiping Luo, Jie Tian, Yan Wu, 2000. A minutiae matching algorithm in fingerprint verification. Pattern Recognition, 2000. Proceedings. 15th International Conference, 4, 833 -836.

[15]. Sanchez-Reillo, R., Sanchez-Avila, C., Gonzalez-Marcos, A. 2000. Biometric identification through hand geometry measurements. Pattern Analysis and Machine Intelligence, IEEE Transactions, 22 (10), Oct, 1168 -1171.

[16]. Sanchez-Reillo, R. 2000. Hand geometry pattern recognition through Gaussian mixture modeling. Pattern Recognition, 2000. Proceedings. 15th International Conference, 2, 937 -940.

[17]. Cross, J.M., Smith, C.L. 1995. Thermographic imaging of the subcutaneous vascular network of the back of the hand for biometric identification. Security Technology, 1995. Proceedings. Institute of Electrical and Electronics Engineers 29th Annual 1995 International Carnahan Conference, 18-20 Oct, 20 -35.

[18]. Sang Kyun Im, Hyung Man Park, et. al. 2001. An biometric identification system by extracting hand vein patterns. Journal of the Korean Physical Society. March, 38 (3), 268-72.

[19]. Yong Zhu, Tieniu Tan, Yunhong Wang. 2000. Biometric personal identification based on iris patterns. Proceedings 15th International Conference on Pattern Recognition. 2, 801-4.

[20]. Negin, M., Chmielewski, T.A., Jr., et. al. 2000. An iris biometric system for public and personal use. Computer, 33 (2), Feb, 70 -75.

[21]. de Martin-Roche, D., Sanchez-Avila, C., Sanchez-Reillo, R. 2001. Iris recognition for biometric identification using dyadic wavelet transform zero-crossing. Security Technology, 2001 IEEE 35th International Carnahan Conference, Oct, 272 -277.

[22]. Daugman, J. 2002. How iris recognition works. Image Processing. 2002. Proceedings. 2002 International Conference, 1, I-33 -I-36.

[23]. Li Ma, Yunhong Wang, Tieniu Tan. 2002. Iris recognition using circular symmetric filters Pattern Recognition, 2002. Proceedings. 16th International Conference, 2, 414 -417.

[24]. Podio, F.L. 2002. Personal authentication through biometric technologies. Networked Appliances, 2002. Proceedings. 2002 IEEE 4th International Workshop, 57 -66.

[25]. Xiaoyi Jiang, Mojon, D. 2003. Adaptive local thresholding by verification-based multithreshold probing with application to vessel detection in retinal images. Pattern Analysis and Machine Intelligence, IEEE Transactions, 25(1), 131 -137.

[26]. Camino, J.L., Travieso, C.M., Morales, C.R., Ferrer, M.A. 1999. Signature classification by hidden Markov model. Security Technology, 1999. Proceedings. IEEE 33rd Annual 1999 International Carnahan Conference, 481 -484.

[27]. Vergara da Silva, A., Santana de Freitas, D. 2002. Wavelet-based compared to function-based on-line signature verification. Computer Graphics and Image Processing, 2002. Proceedings. XV Brazilian Symposium, 218 -225.

[28]. Lin Hong; Anil Jain. 1998. Integrating faces and fingerprints for personal identification. Pattern Analysis and Machine Intelligence, IEEE Transactions, 20(12), Dec, 1295 -1307.

[29]. Jain, A.K., Ross, A. 2002. Learning user-specific parameters in a multibiometric system. Image Processing. 2002. Proceedings. 2002 International Conference, 1, I-57 -I-60.

[30]. Dugelay, J.L., Junqua, J.C., Kotropoulos, C., Kuhn, R., Perronnin, F., Pitas, I. 2002. Recent advances in biometric person authentication. Acoustics, Speech, and Signal Processing, 2002 IEEE International Conference, 4, IV-4060 -IV-4063.

[31]. Osadciw, L., Varshney, P., Veeramachaneni, K. 2002. Improving personal identification accuracy using multisensor fusion for building access control applications. Information Fusion, 2002. Proceedings of the Fifth International Conference, 2, 1176 -1183.

[32]. Kittler, J., Messer, K. 2002. Fusion of multiple experts in multimodal biometric personal identity verification systems. Neural Networks for Signal Processing, 2002. Proceedings of the 2002 12th IEEE Workshop, 3 -12.

[33]. Czyz, J., Kittler, J., Vandendorpe, L. 2002. Combining face verification experts. Pattern Recognition, 2002. Proceedings. 16th International Conference, 2, 28 -31.

[34]. Belhumeur, P.N.; Hespanha, J.P.; Kriegman, D.J. “Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection”, Pattern Analysis and Machine Intelligence, IEEE Transactions on , Volume: 19, pp 711-720, Issue: 7 , July 1997

[35]. K. C. Tan, T. H. Lee, and M. L. Wang “Computational Intelligence Approach To Object Recognition” Chapter 16 Chapter 16 – Object Recognition,Page No. 351-364.