الجامعة الإسلامية العالمية ماليزيا
INTERNATIONAL ISLAMIC UNIVERSITY MALAYSIA
يونيرسيتي إسلام انتارابغسا ملايسيا

BIOMETRICS-BASED INTELLIGENT SECURITY SYSTEM

BY
SURIZA AHMAD ZABIDI

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENT FOR THE DEGREE OF COMPUTER AND INFORMATION ENGINEERING

KULLIYAH ENGINEERING
INTERNATIONAL ISLAMIC UNIVERSITY MALAYSIA

MARCH, 2004

ABSTRACT

This thesis presents the design and development of a biometric-based security system. User authentication forms an essential part of the overall system security. Traditionally, user authentication means providing an identification number or a password that is unique and this has been in use for decades. This type of security system is very fragile in an area where a higher level of security system is required. Biometrics-based system offers a new and better approach to user authentication. Biometrics authentication is an automated method whereby an individual identity is confirmed by examining a unique physiological trait or behavioural characteristic, such as fingerprint, iris, or signature, since physiological traits have stable physical characteristics. In this thesis, the design and implementation of fingerprint security system comprising the scanner, interface system, Boltzmann Machine Neural Network and the access control system is investigated. The results obtained both for the simulation studies and testing of the real physical system have demonstrated the practicality of such system, which has potential applications in many fields.

ملخص البحث

قام هذا البحث بتصميم منظومة أمان تعتمد على قياسات حيوية للمستخدم. التعرف على المستخدم من أساسيات الأمان الكلى للمنظومة كان يتم بطريقة تقليدية، حيث كان التعرف على المستخدم يتمثل فى وجود كلمة سر أو رقم خاص بكل مستخدم، وظل هذا هو المتعارف عليه فى مجال الأمان لعهود متعاقبة. يعتبر هذا الأسلوب التقليدى سهل الاختراق عندما تكون هناك متطلبات عليا لأمان المنظومة. منظومة الأمان المعتمدة على المقاييس الحيوية تتيح أسلوباً جديداً وأفضل من سابقه للتعرف على المستخدم. التعرف على المستخدم عن طريق القياسات الحيوية يتم تلقائياً حيث يتم التأكد من شخصية المستخدم عن طريق اختبار صفة معينة من صفاته العضوية أوالسلوكية، مثل بصمة الأصبع، قذحية العين أو الأمضاء، وذلك لأن الخصائص العضوية لها صفات طبيعية ثابتة. قام هذا البحث بدراسة، تصميم وتنفيذ منظومة أمان تعتمد على بصمة الاصبع وتشتمل على ماسحة ضوئية، نظام تداخلى، شبكة عصبية بآلية بولتزمان ومنظومة تتحكم فى الدخول. النتائج التى تم التوصل اليها فى هذا البحث عن طريقتى المحاكاة والتجارب الفعلية على المنظومة الحقيقية بينت الجدوي العملية لهذه المنظومة، مما سيكون له تطبيقات مهمة فى مجالات عديدة.

APPROVAL PAGE

I certify that I have supervised and read this study and that in my opinion, it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a thesis for the degree of Master of Computer and Information Engineering.

Momoh-Jimoh E. Salami
Supervisor

I certify that I have read this study and that in my opinion, it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a thesis for the degree of Master of Computer and Information Engineering.

for Othman O. Khalifa
Examiner (Internal)

Ramachandran Nagarajan
Examiner (External)

This thesis was submitted to the Department of Electrical and Computer Engineering and is accepted as partial fulfillment of the requirements for the degree of Master of Computer and Information Engineering.

Farhat Anwar
Head, Department of Electrical and Computer Engineering

This thesis was submitted to the Kulliyyah of Engineering and is accepted as partial fulfillment of the requirement for the degree of Master of Computer and Information Engineering.

Ahmad Faris Ismail
Dean, Kulliyyah of Engineering

DECLARATION

I hereby declare that this thesis is the result of my own investigation, except where otherwise stated. Other sources are acknowledged by footnotes giving explicit references.

Name: Suriza Ahmad Zabidi

Signature: _____ Date: 10-05-2004

INTERNATIONAL ISLAMIC UNIVERSITY MALAYSIA

DECLARATION OF COPYRIGHT AND AFFIRMATION OF FAIR USE OF UNPUBLISHED RESEARCH

BIOMETRICS-BASED INTELLIGENT SECURITY SYSTEM

Affirmed by..... Suriza Ahmad Zabidi

.....
Signature

10-05-2004
.....
Date

ACKNOWLEDGEMENTS

First I would like to thank God for giving me the strength, motivation and guidance to complete this paper.

I would like to gratefully acknowledge the contributions of several people who have helped me to complete this work. Firstly, I would like to convey my grateful thanks to my supervisor Professor Dr Momoh-Jimoh E.Salami for his help.

Secondly, I would like to thank to my beloved husband, Sany Izan b Ihsan for helping me with so many ways so as to complete this thesis. My thanks also go to my children that have shown a lot of patience and understanding during my period of study.

My words of gratitude also go to Professor Dr Ahmad Faris Ismail , Dean of Kulliyah of Engineering, Associate Prof. Dr Farhat Anwar, Head of Electrical and Computer Engineering Department, and to all my colleagues for their support.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ABM	Attrasoft Boltzmann Machine
AFI	Automated Fingerprint Identification
AOI	Area of Interest
EMD	Euclidean Minimum Distance
ESD	Electro-Static Discharge
ERR	Equal Error Rate
FA	False Acceptance
FRR	False Rejection Rate
FAR	False Acceptance Rate
GIF	Graphical Interchange Format
HMM	Hidden Markov Models
JPEG	Joint Photographic Experts Group
MLP	Multi-layer Perceptron
NIST	National Institute of Standards and Technology
NRML	Normal
PIN	Personal Identification Number
POPFNN	Pseudo Outer Product-based Fuzzy Neural Network
POPFNN-FVS	Pseudo Outer Product-based Fuzzy Neural Network – Fingerprint Verification System
QDM	Quadratic Minimum Distance
ROC	Receiver Operating Characteristic
TCM	Traditional Chinese Medicine
TR	True Rejection
WSNN	Weighted Several Nearest Neighbor

NOMENCLATURES

The notation below will be use in the descriptions of the classification functions.

N	=	number of classes. For fingerprint Pattern-level Classification Automation (PCA), N = 5
$p(i)$	=	probability that a random fingerprint is of class i ($1 \leq i \leq N$)
$\hat{p}(i)$	=	an estimate of $p(i)$
n	=	number of features used
R^n	=	the set of all n-tuples of real numbers = 'feature space"
X	=	a "feature vector" $x \in R^n$
M_i	=	number of training prints of class i ($1 \leq i \leq N$)
x_j^i	=	feature vector for class i ($1 \leq i \leq N$) ($1 \leq j \leq N$) ($x_j^{(i)} \in R^n$)
μ_i	=	mean feature vector for class i ($1 \leq i \leq N$) ($\mu_i \in R^n$)
m_i	=	an estimate of μ_i
$\sum_i$	=	covariance matrix for class i ($1 \leq i \leq N$) ($\sum_i$ is an n x n matrix)
S_i	=	an estimate of $\sum_i$
$d^2(x,y)$	=	$(x - y)^T (x - y)$= squared Euclidean distance between x and y (x, y $\in R^n$)
$D_i(x)$	=	i^{th} discriminant function ($1 \leq i \leq N$, $x \in R^n$)
$\lambda(i \mid j)$	=	loss incurred by classifying to i a print that is of class $j (1 \leq i, j \leq N)$
$p(x)$	=	mixture density: for $S \subseteq R^n$, $\int_S p(x)dx = P(x \in S)$
$p(x \mid i)$	=	conditional density: $S \subseteq R^n$, $\int_S p(x \mid i)dx = P(x \in S \mid x$ is from a class – i print)
$p(i \mid x)$	=	a posteriori probability: for a particular x, $p(i \mid x) = P$ (x is from a class – i print)
$N^{(i)}$	=	number of nodes in i^{th} layer (i = 0, 1, 2)

$f(x)$	=	$1/(1 + e^{-x})$ = sigmoid function
$b_i^{(k)}$	=	bias weight of i^{th} node of k^{th} layer
$w_{ij}^{(k)}$	=	weight connecting ith node of k^{th} layer to j^{th} node of $(k-1)^{th}$ layer (k = 1, 2 ; $1 \le i \le N^{(k)}$;$1 \le j \le N^{(k-1)}$
x	=	$(x_1 \ldots xn)^T$ = a feature vector
α	=	neighborhood-size factor

CHAPTER 1: INTRODUCTION

1.1 Overview of the biometrics system

Biometrics system has been in use since the year the pyramid was built. The man in charge of the administration and provision of food and wages already created a system to overcome the difficulty in handling all the workers in the pyramid site. This is one of key elements in the overall smooth running and success of the project[1]. Since that time, the development of biometrics-based system has increased rapidly. The system during the Pyramid building time was done manually but now with the advancement of the microelectronics, especially computer and microprocessor the use of biometrics system has improved tremendously.

The term *biometrics* refers to a science involving the statistical analysis of biological characteristics. Julian Ashbourn in his book [4] defined biometrics as a measurable physiological and/or behavioral trait that can be captured and subsequently compared with other instances at the time of verification. Some of the examples of biometrics system are in matching fingerprints, voice patterns, hand geometry, iris and retina scans, vein pattern and this list would continue etc.

In other technologies, we usually analyze human characteristics so as to automatically recognize or verify a particular identity, whereas in biometrics technique we measure physical or behavioral characteristics of an individual. The physical and

behavioral characteristics chosen for identification should basically satisfy the following conditions [38,61]:

- *Universality*: indicates that every person should have his own or this characteristic;
- *Uniqueness*: means that any two persons should be different enough to distinguish each other based on this characteristic;
- *Permanence*: indicates that the characteristic should be stable enough and should not change significantly with environment or time;
- *Collectable:* which means that the characteristic can be measured quantitatively;
- *Acceptability*: indicates to what extent people are willing accept the biometrics system;
- *Performance*: refers to the achievable identification accuracy, the resource requirements to achieve an acceptable identification accuracy, and the working or environment factors that effect the identification accuracy;
- *Circumvention*: refers to how easily it is to fool the system by fraudulent techniques.

Generally, physical and behavioral characteristics used in biometrics include the following [5,16,22,60,63]

Physical characteristics

- Chemical composition of body odor
- Facial features and thermal emission
- Features of the eye, i.e., retina and iris
- Fingerprints
- Palm-prints

- Hand geometry
- Skin pores
- Wrist/hand veins

Behavioral characteristics

- Handwritten signature
- Keystrokes or typing
- Voiceprint
- Gait
- Gesture

Some examples of physical and behavioral characteristics are illustrated in Figure 1.1.

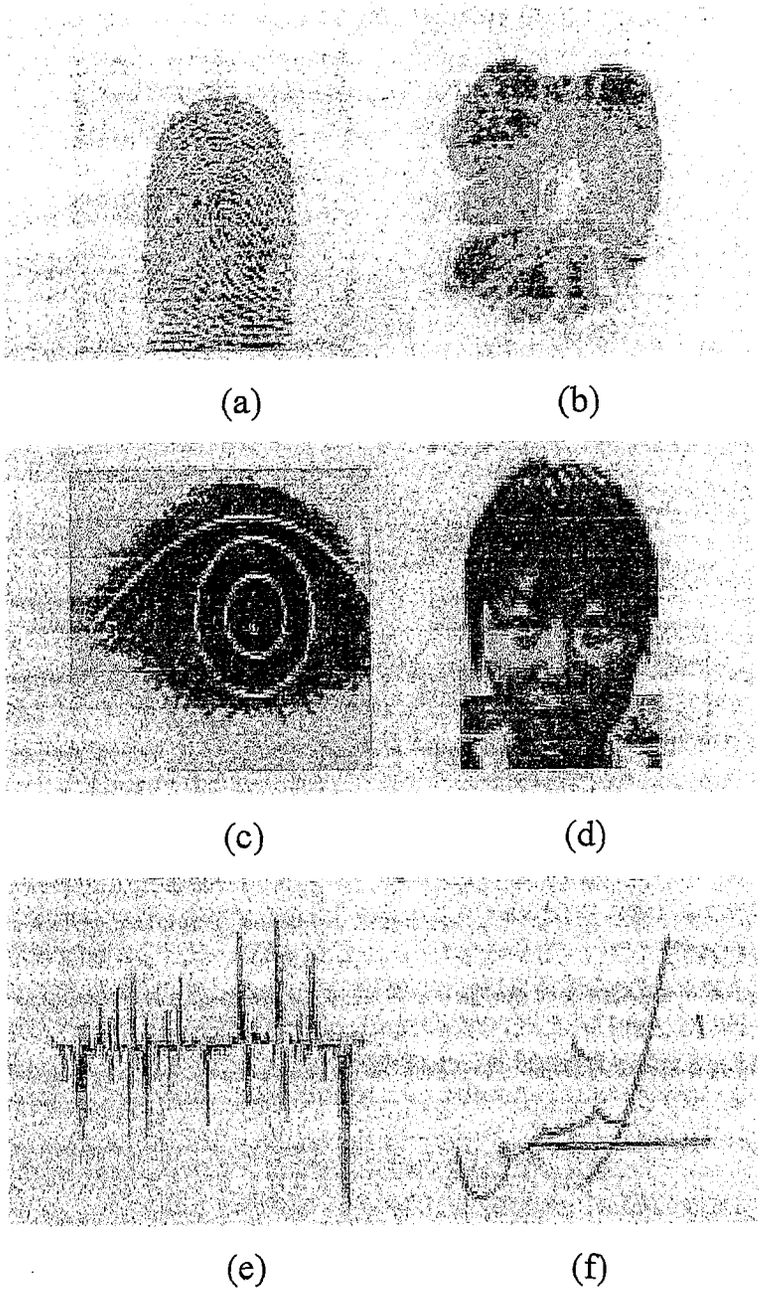


Figure 1.1: Physical characteristic (a) Fingerprints (b) Palm print (c) Iris (d) Face; Behavioral characteristic: (e) Voiceprint, and (f) Handwriting signature

1.2 Biometrics System Structure

In general, there are two parts in biometrics system; enrollment part and identification part. The function of the enrollment part is to have a user's characteristic registered so that it can be used as a criterion when identification is performed. The function of the identification part is to provide a user interface, which has the end user's characteristic, captured and verified. For the enrollment part, the procedure consists of sample capturing, feature extraction and storage whereas, in the

identification part, the procedure is formed by four stages; capture, feature extraction, comparison and decision [37]. This is shown in Figure 1.2.

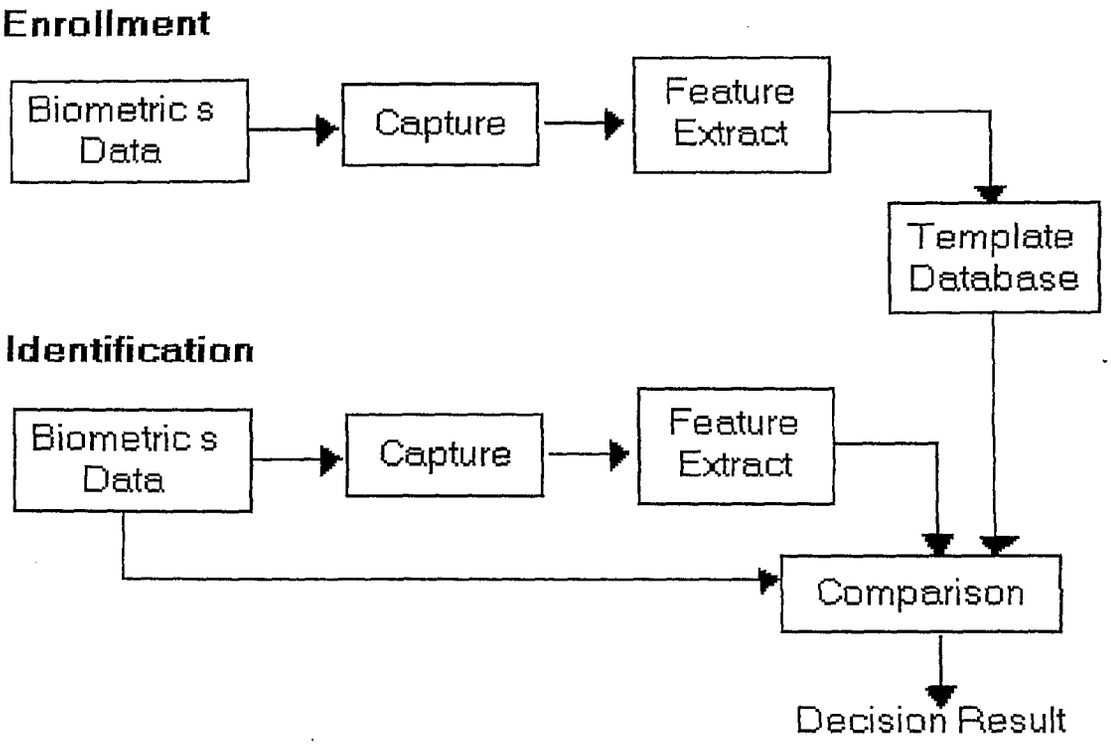


Figure 1.2: General Procedure of Biometrics system

Since the capture and feature extraction stages in the enrollment part are the same as the first two stages in the identification part, a biometrics system can be simplified by four processing parts as shown in Figure 1.3

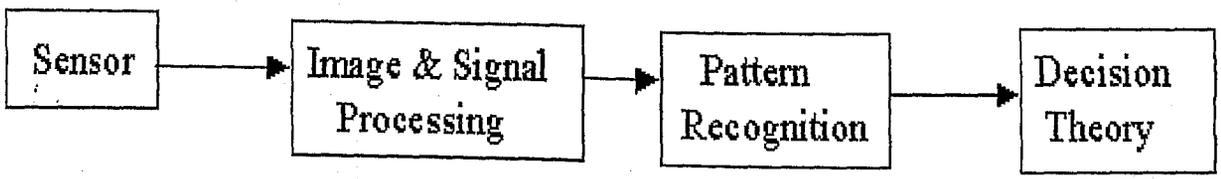


Figure 1.3: Biometrics system general procedure with four processing stages

Capture Stage

In this stage, a physical or behavioral sample is input into the system. Different system may use different devices to get the sample. The original device

signals are then translated into digital codes with or without preprocessing. The quality of the captured data is very important since the capture is the first stage of automatic person identification.

Feature Extraction Stage

At this stage, a unique data from the sample is extracted and a template is created. The template for any two persons should be different and different samples from the same person should be similar enough.

Comparison Stage

Comparison stage is a stage to compare the newly extracted template from a sample to a registered template in the system. Since even samples from the same person may vary from time to time, the comparison algorithm should tolerate the tiny change from the same person yet capable enough to distinguish different person. In practice, there are no two samples that are exactly the same.

This comparison stage can be divided into two categories; identification and verification. These two categories are different in their ways of comparing the templates in the system. Identification means the new template will be compared with all registered templates in the system where as verification implies that the new template will only be compared to a particular registered one. In short, identification is one-to-many comparison (1: N) while verification is one-to-one comparison (1:1).

Decision Stage

In decision stage, the system will decides whether the template extracted from the new sample match the registered one. For this reason threshold is set to get the definite answer of *yes* or *no*. When the matching score is greater than the threshold, an answer *yes* is given, otherwise *no* is as an output.

The biometrics industry has used two performance measurements to run the level of matching accuracy for many years [52,75]. As mentioned above, when a biometrics system is used, it will either match or not match the extracted biometrics data. The score is given to compare between the new sample and a registered template. If the score is higher than a given threshold then a match is returned. This technique gives biometrics far more flexibility than the *yes* or *no* approach used by the PIN or password-based technologies.

The two performance measurements for biometrics system are False Rejection Rate (FRR) and False Acceptance Rate (FAR). FRR is the rate at which the system incorrectly rejects legitimate attempt of verification. FAR is the rate at which the system incorrectly accepts an invalid verification attempt. Both rates are expressed as

$$FRR = \frac{NFR}{NAA} \times 100\% \tag{1.1}$$

$$FAR = \frac{NFA}{NIA} \times 100\% \tag{1.2}$$

where NFR and NFA are the Numbers of False Rejections and False Acceptances respectively, NAA is the Number of Authorized Identification or verification attempts and NIA is the Number of Impostor Identification attempts.

There is one more performance measurement that is less frequently used, Equal Error Rate (EER) or Crossover Rate which is the point at which the FRR and FAR meet or crossover, as shown in Figure 1.4. For example, a system with an FRR and FAR of 1% will also have an EER of 1%.

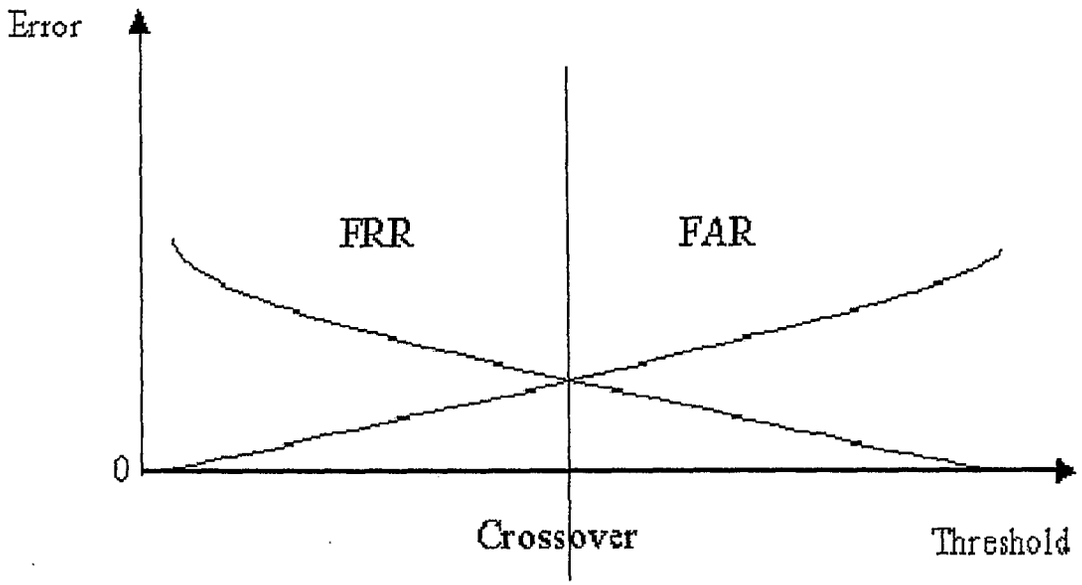


Figure 1.4: FRR, FAR, and ERR

The use of FRR and FAR as a performance measurement has become a disputable matters within the biometrics industry, particularly concerning the statistical significance of such a simplified calculation. The given formula to calculate FRR and FAR are spectacularly simple compared to the large number of variables that may be present in the biometrics system. The performance of biometrics system may be affected by:

- Environmental conditions (for example, extreme temperature and humidity can effect a system's performance).
- The beliefs, desires and intentions of the user (if the user does not wish to interact with the system, then the performance will be affected).
- The age, gender, ethnic background and occupation of the user.

- The physical make-up of the user.

When the vendor of biometrics publishes both rates to demonstrate performance, the result can be from the laboratory test under controlled conditions. Performance claimed should therefore be treated with care because they can be calculated using a 'one-try' or 'three-try' method, depending on the nature of the biometrics application and the type of biometrics system used. That is the user may either be given a single attempt or three attempts, at comparing the new sample with the template. Common sense rules that if three attempts are used, this will improve the FRR.

The other problem in the lack of interest of the FRR and FAR is that the rates can be manually configured. For example, a bank may require a very high level of security such that the FAR must be less than 0.1%, i.e. the system would only grant unauthorized access for one in a thousand instances. This rate can be achieved by alter the system FAR and by doing so however will consequently effect FRR.

In short, the FRR and FAR are a simplistic means of evaluating performance with merits and demerits. It is helpful to understand the basic accuracy of a biometrics system, but one must remember that the circumstances of an application can affect the performance rates.

In order to provide a more reliable assessment of a biometrics system, some more descriptive performance measures are necessary. Receiver Operating Curve (ROC) and statistical metrics denoted as *d'* are the two other commonly used measures

[2]. The ROC is the measure of correct acceptance rate against FAR, which gives a good representation of the trade-off between false acceptance and false rejection errors and can be used to select an appropriate operating point for a particular application. The statistical metric d' gives an indication of the separation between the genuine distribution and impostor distribution. It is defined as the difference between the means of the genuine distribution and imposter distribution divided by a conjoint measure of their derivative, that is [39]

$$d' = \frac{\| M_{impostor} - M_{genuine} \|}{\sqrt{(SD^2_{impostor} + SD^2_{genuine})/2}} \tag{1.3}$$

where $M_{genuine}, M_{impostor}, SD_{genuine},$ and $SD_{impostor}$ are the means and standard deviations of the genuine distribution and impostor distribution respectively. However, this measurement also depends heavily on the test data and test environment. To precisely represent the accuracy test of the entire population of interest, enough samples should be available, and the samples should represent the population and the categories (impostor and genuine) to be identified.

1.2.1 Biometrics Technology

In this section, some of the popular biometrics technology going to be explained, how they work and what applications they may be best suited for. There is no one methodology that can claim as the best technology of all. They all depend on what one is trying to achieve and under what conditions or who are the target users. A methodology that works well within a restricted environment, for example, may be less suitable for busy public airport or a factory shop floor.

Fingerprints

When the term *biometrics* is mentioned, majority of people immediately associate it with fingerprint. This is partly because biometrics is often associated with security and identification. In fact, fingerprints are the primary identification among law enforcement agencies all over the world. In fact, many designs of fingerprint system are reflected from the way fingerprints have been utilized manually over the years, that is by seeking to identify minutiae features and their relative position with their print [3]. Fingerprint experts have categorized fingerprints into primary types such as whorls, loops, tent, and arches. These are illustrated in Figure 1.5 [54]. There are subclasses, which are subsets of the primary fingerprint classes. Such sub-class includes twin-loop, tented arch, left and right loop. The cores and deltas of a fingerprint are usually used as reference points [47] in order to obtain its basic classification.

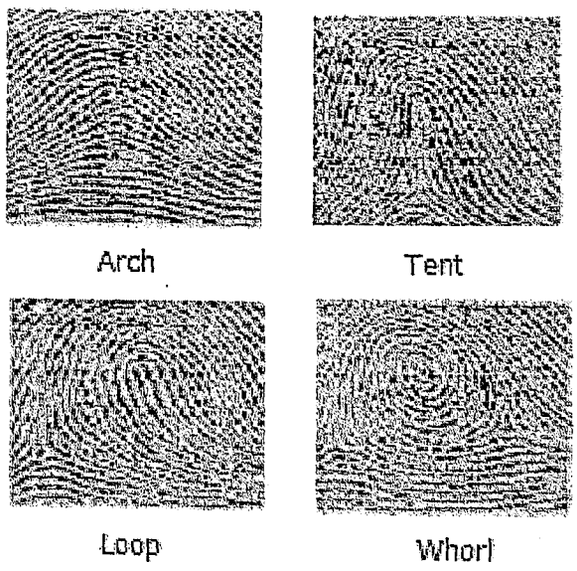


Figure 1.5: Primary fingerprint classes

Fingerprint characteristics or ridge features are the essential elements for fingerprint identification. They are actually interruptions to the normal flow of the

ridges and are broadly classified into two classes: primitives and compound features (combination of primitives). Figure 1.6 shows the primitive and compound in fingerprints [31].

Presently, automatic fingerprint identification (AFIS) computer matching systems have drastically reduced the time needed to scan even a very large database of fingerprints to have potential matches. Fingerprints have been used in a fairly broad range of biometrics applications depending on how well the systems have been designed and how the systems meet the requirements. However, there are some situations where the methodology would not be well suited such as, applications where the biometrics reader would be deployed in a harsh environment and subjected to higher than usual levels of contamination or abrasion. Similarly, an application where users are likely to have dirty hands due to the nature of their work would not be ideally suited for fingerprint system. The system is also not suitable in cold whether and wet external situation.

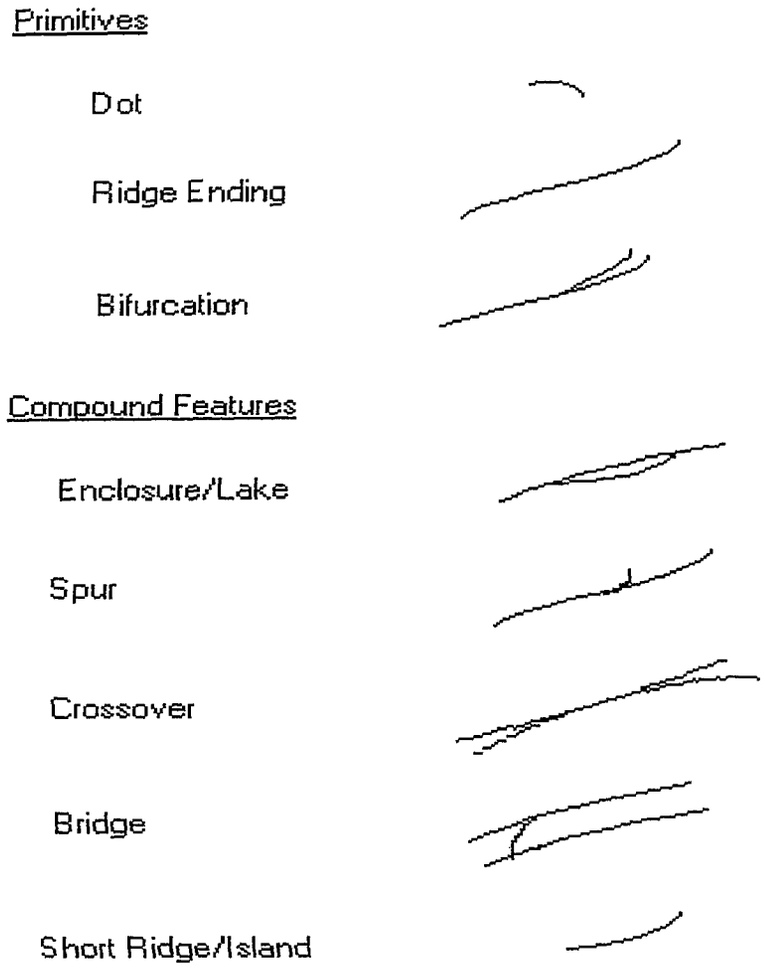


Figure 1.6: Primitive and Compound Features of Fingerprints

Some common applications for fingerprints systems are [6]:

a) PC or network access where the readers can be deployed in controlled interior situation and integrated seamlessly into a familiar process. Furthermore, the environment is relatively clean, controlled and the user will probably adjust well to using the fingerprint device.

b) Time and attendance monitoring terminals where a fingerprint scanner could successfully integrate into the existing process while adding a valuable extra level of identity authentication.

c) Integration into chip cards applications.

d) Ability to integrate biometrics via custom application development and the provision of readers on an OEM basis is another area that can serve by the variety of fingerprint readers available with SDKs (software development kits).

In conclusion, fingerprint biometrics-based system can be a suitable application in many areas of security system. The development of such a system will ensure the enhancement of the security level. Furthermore, the design and development of fingerprint biometrics-based system has evolved to become matured and more user friendly from year to year.

Hand geometry **[76]**

Hand geometry is one of the first biometrics applications to prove practicable for use across a variety of real world applications. This is because of its easy-to-use, good performance, and easily adaptable to a variety of applications in physical access control, time monitoring and other areas. It is easy to configure and administer.

However, hand geometry readers are a little bulky and are not easy to blend seamlessly into other equipment. In a heavily used environment, the platen surface will require periodic cleaning to satisfy both operational and aesthetic points of view. There are some sectors in the application that would not make hand geometry system applicable to all users, for example, young children and those with physical disability such as arthritis.

In general, hand geometry biometrics is a good choice for a general purpose biometrics due to it easy deployment. However, its bulkiness limits its application.

Iris scanning [64]

Irises were first realized to be unique when ophthalmologists noticed that iris pattern were not only very individual, but also didn't seem to change with age. Further study of clinical photographs spanning several decades served to confirm this observation. After that the idea of iris being at least as unique as fingerprint was established. Iris pattern is not only unique to individual, but that left and right irises within the same individual themselves were unique. Furthermore, this is also true for family siblings and even identical twins, where other genetic details such as facial appearance are so similar.

The texture of the iris is made up of a complex fibrous and elastic structure sometimes referred to as the trabecular meshwork, the fine detail being established prior to birth and remains intact throughout ones life. Some of the potential applications for iris scanning are financial sector for ATMs, general access control in several sectors and various high-security applications.

In conclusion, iris scanning is a very high accuracy biometrics technique for certain or appropriate applications when all considerations like user interface and mechanical requirements have been taken care of. Without good consideration and technique, iris scanning biometrics will not be suitable or applicable for certain applications. Furthermore, the installation cost might be higher than some other methodologies. However, this is all relative to the application under consideration and the associated perceived benefits of introducing a biometrics into the process.

Retina scanning **[62]**

Before the iris scanning was developed, retina devices were available commercially in various military and high-security applications. The principle behind retinal scanning is that the blood vessels at the retina provide a unique pattern, which may be used as a tamper-proof personal identifier. Furthermore, the patterns of vascular retina were both intricate and stable.

Retinal scanning is hardly the most user-friendly biometrics techniques but it does promise high level of accuracy. Although the size of the database being search might have an effect on the individual transaction time but retinal scanning operates satisfactorily in identification mode.

Facial recognition **[44]**

Facial recognition systems can be divided into two primary groups. The first group is called the *control scene group* and the second group is called *random scene group.* For the first group, the tested subject is located in a known environment with a minimum variation of the scene. An example of this application is in a typical access control situation where the subject will ordinarily be facing the camera at a fairly constant distance. The second group the tested subject located at various distances firm the camera and in various degrees of axis from the straight-ahead position. An example of such application is when a system attempting to identify the presence of an individual within a group or crowds. These two primary groups represent quite different propositions to the system designer.

As a conclusion, facial recognition is a very interesting biometrics technique and more sophisticated security that is based on this technique would be developed in the coming years. Its applications do require a little more thought than other biometrics and would probably be well suited to modified situation, which can be carefully designed to accommodate the special requirements of this technique.

Voice verification **[40]**

Voice verification can be considered as a fairly natural technique as we are using our voices in everyday conversation to expedite a number of transactions. It is also one of the early biometrics examples in terms of commercially available products.

The principle behind voice verification is that the physical construction of an individual's vocal cords, vocal tract, palate, teeth, sinus and tissue within the mouth will affect the dynamics of speech. If the speech of two identical persons from the same family (say a brother and sister) is analyzed, then one would find quite noticeable and repeatable differences, even though to our ears the two sounds are superficially alike.

In the early time, the implementation of voice verification tended to take the form of a wall-mounted stand-alone device which users would be required to speak into in order to annunciate their password or phrase and be verified. Another approach is to use a proximity transducer, such as telephone handset. However, both systems have their own strength and weakness and more research work are needed.

As a conclusion, facial recognition is a very interesting biometrics technique and more sophisticated security that is based on this technique would be developed in the coming years. Its applications do require a little more thought than other biometrics and would probably be well suited to modified situation, which can be carefully designed to accommodate the special requirements of this technique.

Voice verification **[40]**

Voice verification can be considered as a fairly natural technique as we are using our voices in everyday conversation to expedite a number of transactions. It is also one of the early biometrics examples in terms of commercially available products.

The principle behind voice verification is that the physical construction of an individual's vocal cords, vocal tract, palate, teeth, sinus and tissue within the mouth will affect the dynamics of speech. If the speech of two identical persons from the same family (say a brother and sister) is analyzed, then one would find quite noticeable and repeatable differences, even though to our ears the two sounds are superficially alike.

In the early time, the implementation of voice verification tended to take the form of a wall-mounted stand-alone device which users would be required to speak into in order to annunciate their password or phrase and be verified. Another approach is to use a proximity transducer, such as telephone handset. However, both systems have their own strength and weakness and more research work are needed.

There are several voice verification systems. The large application of voice verification can be divided into two groups; a conventional physical access control and in broader area of remote identity verification as may be utilized for prison inmates, on line transaction processing, automated call center application and other similar areas.

In conclusion, voice verification is one of biometrics methodologies with a lot of potential uses however more efforts in research work are needed before it can be widely used.

Signature verification **[45]**

Signature verification is a behavioral biometrics rather than an anatomical biometrics such as fingerprint or iris pattern. One advantage from a user point of view is that it is perceived as a natural and familiar action. We have been signing our name as a form of identity verification for years, thousand of years in fact, from the great civilizations of ancient Egypt, China and Mesopotamia to the current day.

In biometrics signature verification not only the appearance of the signature is analyzed but also the dynamic inherent in writing it; how hard do we press down on the writing surface?, how quickly do we execute the first pen stroke?, how does the writing speed vary from the beginning to the end of the signing process?, and how long does it take on an average to write a given signature? There are a number of such parameters inherent within dynamic process of signing our signature.

Some of the applications of signature verification are in the area where the signature is currently used as an identifier, for example in financial transaction such as

degree of universality, acceptability, collectability, and easy of use. Although, iris scanning has a high level of universality, uniqueness and performance but it is low in acceptability level. Therefore, fingerprint techniques are a good choice for security system compare to other methods.

1.3 Problem statement

In today transaction environment, a reliable user authentication system is becoming increasingly important. The consequence of an insecure authentication system can be catastrophic, and may include loss of confidential information. There is an increasing interest and need to use biometrics in everyday authentication application.

One of the earliest identifiers to offer automated recognition is Personal Identification Numbers (PIN). However, it should be understood that this means recognition of the PIN, not necessarily recognition of the person. The same applies with cards and other tokens. A biometrics security system however cannot be easily transferred between individuals as it represents a unique identifier.

The aim of this thesis is to design and develop a security system using biometrics recognition system. In this thesis, the proposed recognition system will be designed and is referred to as automated *fingerprint verification system.* Fingerprint system is chosen because its use as a biometrics system is both the oldest in identification and the most prevalent in use today. Fingerprints also are one of the most matured biometrics technologies used in forensic division worldwide. Furthermore, the formation of fingerprints depends on the initial conditions of the embryonic

development. So, they are unique to each person and each finger. Fingerprints pattern also would not going to change or ageing from year to year.

In this thesis, neural network will be used in the verification process. Neural network is preferred due to its robustness in many applications. The design and development of an intelligent security system will be carried out here in various stages as subsequently discussed. The first step will be to capture the fingerprint image from a scanner or an optical sensor. Then, the image will be converted into neural data which would be used as input to the artificial neural network so as to recognize the different types of fingerprint images. The fingerprint image is then stored in a directory. The stored images will be used to verify life fingerprints whenever it is required. The verification of the image will be done by interfacing the scanner and the matching engine system which in this case is Attrasoft ImageFinder 5.4 DOS version. The final stage is to integrate and test the developed system.

1.4 Methodology

For the verification level, neural network software developed by Attrasoft Inc. is used. This software is based on Boltzmann Machine neural. The Attrasoft ImageFinder trains the artificial neural network as to recognize an image. Similar action is taken by Attrasoft ImageFinder if there are several images. Attrasoft ImageFinder can therefore match images in the form of Joint Photographic Experts Group (jpg/jpeg) or Graphic Interchange Format (gif) which:

- Look like an image (called key-image) or a segment of an image (called key-segment)
- Look like several key-image or key-segment.

After the artificial neural network is trained with a sample image, a template is created to store the image. The image will be compared with the captured image by the scanner later. The scanner will need to be integrated with the software so that a verification function can be done. The scanner will scan an image by placing a finger or fingers in the surface of the scanner. The image will be captured and the network will send the image to be verified. In order to do this and to make the process automated we need an interfacing system. In short, a scanner will be connected to a computer where the verification software is installed. The communication between the scanner and the software will be handled by the interface system. A good interface system is needed for the system to run successfully and smoothly.

1.5 Objectives of Research

The objectives of the design and development of this fingerprint verification system is to improve the level of security of the existing or traditional verification system. Verification systems that exist nowadays still use the traditional methods of security such as keys, passwords, tag, token and other methods. This type of security is very easily being exposed to fraud. The keys, passwords, tag and token are easily being misplaced, forgotten, duplicate and lost. Furthermore, this traditional system does not represent us as an individual with unique characteristic but something that we possess. This is where biometrics system comes in to make the verification system to represent our individuality and therefore will increase the level of security in the area of applications. Our individuality is something unique that we have from birth until death. Therefore it is hard to be duplicated, forgotten, lost or stolen. The application of the biometric system also is one of the objectives of this design. One of the aims is

to study the available biometrics system that being used in real application and decide on one method that best suites the design and development of this particular design.

Another objective of the design and development of the system is to investigate how artificial neural network can help in improving the verification system. By having neural network in the system it will add the level of intelligent to the system.

Furthermore, it is also the aim of the thesis to study and develop the interface for the whole system and test it applicability in real time.

1.6 Summary

Security has become a major concerned in many parts of the world. In our daily life we deal with a lot of transactions such as ATM, computer or gain access to very restricted areas. For each of these applications security is always an issue. With respect to this, a high security system is required to overcome the flaws in the existing systems. Biometrics-based security system is one of the techniques that can guarantee a high security system. An overview of biometric system is discussed in this chapter. In addition, the discussion further extends to explain about the existing biometric technologies and how the performance of biometrics system is measured. From the discussion, a comparison between biometrics technologies is done. Based on the comparison, one technique for biometrics system is proposed for the design and development in this thesis. The methodology and objectives of the thesis are also discussed in this chapter. The proposed biometrics system is a biometrics-based security system using fingerprint as the verification identity for accessing a restricted area. An Artificial Neural Network is selected for the verification level.

In the next chapter, a literature review is discussed. The discussion is based on the available techniques and to verify that Artificial Neural Network is one technique that can enhance the security system.

CHAPTER 2: LITERATURE REVIEW
TECHNIQUES FOR FINGERPRINT SECURITY SYSTEM

2.1 Introduction

There are different types of fingerprint verification system. Different models use different techniques for fingerprint verification. In the verification or identification, the most important part before the verification or the identification takes place is the classification. How the classifier works will determine whether the sample is matched or not with the stored data. After that the system will verify and reject or accept the identification request. In this chapter, the existing fingerprint classification and verification techniques will be discussed subsequently.

In general, most of the security systems using fingerprint as the identification or verification element will follow similar procedures as explained in chapter one and shown in Figure 2.1

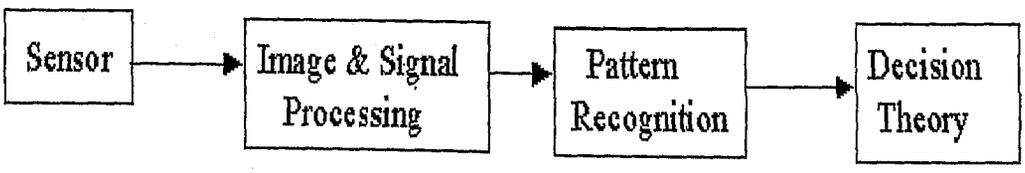


Figure 2.1: Biometrics systems general procedure with four processing

The sensor, and image and signal processing procedures are almost exactly the same for most of the security system. Major differences are mainly in the type of sensor work and the material used to develop it.

Image and signal processing are procedures for noise reduction and image enhancement in order to make it readable by pattern recognition procedure. Pattern recognition is where the classification procedure for the identification or verification occurs. Due to this reason, this chapter is devoted to explain this very important procedure. The last stage, which is a decision-making procedure that decides whether to reject or accept the identity provided to the system. The techniques that will be discussed are:

1. Comparative performance of Classifications Methods for Fingerprint.
2. Fuzzy neural network fingerprint verification system.
3. Hidden Markov Model Fingerprint Classifier.

2.2 Comparative performance of Classifications Methods for Fingerprint

This method categorizes its classification system component as below.

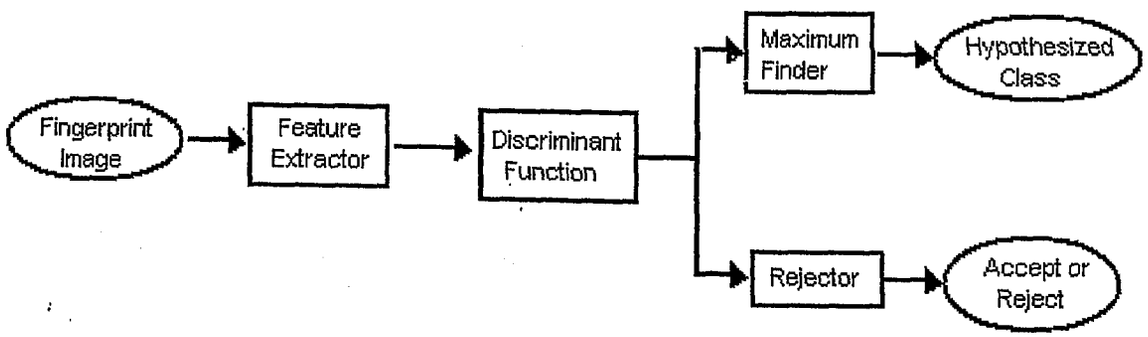


Figure 2.2: Component of Classification System for Comparative performance

The classifier can be divided into four categories, (bearing in mind that the category names are somewhat arbitrary and that some classifiers have attributes of more than one category), namely, Ad-Hoc, Normal Parametric, Nearest Neighbor, and Neural Net.

For each discriminant function, a result of hypothetical class region is provided in the diagram in two-dimensional feature space. However, discriminant that has an adjustable parameter for example a number of hidden nodes, more than one diagram are provided. These diagrams show the hypothesized classes which each one is made using a square array of 512 by 512 points with (0,0) at the center and with the extent large enough to contain all the training feature vectors. The result in this section belongs to the same experiments.

2.2.1 Ad Hoc Classifier

Ad-Hoc classifier consists of the simple Euclidean Minimum Distance (EMD) and the more advanced Quadratic Minimum Distance (QDM) classifiers.

Euclidean Minimum Distance (EMD)

This is perhaps one of the simplest classifiers that one can design. The discriminant function is of the form [23]:

$$D_i(x) = d^2(x, m_i) \tag{2.1}$$

where $D_i(x)$ is the i^{th} discriminant function ($1 \le i \le N, x \in R^n$), N is the number of classes and $d^2(x, y)$ is the square Euclidean distance between x and y ($x, y \in R^n$). Whereas m_i is an estimate of mean feature (μ_i) vector for class i ($1 \le i \le N$)($\mu_i \in R^n$) and R^n is the set of all n-tuples of real numbers or "feature space".

This means classifying an unknown to the highest-valued discriminant function is equivalent to using the class label of the estimated class-mean that is closest to the unknown in the sense of squared Euclidean distance. The resulting hypothetical class

regions are convex polygons. Figure 2.3 shows the class regions when only two features are used. The estimated class mean vectors, $\mathbf{m}_i$ are marked with plus sign.

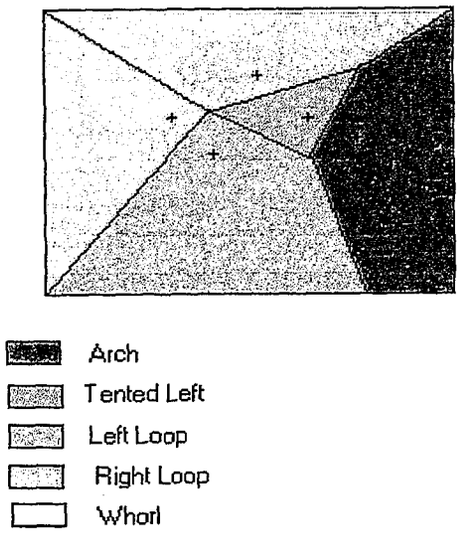


Figure 2.3: EMD class regions. Estimated class means are marked

Quadratic Minimum Distance (QMD)

In this classifier, the training examples of each class i are used to produce an estimate S_i of the class covariance matrix, as well as an estimate m_i of the class mean vector. Then the discriminant function is given by [23]:

$$D_i(x)=(x-m_i)^T S_i^{-1}(x-m_i) \tag{2.2}$$

where $D_i(x)$, m_i is the as the same definition as given in the previous equation, x is a "feature vector" representing a fingerprint $(x \in R^n)$ and S_i represents an estimate of covariance matrix for class i.

The equation also can be thought of as a form of intermediate between EMD and the Normal (NRML) classifier, which will be discussed in the next section. Figure 2.4 shows the resulting class region where boundaries are quadratic in nature.

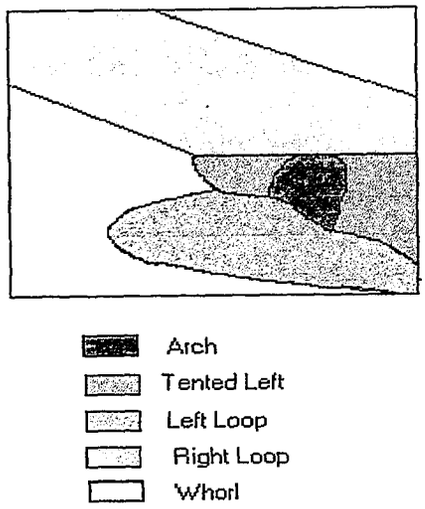


Figure 2.4: QMD class regions

2.2.2 Normal (NRML): A Parametric Classifier

This classifier is based on parametric density estimation that presupposes that a multivariate normal distribution for each class of fingerprints.

Given a particular loss function, $\lambda(i \mid j)$, the optimal or "Bayesian" classifier is the one that minimizes the expected loss. Suppose the "symmetric" loss function is used[27]:

$$\lambda(i \mid j) = \begin{cases} 0 & i = j \\ 1 & \text{otherwise} \end{cases} \qquad (2.3)$$

where $\lambda(i \mid j)$ is a loss incurred by classifying to i a print that is of class $j (1 \le i, j \le N)$.

This means that correct classifications produce no losses and that all kinds of incorrect classifications produce equal loss values of unity. In this case, the Bayesian classifier is the one that classifies each unknown **x** to i for which the posteriori probability is highest.

The hypothetical class regions for the Normal Classifier are shown in Figure 2.5. Similarly to QMD classification, their boundaries are also quadratic. However, the arch region has become much smaller and the tented arch region has disappeared completely.

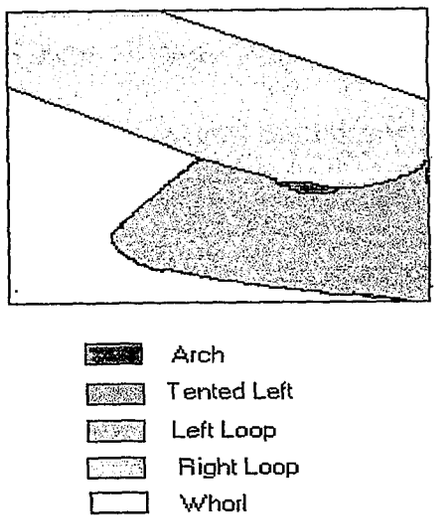


Figure 2.5: NRML class region

2.2.3 Nearest Neighbor Classifiers

Single Nearest Neighbor (1-NN)

This classifier can be thought of as a generalization of EMD. Instead of using just m_i, the estimated mean vector for class i, as a single prototype for the class (as EMD does), the 1-NN classifier uses all of the class-i training examples as a prototype for the class. The hypothetical class produced by 1-NN for unknown vector is simply the class of the closest prototype. According to Cover and Hart [74] this classifier is intuitively appealing as it has good asymptotic behavior that its large sample probability or error is bounded above by twice the Bayes (i.e. minimum possible) probability or error. Discriminant function of the following form may be used to implement the 1-NN classifier:

$$D_i(x) = -\min_{1 \le j \le M_i} d^2\left(x, x_j^i\right) \tag{2.4}$$

The determinant function in this equation indicates a distance call Mahalanobis distances. The smaller the Mahalanobis distance, the closer the case is to the group centroid and the more likely it is to be classed as belonging to that group. However, the result of 1-NN can be improved by a generalized form of this particular classifier that will be discussed in the next section. Figure 2.6 shows the class regions. Each region is the union of many convex polygons containing a single prototype of the class which as shown in the diagram is a very complicated polygon, not necessarily convex or even connected.

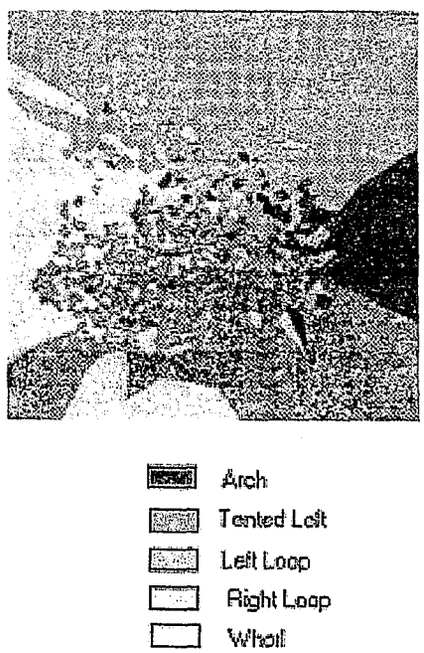


Figure 2.6: 1-NN class regions

Weighted Several Nearest Neighbours (WSNN)

Due to the drawbacks of the 1-NN it is necessary to consider a generalized classifier which is denoted as "k-NN classifier" so as to obtain much more reliable and accurate results. This classifier finds the k nearest prototypes to the unknown and classifies it to the class that is most abundantly represented among these near

neighbors. For practical purposes, "k-NN classifier" finds which value produces the highest test scores that optimize k.

This classifier finds the closest prototype to the unknown, then defines the *neighboring* prototypes to be those whose squared Euclidean distance from the unknown is less than α times the squared-distance of the nearest prototype, where α is a constant. The number of *votes* received by class *i* is divided by the square root of the sum of squared-distances of class *i* near neighbors form the unknown, so as to penalize the far away neighbors. This modified classifier is called Weighted Several Nearest Neighbor (WSNN) classifier. Surprisingly, in this case it always produces lower scores for $k > 1$ than for $k = 1$. However, the overall results are better than the 1-NN classifier. The range α is between 0 and 100. In Figure 2.7 shows the WSNN class regions resulting for $\alpha = 3$, 10, 50, and 90.

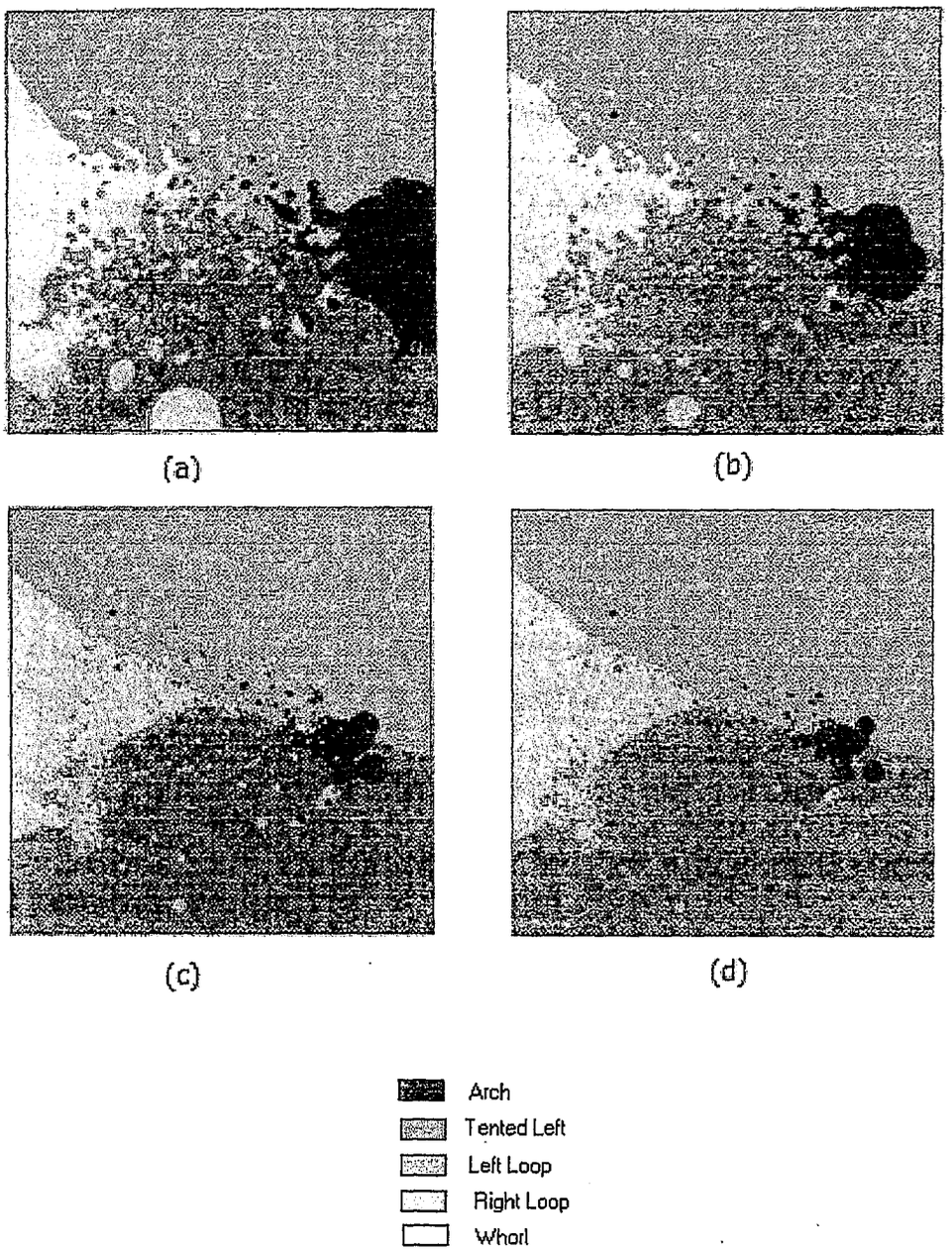


Figure 2.7: WSNN class region, (a) $\alpha = 3$, (b) $\alpha = 10$, (c) $\alpha = 50$, (d) $\alpha = 90$

2.2.4 Neural Net Classifier

Multi-Layer Perceptron (MLP)

This classifier is also known as a feed-forward neural net. MLP is used in three layers (taking the input as a layer). The classifier function is of the form:

$$D_i(x) = f\left(b_i^2 + \sum_{j=1}^{N^1} w_{ij}^2 f\left(b_j^1 + \sum_{k=1}^{N^0} w_{jk}^1 x_k\right)\right) \qquad (2.5)$$

where $f(x) = 1/(1 + e^{-x})$ is equal to sigmoid function and $b_i^{(k)}$ is the bias weight of i^{th} node of k^{th} layer. Whereas $w_{ij}^{(k)}$ represent the weight connection i^{th} node of k^{th} layer to j^{th} node of $(k-1)$ layer ($k=1,2$; $1 \le i \le N^{(k)}$;$1 \le j \le N^{(k-1)}$) and $x = (x_1,...,x_n)^T$ is a feature vector. Note that

$N^{(0)}$ = number of input nodes = n = number of features

$N^{(1)}$ = number of hidden nodes

$N^{(2)}$ = number of output nodes = N =number of classes

For the training of the weights of this network, the optimization algorithm is to minimize the mean-squared-error procedure over the training set. The training example consists of the strings of 1's and 0's. For example, if a training feature vector is of class 2, then its target vector of discriminant values is set to (0, 1, 0, 0, 0). It is more feasible to minimize this kind of an error function than to attempt to directly minimize the number of incorrectly classified training examples, since the latter will take only relatively few values and is a discontinuous step function. In fact, the error function that is minimized is defined to contain an additional regularization term in addition to the mean-squared discriminant error. This term consists of a factor times the average of the squares of the weights.

The motivation for including the regularization term is that the resulting weights will be somewhat controlled in magnitude, and that this will increase the generalization ability of the network. The goal of training is to produce a network that will give clearer identification that is accurately classify new examples which were not part of the training set; even if the training algorithm produces weights that result in

perfect classification of the training examples, there is no guarantee that the network will generalize well.

Network of MLP type are the most commonly used neural network today, and are usually trained using back propagation algorithm [21]. However, in this paper a scaled conjugate gradient training method is used [41]. The procedure trains network much faster than back propagation. Figure 2.8 shows MLP class regions resulting from the use of 1,2,24, and 80 hidden nodes.

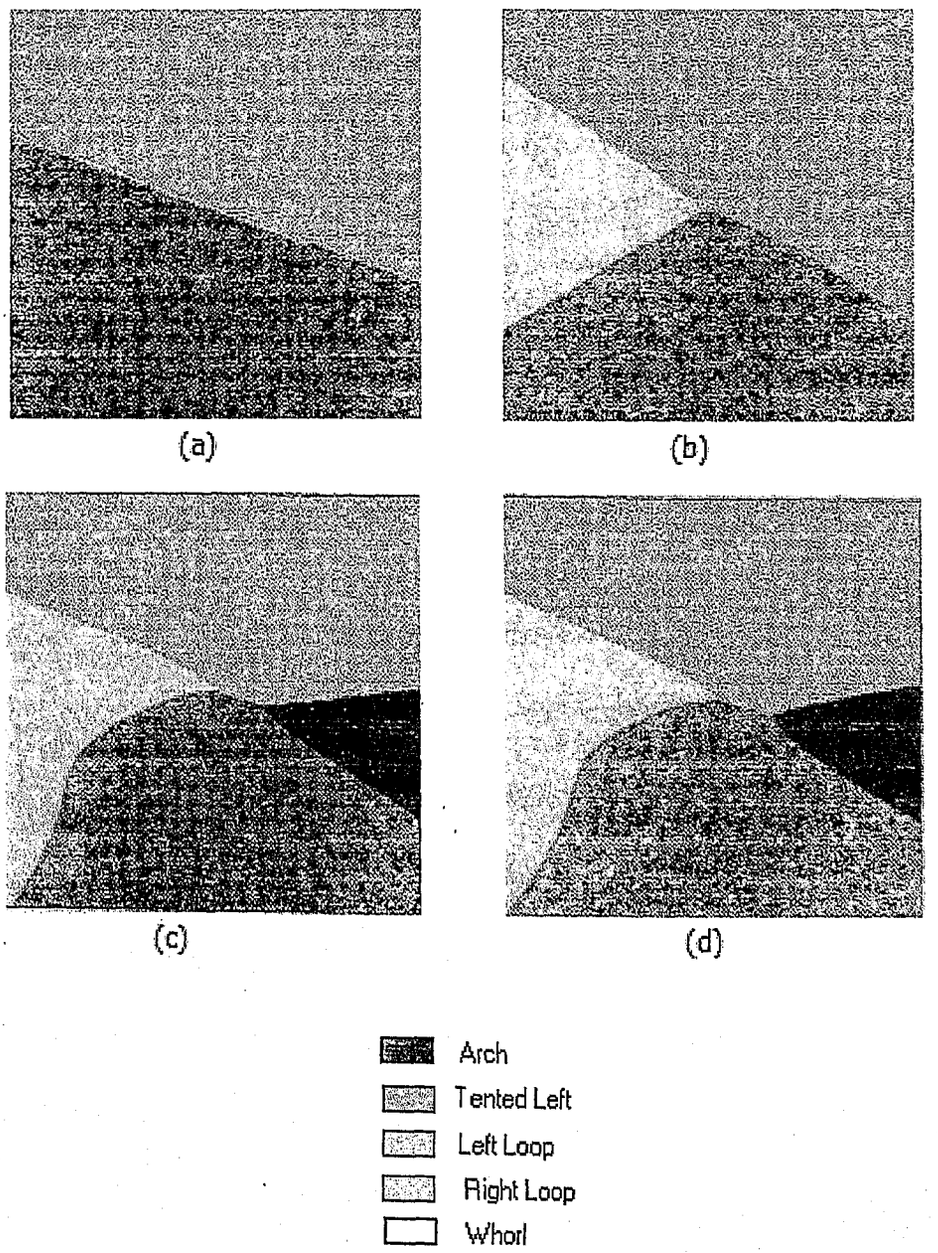


Figure 2.8: MLP class regions for (a) 1 hidden node, (b) 2 hidden nodes, (c) 24 hidden nodes, and (d) 80 hidden nodes

2.2.5 Merits and Demerits

Table 2.1 shows the comparison for each classifier type. The comparison is based on the lowest error rates that were obtained for each of several numbers of features.

Table 2.1: Error percentages for classifiers and number of features

Classifier	Number of features						
	16	32	48	64	80	96	112
EMD	26.9	26.6	26.4	26.3	26.2	26.3	26.3
QMD	12.8	15.6	18.0	20.1	20.7	21.6	23.0
NRML	13.5	12.8	16.8	18.1	19.7	20.7	23.0
1-NN	10.7	9.6	9.7	9.3	9.1	9.0	9.3
WSNN	10.3	9.3	9.1	9.1	8.9	8.9	9.0
MLP	9.1	8.8	8.6	8.2	8.2	8.4	8.5

From the result it is clear that the optimal number features is not the same for all of these classifier types. The EMD shown that it performance is almost stable for any numbers of features. While QMD showing that it performance is decreasing with the increase of features. Among all the classifiers, MLP performance is the most excellent. The increase of features resulting in decrease of error percentage which indicates that the MLP performance enhance with the increase of features.

Pattern-level classification of fingerprints is a nontrivial task even for humans; it appears to be very difficult to automate, even using a massively parallel computer. The test results indicate that the accuracy differences are fairly small when certain reasonable classifiers are compared. Production of special-purpose hardware is probably required if fingerprint classifiers are to be implemented at the lowest cost and with the highest possible classification speed.

2.3 Fuzzy neural network fingerprint verification system

An article written by C. Quek, K.B. Tan and V.K.Sagar From Nanyang Technology University, Singapore [10] describing the system was published by Elsevier Science Ltd in October 2000. In the introduction, they stated that conventional identification methods are based on memory data such as identification numbers and passwords. The other methods are based on possession such as personal seals and magnetic cards. These two methods do not offered a high degree of security. The use of a biometrics identification method is an effective technique in improving identification confidentiality. Fingerprint recognition is a representative of such technique.

Automated fingerprint identification system (AFIS) technology has been developed, refined and accepted by demanding law enforcement agents over the past two decades. However the use of fuzzy neural network for fingerprint verification is still at its infancy stage. There are reasons to believe that the fuzzy neural network approaches may remarkably well suit for fingerprint problems. Firstly, fingerprints form a very specific class of pattern with very different statistical characteristics. Thus, the corresponding pattern recognition problems seem to be well confined and constrained. It is even more so than other pattern recognition problems, such as the recognition of signatures where fuzzy neural networks have already been applied with reasonable success [80].

Secondly, fuzzy neural network as hybrid intelligent systems possess the advantages to both neural networks and fuzzy rule-based systems. It is particularly

powerful in handling complex, non-linear and imprecise problems such as fingerprint verification. Their use avoids some of the pitfalls inherent in other more conventional approaches. The comparison of two fingerprint images can be matched on the basis of minutiae or ridge characteristics. Minutiae are essentially interruptions of the normal flows of the ridges, such as ridge ending, bifurcation, etc. However, the minutiae-based approach has two obvious weaknesses namely:

1. Sensitivity to noise (especially with inked fingerprint images) and
2. Computationally expensive (essentially a graph matching problem)

Thirdly, since fuzzy neural networks are robust, adaptive and trainable from examples, fingerprints images can include different sources of deformation and noise such as finger skin condition and the positioning of the finger during the image acquisition phase. Furthermore, fuzzy neural networks can be tailored and trained differently to fit the respond time requirement in forensic applications. Efficiency is an important aspect in forensic applications, which require rapid searches through large databases for fingerprint images. In this respect a database of training and testing examples with various skin conditions has been constructed to test the robustness of neural algorithm pseudo outer product based fuzzy neural network (POPFNN).

2.3.1 Data acquisition

The database is relatively small as the principal requirement is for fingerprint verification using a fuzzy neural network. Only five fingerprint images for each person are required to train each type of POPFNN. These fingerprint images are used as test sample to verify against the fingerprint images fed into the fuzzy neural network. Altogether, four sets of the fingerprint images (under different conditions) were

collected from four different persons, to form the fingerprint database. The fingerprint training samples and the test set are collected separately.

2.3.2 Fingerprint acquisition technique

The technique chosen for capturing the fingerprint images is based on the use of fingerprinting ink and image scanner. The fingerprint images are scanned at 400 dpi (dots-per-inch) and stored in Windows-standard bitmap files in 256 grey levels. The thumbprints acquired are either of the left hand or the right hand; whichever is the non-master hand.

Recent advances in computing and digital imaging technology have led to the introduction of new AFIS methodologies using plain impression (referred to as 'plane' or 'flat' impressions) fingerprint images as the basis for identification [19]. The advantage of using plain impression fingerprint images is that it minimizes capture time and storage requirements. A plain impression print can be less than 50% of the area of the equivalent rolled print, providing significantly less data needed for processing.

2.3.3 Fingerprint pre-processing techniques

In the fingerprint pre-processing stage, the quality of input fingerprint images is improved using the different image enhancement techniques and making the necessary transformation to the fingerprint images so as to enable the extraction of suitable features for processing in the fuzzy neural network. They are namely: segmentation and alignment [7]; histogram equalization [67]; median filtering [25]; threshold [73]; and skeletonisation [81].

2.3.4 Pseudo outer product based fuzzy neural network (POPFNN)

Pseudo outer product based fuzzy neural network (POPFNN) is a hybrid system. It possesses the advantages of both neural networks, and fuzzy systems. It is constructed using numerical information and the adjustments of such networks can be achieved through neural network techniques. The initial set of parameters and structures are constructed from a set of training data. Functions performed by each layer of POPFNN correspond to the fuzzy interference steps of the truth-value restriction method [53]. POPFNN has five layer structures. It must be noted that the input and output data of POPFNN are non-fuzzy data. The fuzzification of the input data and the defuzzification of the output data are accomplished automatically in POPFNN.

The learning process in POPFNN consists of three phases: self-organization; POP learning; and supervised learning. A self-organizing algorithm is employed in the first phase to initialize the membership functions of both the input and output variables by determining their centroids and widths. In the second phase, the Pseudo outer product learning algorithm is performed to identify the fuzzy rules that are supported by the set of training data through a set of criteria to expunge spurious ones. Consequently the chances of generating irrelevant and misinforming fuzzy rules are greatly diminished. Furthermore, the algorithm also modifies the network's structure. Some of the modifications include:

1. *Invalid input variables:* means the deletion of input linguistic variables that have little or no relation to the output linguistic variables.

2. *Irrelevant output variables:* means the marking of output linguistic variables that have little or no relation with the input linguistic variables.

2.3.5 Experimental result and analysis

Fingerprint verification is basically a two-class pattern classification problem. After feature extraction, a feature vector, which consists of several features, is obtained and used to compare against a reference set of feature vectors. A decision is then made to classify the vector, that is, the fingerprint is grouped into one of two classes, either *authentic* or *spurious*. Ideally, the class of authentic fingerprints and the class of spurious fingerprints should be well separated. However, this is not the case in practice. There is always a certain amount of overlap between the two classes in the feature space.

In the fingerprint verification system, POPFNN is employed to accomplish the comparison process. As hybrid intelligent systems, fuzzy neural networks possess the advantages of both neural networks and fuzzy rule-based systems and are particularly powerful in handling complex, non-linear and imprecise problems such as fingerprint verification. The desired output for an authentic fingerprint is value 1.0, and the desired output for a spurious one is value 0.0.

The training sets used in the experiment are described in this section, and they are divided into three different classes:

1. POPFNN trained with both authentic and spurious fingerprints;
2. POPFNN trained with authentic fingerprints under normal conditions as well as fingerprints under the influence of external conditions; and
3. POPFNN trained with only authentic fingerprints under normal conditions.

For every class, a model is trained by using only one feature, and all the features train the others. Altogether, there are 10 types of POPFNN, which were designed and implemented to explore the effect of different features on the performance of the networks. These 10 types of POPFNN are listed in Table 2.2.

Table 2.2: The classification of POPFNN designed for the experiments

Type of POPFNN	Total no. training of samples	Type of training samples authentication					No of features taken
		Normal	Washed hands	Taken shower	Held pineapple	Defect	
Model I	10	5	0	0	0	5	ONE
Model II	10	5	0	0	0	5	ALL
Model III	10	5	0	0	5	0	ONE
Model IV	10	5	0	0	5	0	ALL
Model V	10	5	0	5	0	0	ONE
Model VI	10	5	0	5	0	0	ALL
Model VII	10	5	5	0	0	0	ONE
Model VIII	10	5	5	0	0	0	ALL
Model IX	5	5	0	0	0	0	ONE
Model X	5	5	0	0	0	0	ALL

Among the 10 types of POPFNN, the tenth type is of utmost importance as a positive result will point to the fact that the fuzzy neural network is able to differentiate the authentic fingerprints from the spurious ones, including fingerprints that are exposed to external conditions. It will also show that a training set of five samples is adequate to train the POPFNN.

For Models III – VII POPFNN, the experiment is carried out to assess the robustness of the system on each variant type of authentic fingerprints. As for Models I and II POPFNN, the idea is to train the system with five authentic fingerprints and five spurious fingerprints which are of the same class as the authentic ones and the results can be used for comparison purposes with Models IX and X POPFNN.

Similar to any other neural network system, the performance of the system can be also rated according to the rejection rate and acceptance rate as illustrated in Figure 2.9.

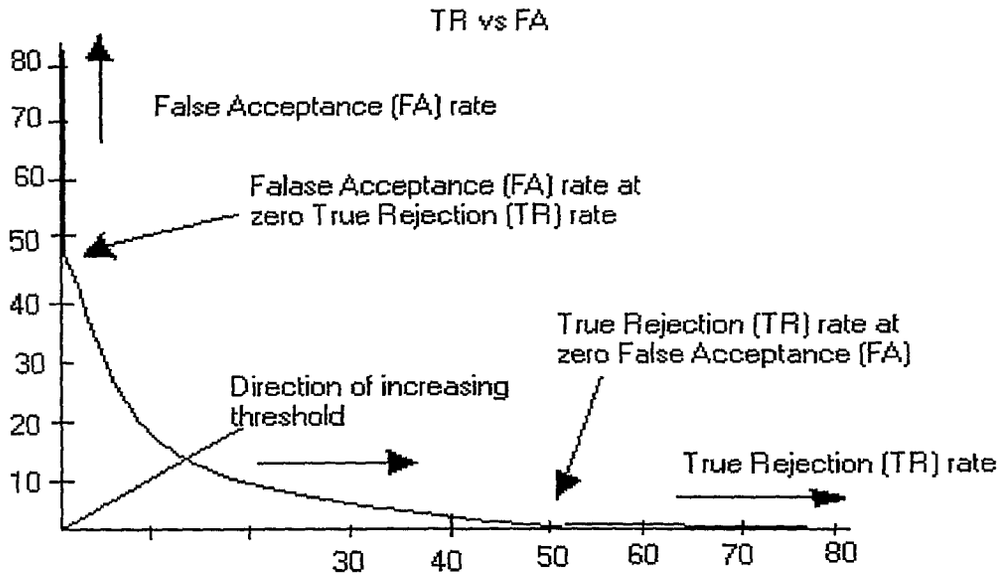


Figure 2.9: Sample receiver operating characteristics (ROC) curve

This error rate gives an indication of the overall separability of authentic fingerprints from spurious ones. The second and third points are the true rejection rate at zero false acceptance and false acceptance rate at zero true rejection.

A fingerprint is classified as authentic if its network is above the threshold value. Otherwise, it is classified as a spurious fingerprint. For example, the rules identified by type IX POPFNN, which was presented with the ridge ending feature of an individual, are describe as follows:

Rule 1: if x_0 is large and x_1 is large, the 'fingerprint is authentic' is 1.000000 true;

Rule 2: if x_0 is median and x_1 is median, then 'fingerprint is authentic' is 0.959801 true;

Rule 3: if x_0 is small and x_1 is small, the 'fingerprint is authentic' is 0.618143 true.

2.3.6 Merits and Demerits

The POPFNN-driven fingerprint verification system (POPFNN-FVS) was able to yield good results when tested against authentic and spurious fingerprints. This shows that the number of fingerprints requested can be kept at a reasonably small number and it will not discourage potential users who would want to implement such a system. However, directional codes feature method, did not yield good results when different features were combined as inputs for the system. Although the sample used to train the network is only five fingerprint samples, the overall results showed that POPFNN fingerprint system is able to yield good results when tested against authentic and defect fingerprints. Furthermore, POPFNN is robust to external conditions on the fingerprints and thus making it even more favorable.

2.4 Hidden Markov Model Fingerprint Classifier

An article on this classifier was published in IEEE in 1998 [71]. Andrew Senior of IBM Research Centre stated that Hidden Markov models are a form of stochastic finite state automation well suited for pattern recognition and has been successfully applied to speech recognition. They are appropriate to the problem posed here because of their ability to classify patterns based on a large unknown quantity of features with have certain types of underlying structure, especially if that structure results in stationary of the feature distributions over some spatial or temporal period. This is found in fingerprints, where ridge orientations, spacing, and curvatures are for the most part only slowly varying.

Typical Hidden Markov models (HMM) [49] are one-dimensional structures suitable for analyzing temporal data. Here, the data are two dimensional, but process of feature extraction can also be described as one-dimensional array of one-dimensional process. Each row is a one-dimensional process, and the ensemble of rows is itself a one-dimensional process. Thus we can apply a two dimensional HMM which consists of a nesting of row models within whole-print models as shown in Figure 2.10.

The system describes here has been designed to operate on both rolled and 'dab' fingerprints, where some of the structural information used by other systems (such as the delta position) may not be available in the fingerprint image. The system described has been tested on the National Institute of Standards and Technology version 4 [9] (NIST-4) databases of fingerprint images.

This successful system deals with fingerprint images stored as arrays of grey levels and obtained with scanner or camera device, either from an inked fingerprint on paper or as a 'live-scan' directly from the finger. In most of the reported work, the NIST-4 database of rolled fingerprint images has been used, since this provides a large number (4000) of fingerprints with associated class labels.

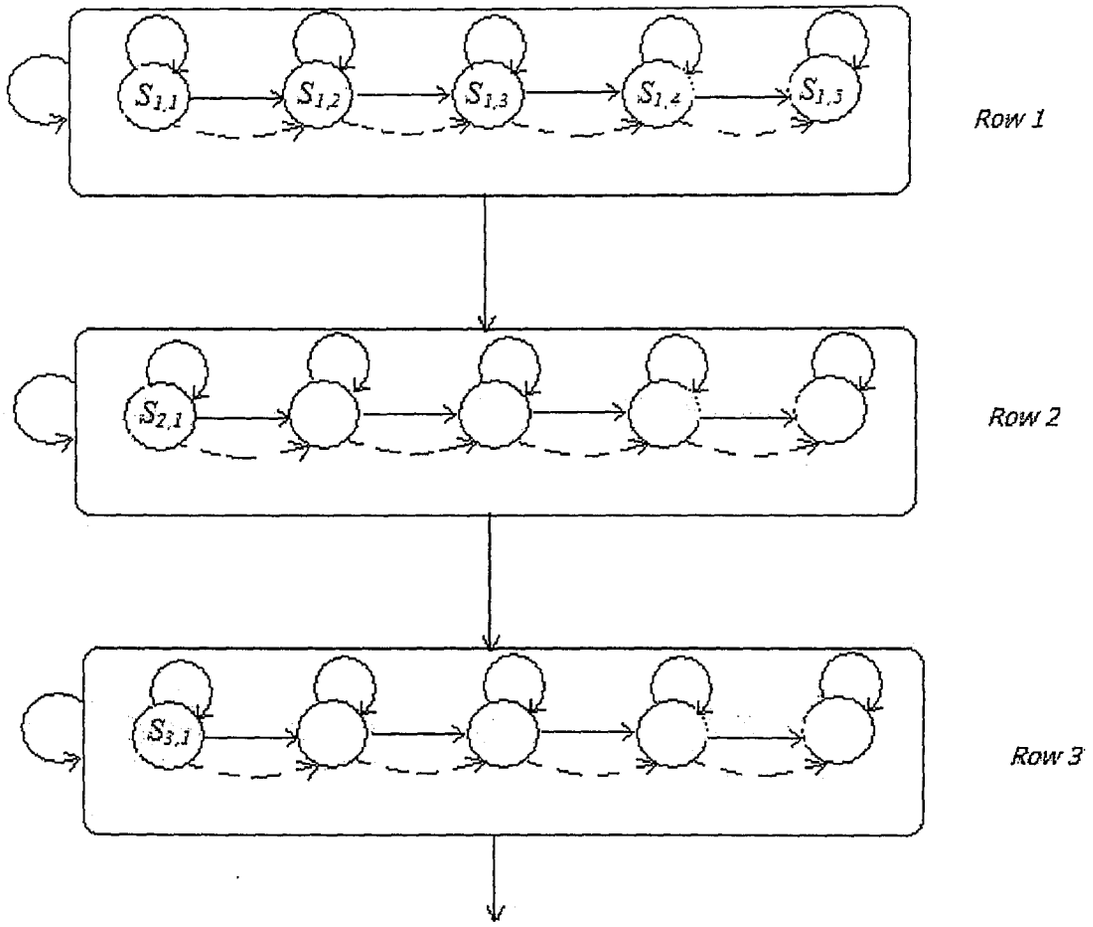


Figure 2.10: A schematic of the two-dimensional structure of the HMM, showing three row models of five states each forming a global model

The features provided to the recognizer are based on the characteristics of the intersections of ridges with a set of lines that are laid across the fingerprint image. To find the ridge locations, a number of image processing techniques are used. The basic steps are:

1. Smoothing;
2. Finding the predominant direction in each of an array of blocks covering the image;
3. Segmenting the image into the area of the print (foreground) and the unwanted background, based on the strength of directionality found in each block;
4. Applying directional filters to highlight the ridges and detect pixels that are parts of ridges; and

5. Thinning the ridge image so that each ridge is left represented by an eight-connected, one-pixel-wide line termed the skeleton.

Given the skeleton image of the ridges, parallel lines are laid across the image each one followed in turn. For each intersection of a line with a ridge, a feature is generated. Each feature consists of a number of measurements:

1. The distance since the last intersection.
2. The angle of intersection.
3. The change in angle since the last intersection.
4. The curvature of ridge at the intersection.

2.4.1 Row Modeling

To simplify the analysis of the model, first consider a row modeling a single row of fingerprint data. Each row model M_j, consists of a number of states, which model the small, stationary regions in a row. Any row R_i, is assumed to have been generated by the row automation transiting from state to state, producing the frames in the observed order at each transition, with the k^{th} state being S_{ijk}. The state transitions are controlled by probabilities $P(S_{ijk}|S_{ijk-1})$ trained with certain constraints; the state must monotonically increase $S_{ijk} > S_{ijk'}$, *for* $k > k'$ and it may be possible to skip state at the edge. It is due to the nature of the printing process, especially for dabs it is to be expected that edge regions of the fingerprint will be missing but the central regions will always be present. This effectively constrains the initial state distribution $P(S_{ijo})$.

The frames are modeled with mixtures of diagonal covariance, multivariate Gaussian distributions. Thus for any frame, it is possible to calculate the probability.

The efficiency measure permits the evaluation of rejection and backing-off strategies. It is clear that accuracy can be traded off for efficiency – searching more than just the top one class will give higher accuracy but lower efficiency.

Previous classification works have quoted error rates that would be unacceptable in real-world systems. If a reliable measure of confidence in the classifier's answer is available, it is possible to devise methods to adjust the reliance on the classifier answer to reduce the number of errors made. Some classifiers have used a rejection mechanism, which improves the accuracy at the cost of not pruning the search with those prints that are rejected.

2.4.2 Results

The test set was a random sample of 542 prints from the 4000 available. The class labels given by the database were used, but since the efficiency is hardly affected, the classifier was only trained to distinguish four classes, treating arch and tented arch as identical.

Table 2.3: Errors rates, testing 2DHMM classifier.

Experiment	Error (%)
Vertical (v) features only	23.2
v and h classifiers	18.4
v and h with priors and weighting	11.8
ditto with decision tree	10.0

A model with more parameters (8 states, 8 rows and a pool of 200 Gaussians) achieves an error rate of 8.4% on the same test but takes more time to train for classification. The classifier gave a 9.9% error rate, though it had been trained on a different training set, with more but perhaps less well matched fingerprint image [72].

Table 2.4: Efficiencies for the 6/9/80 models with rejection thresholds set to give less than 1% error rate

Experiment	Error (%)	Efficiency (%)
v and *h*, priors and weighting	0.6	2.0
Ditto with decision tree	0.8	2.3

2.4.3 Classifier efficiency

Since the purpose of a fingerprint classifier is to partition large fingerprint databases, in addition to the classification accuracy, the proportion of classifications that give the correct class, the classification efficiency must also be considered. The classification efficiency can be considered as a measure of reduction of search space. In practice, the proportion of the database to be search will vary with each query, so over a test set, the average efficiency can be calculated as:

$$\frac{\text{Number of matches required with no classifier}}{\text{Number of matches required when classifier is used}} \tag{2.6}$$

where an exhaustive 1: N matches against a database of N prints are counted as N matches.

2.4.4 Merits and Demerits

For automated fingerprint identification (AFI) system, classification playing a bigger role. The existing classifier accuracies, however, fall short of what required in making a significant contribution to an AFI system. The Markov method for comparing the efficiencies of different classification scheme has achieved a good degree of accuracy. The method can achieve filtering of a factor 2.3 with and error rate of only 0.8%. However, the only demerit of the system is when more parameters are

introduced, the error rate will reduce but the method takes more time for training and classification.

2.5 Summary

Some of the available techniques for fingerprint security system is discussed in this literature review section. The first review is on the performance of classification methods for fingerprint. The next review is on Fuzzy Neural Network fingerprint verification system. The last review is on Hidden Markov Model Fingerprint Classifier. From the review, a conclusion can be made about the best method for the classification or verification system. The system that uses Artificial Neural Network (ANN) method is more reliable, accurate and can provide good result compare to other methods. Due to the above reason, ANN is proposed to be used in the thesis as the means for classification and verification. The design and development of the proposed method will be discussed in the next chapter.

CHAPTER 3: FINGERPRINT SECURITY SYSTEM DESIGN

3.1 Introduction

Biometrics deal with identification of individuals. The identification is based on their biological or behavioral characteristics. The need for security becomes more and more important in our modern life especially with the enhancement of automation systems. Quite often the questions of "should this person be given access to a secure system?", "Does this person have authorization to perform a given transaction?" arises in many situations. All these questions are dealing with the same security issue which is how to correctly identify human beings.

To solve such a security problems there are two popular ways introduced; one is related to "something that you have" and the other way depends on "something that you know". The examples of "Something that we have" is identification cards, credit card, physical key, etc., whereas "Something that we know" is password, PIN, etc. Both methods give the authority to most users, other than end users. This implies that if a user gets the password or other way of identification, he will get the authority to enter the area or to access the secure place. Under such security system, people have to keep various cards and remember a lot of passwords. By losing a card or forgetting a password a user might be in a great trouble. If this problem occurs, once the unauthorized person in control of the identifying possession, that person could abuse the privileges of the authorized user.

Furthermore, to use knowledge as identification mechanism is not very reliable since it is difficult to remember the passwords/PIN; while easily recallable passwords/PIN such as pet's name, spouse's birthday could be easily guessed by the adversaries. As a result, these techniques cannot distinguish between an authorized person and an impostor who acquires knowledge/possession, enabling the access privileges of the authorized person.

In order to reduce the problem of identification to become the problem of identifying physical characteristics of the person, many researchers have been working on various ways on improving security and biometrics approach appears to be one of the most promising ones. Biometrics is a technology that uses human being's unique physical or behavioral features to identify or verify persons. It relies on "something that you are born with" to make personal identification and therefore can inherently differentiate between an authorized person and a fraudulent impostor [20]. Because one's unique characteristics cannot be stolen, forgotten, duplicated, shared or observed. Biometrics based security system is nearly impossible to fraud.

3.1.1 Taxonomy of biometrics system

According to David Zhang [18] biometrics system can be classified into *application type* and *technology type*.

Application Type

Although biometrics technology is associated to security services but it is also shown to be very effective in few other applications. In general, application type biometrics-based system can be divided into four categories as shown in Figure 3.1

- *Personal Authentication*: We may use biometrics technology to identify individuals.
- *Medical Diagnosis*: Tongue, color face, beat of heart and other aspects of our body can be also used as biometrics features for medical diagnosis.
- *Future Expectation*: Do you have the experience of looking at a handsome face at the first glance and thinking of "Oh, he must be a nice guy!". Yes, our outside does represent our inside to a certain degree. Egypt and China have some specialists on this biometrics application. By looking at one's palm, the specialists can tell the personality as well as the future direction of a person.
- *Ethnology exploration*: Measuring body characteristics can be also used to decide one's ethics. We may use this biometrics technology to monitor the population shifting among different areas.

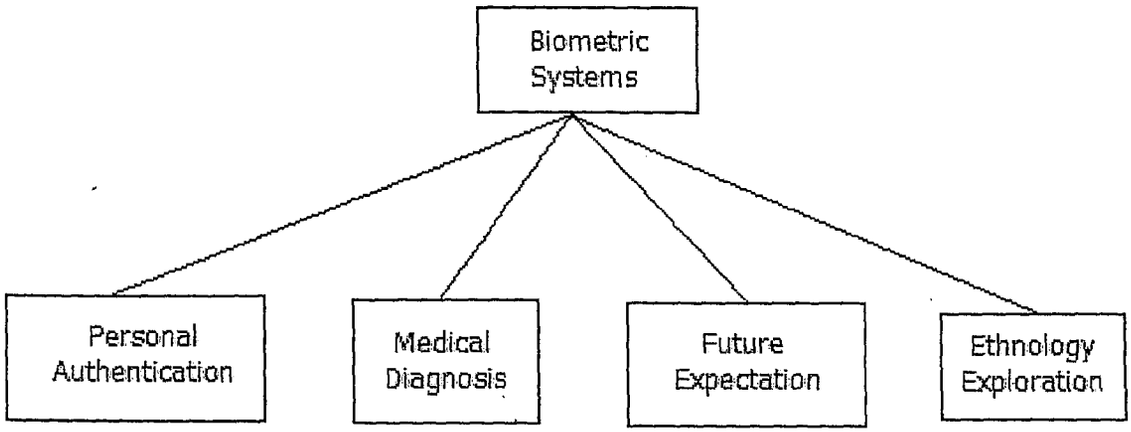


Figure 3.1: Taxonomy by application type

Technology Type

Biometrics technology type systems are categorized in terms of the employed physical or behavioral characteristics, as shown in Figure 3.2. Since in this technology type we will be emphasizing on the application of personal authentication, there are

two main parts in our body dealing with physical characteristics, head (face, iris, and others) and hand (fingerprint, palm print and others). Behavioral characteristics cover signature, voice and others (gesture, gait, keystroke, etc).

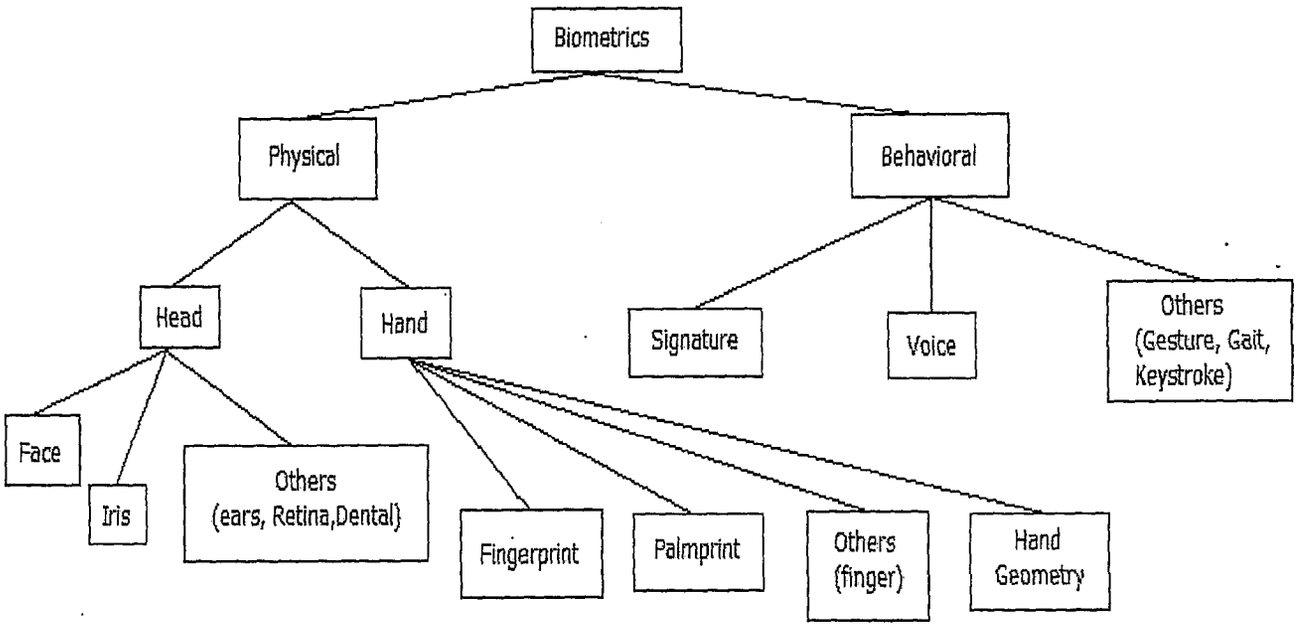


Figure 3.2: Taxonomy by technology type

3.1.2 Concept of biometrics for security system

Biometrics system uses human faces, fingerprints, voice, iris etc as a medium to verify or identify a person. There are three generally accepted methods for performing human verification or identification. These are based on:

- Something the user **knows** (such as password)
- Something the user **possesses** (such as a card/badge, called tokens)
- Something the user **is innately** characterized with (a physical characteristic)

The widely accepted method such as password does not guarantee a high security system since passwords can be compromised in many ways. They can be

forgotten, written down, guessed, stolen, "cracked", or shared. Similarly for tokens, like a telephone card or credit card, can be lost, forgotten, stolen, given away, or duplicated. However, biometrics verification or identification can be accomplished by measurement of a unique biological or behavioral feature of the user to verify identity through automated means.

An effective algorithm is used to determine if an image sample *matches* another image sample. Generally, the result of this comparison is a *score,* indicating the degree to which a match exists. The comparison is performed using the captured image and previously stored images. This score is then captured to a pre-set threshold to determine whether or not to declare a match. There are four types of image matching, namely Verification, Identification, Search or Retrieval, and Classification.

Verification is one-to-one (1:1) matching of a single sample set (biometrics identifier record) against another. Generally, the first sample is newly captured and the second is the enrolled identifier on file for a particular subject. In a user authentication environment, a score exceeding the threshold would return a *match*, resulting in the authentication of the user. A score below the threshold would return a *no-match*, resulting in the denial of access.

Identification is a one-to-many (1:N) matching of a single sample set against a database of samples, with no declared identity required. The single image is generally the newly captured sample and the database contains all previously enrolled samples. Scores are generated for each comparison, and an algorithm is used to determine the matching record, if any. Generally, the result will yield the highest score exceeding the

threshold in identification. The FBI Fingerprint Recognition System is using the identification method since they are deal with large databases.

Search or Retrieval is quite similar to identification, i.e. 1: N matching. However, the result is a set of possible matching images, not a classification. Identification returns a classification, while Search returns multiple matched images. Note that in the first three processes, only one class of image(s) is compared with a set of existing images (i.e. 1:1 or 1:N). *Classification*, on the other hand, is N: 1 or N: N matching.

There are several types of data storage system such as Client-Server and Stand Alone. In Client-Server system during a verification process, the pre-stored image sample is retrieved from the database in a server, based on a unique subject identifier (such as a User ID). In the identification problem, the image database is in the server. On the other hand, in Stand Alone system the pre-stored image sample is retrieved from the data card while in the identification problem, the image database is in a computer.

As previously discussed, there are a few methods of image matching system. There are verification, identification, search or retrieval and classification. For our system design and development we decide to use verification method since it does not require a database system such in identification method. The database system will take up a large amount of the computer memory and will slow down the speed of searching for a match. In this thesis the verification method is selected since the user to the restricted area will be limited to a small number of peoples.

3.2 System Design

The design and development of the system will undergo a few stages. The general design of our developed system is as shown in Figure 3.3.

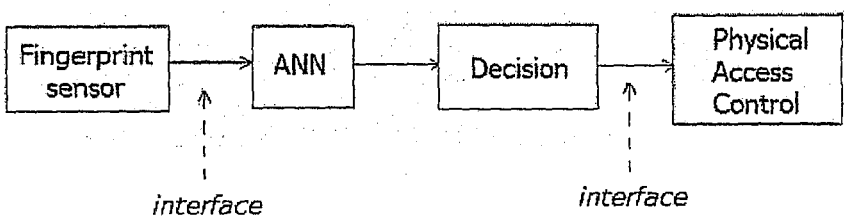


Figure 3.3: General design of the system

As observed in the diagram, there are four operative levels, namely, fingerprint sensor, Artificial Neural Network (ANN), decision and physical access control. In practice there are two levels of processes namely the first level and the second level. The layout of this process is presented in Figure 3.4.

The first level process is the acquisition process. The image acquisition process is the process of obtaining the input image from the fingerprint sensor. The captured fingerprint image, which is the fingerprint of the unknown person to be verified, is also known as a test image. The test image will be scanned in different position, rotation and translation. The image will be then be sent to the ANN, which an image finder is used for training. For the verification process we choose Attrasoft ImageFinder software to be the core of the system. The ANN will be train with all the test images. After that all the train images will be stored in one directory that will be called upon during the verification process. For verification we do not need a database system since it will be just a small stored data and it is a 1:1 recognition process.

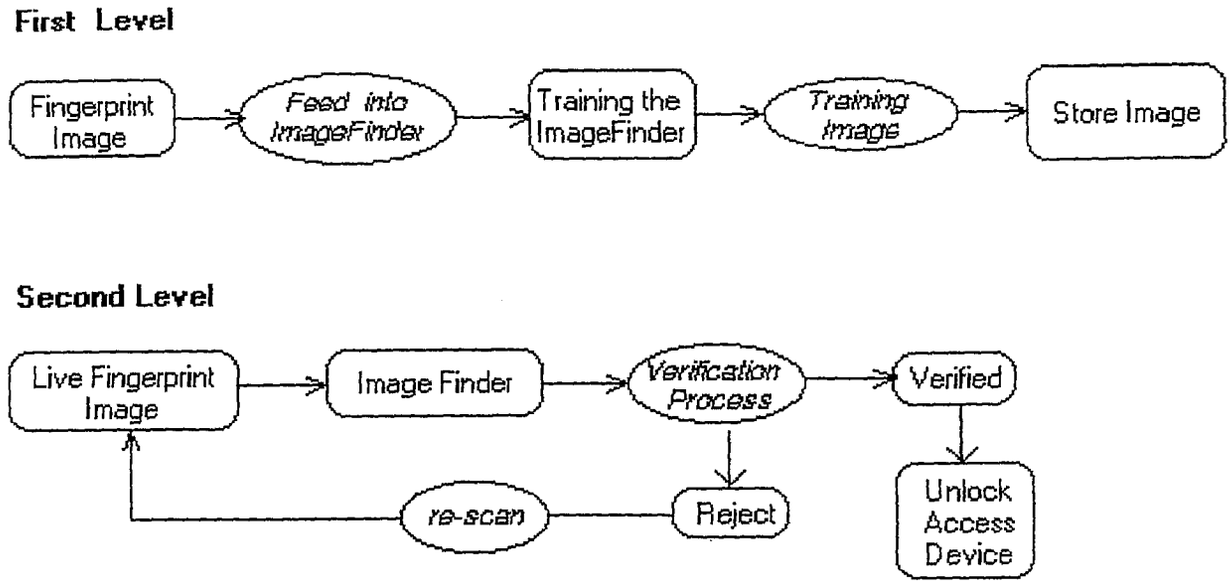


Figure 3.4: The process of the whole system

On the second level of the process, a live person will place his or her finger on the fingerprint sensor. The sensor will scan his or her fingerprint image and the image will be send to the ANN. ANN will do the search in the directory of the train image. If there is any match of the scanned image with the stored image then a verified sign will result and a signal will be sent to the physical device system to unlock the device. If the reject sign is the output then the person will need to re-scan his/her finger.

3.2.1 Sensors Technology

In the following section the discussion of the two most popular sensor technologies is discussed. The two technologies are the optical sensor technology and the capacitance sensor technology. After the discussion of both technologies a table of comparison of the sensors is presented.

3.2.1.1 Optical Sensor

Figure 3.5 showa an optical fingerprint sensor technology (this figure is adapted from a conference paper *"An Overview of Recent Biometrics Technologies"*

given by Alex Kot, [48] from Nanyang Technological University, Singapore and from *Biometrika* website http://www.biometrika.it/eng/wp_fingintro.html) [34].

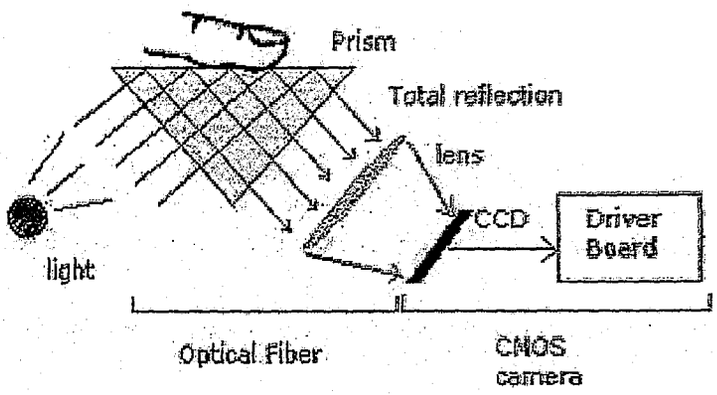


Figure 3.5: Optical sensor technology

The heart of an optical sensor is a charge coupled device (CCD) [32]. A CCD is simply an array of light-sensitive diodes called photosites, which generate an electrical signal in response to light photons. Each photosite records a pixel and collectively, the light and dark pixels form an image of the scanned scene (a finger, for example). An analog-to-digital converter in the scanner system will process the analog electrical signal to generate a digital representation of the image.

The CCD camera takes a picture when a finger is placed on the glass plate. The sensor has its own light source to the ridges of the finger. Typically an array of light- emitting diodes is used as the light source. The CCD system actually generates an inverted image of the finger. The darker areas represent the ridges of the finger and the lighter areas represent the valleys between the ridges.

The sensor processor makes sure that the CCD has captured a clear image by checking the average pixel darkness, or the overall values in a small sample. The scan is rejected if the overall image is too dark or too light. If the image is rejected, the

scanner adjusts the exposure time to let in more or less light, and then tries the scan again. All this process is done before the comparing process of the print image with the stored data.

If the darkness level is adequate, the scanner system goes on to check the image definition (how sharp the fingerprint scan is). The processor looks at several straight lines moving horizontally and vertically across the image. If the fingerprint image has good definition, a line running perpendicular to the ridges will be made up of alternating sections of very dark pixels and very light pixels. If the processor finds that the image is crisp and properly exposed, it proceeds to comparing the captured fingerprint with fingerprints on file.

One disadvantage of the optical sensor scanner is the high cost of the prism, lens and camera, and mechanical assembly.

One advantage of the sensor is the speed of the sensor to capture the image. It just takes a few seconds to grab an image. Furthermore, the sensor can provide a clear image due to the reading of the derma, the sub-surface of the skin, rather than the surface only.

3.2.1.2 Capacitance Sensor

Capacitance sensor is one of the most popular fingerprints sensing technique. It uses capacitors with electrical current to generate an image of the ridges and valleys that make up a fingerprint; while optical sensor uses light as the sensing element. Below is a simple diagram of capacitive sensor. The sensor is made up of one or more

semiconductor chips containing an array of tiny cells (smaller than the width of one ridge on a finger). Each cell includes two conductor plates, covered with an insulating layer. [33]

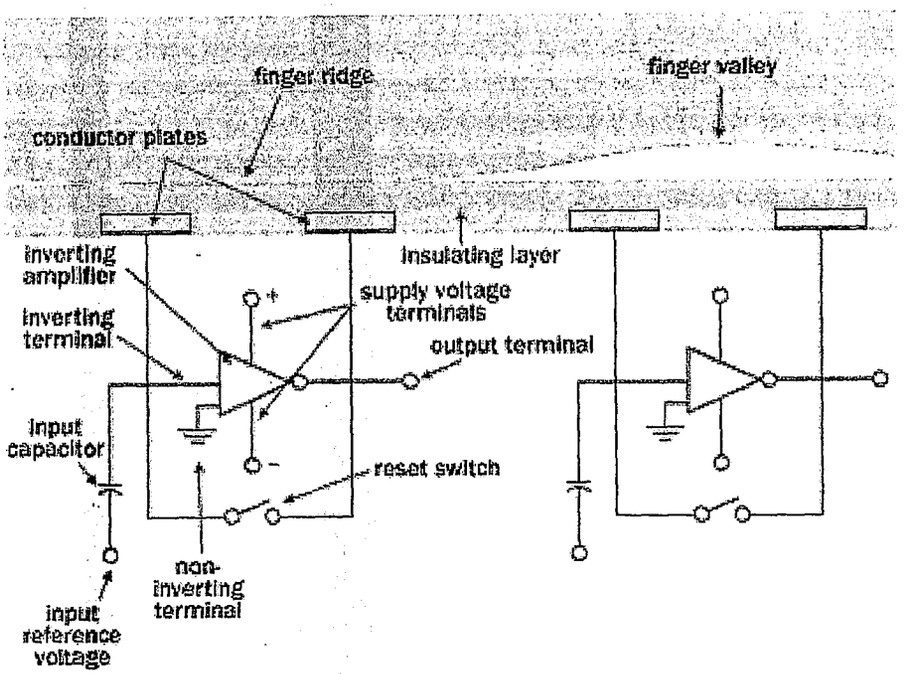


Figure 3.6: Capacitive Sensor Technology

The sensor is connected to an integrator. An integrator is an electrical circuit built around an inverting operational amplifier. The inverting amplifier is a complex semiconductor device, made up of a number of transistors, resistors and capacitors. Like any amplifier, an inverting amplifier alters one current based on fluctuations in another current. Specifically, the inverting amplifier alters a supply voltage. The alteration is based on the relative voltage of two inputs, called the inverting terminal and the non-inverting terminal. In this case, the non- inverting terminal is connected to ground, and the inverting terminal is connected to a reference voltage supply and a feedback loop. The feedback loop, which is also connected to the amplifier output,

includes the two conductor plates. The two conductor plates form a basic capacitor. Capacitor is an electrical component that can store up charge.

The surface of the finger acts as a third capacitor plate. This third capacitor plate is separated by the insulating layers in the cell structure and, in the case of the fingerprint valleys, a pocket of air. Varying the distance between the capacitor plates (by moving the finger closer or farther away from the conducting plates) changes the total capacitance (ability to store charge) of the capacitor. Because of this quality, the capacitor in a cell under a ridge will have a greater capacitance than the capacitor in a cell under a valley.

To scan the finger, the processor first closes the reset switch for each cell, which shorts each amplifier's input and output to "balance" the integrator circuit. When the switch is opened again, and the processor applies a fixed charge to the integrator circuit, the capacitors charge up. The capacitance of the feedback loop's capacitor affects the voltage at the amplifier's input, which affects the amplifier's output. Since the distance to the finger alters capacitance, a finger ridge will result in a different voltage output than a finger valley.

The sensor processor reads this voltage output and determines whether it is characteristic of a ridge or a valley. By reading every cell in the sensor array, the processor can put together an overall picture of the fingerprint, similar to the image captured by an optical sensor.

The main advantage of a capacitive sensor is that it requires a real fingerprint-type shape, rather than the pattern of light and dark that makes up the visual impression of a fingerprint. This makes the system harder to trick. Additionally, since they use a semiconductor chip rather than a CCD unit, capacitive sensors tend to be more compact that optical devices.

Some of the disadvantage of the capacitance sensor is it is expensive and have problem with the external electrical fields. The sensor is vulnerable to strong external electrical fields such as ESD (Electro-Static Discharge). Capacitive sensor also have problem with we and dry fingers. With wet fingers the users sometimes get black images, whilst dry fingers make the image pale.

3.2.1.3 Comparison Table

A comparison is made between optical and capacitive fingerprint sensor as illustrated in Table 3.1;

Table 3.1: Comparison between Optical and Capacitive Sensors

Feature	Optical	Capacitive
Image Area	Large and convenient (15mm x 20mm)	Small and expensive (10mm x 10mm)
Fake Finger Detection	Difficult to fool	Easily fooled
Design Maturity	Hundreds of thousands in daily use	New
ESD Resistance	Withstands 15KV	Withstands 1KV
Water Resistance	Resistant, sealed	Non-resistant. Can be ruined by water
Protected Sensor Window	Sealed protective window	Unsealed. Direct contact w/semiconductor surface is required.
Manufacturing Lead Time	Short, predictable	Long
Manufacturing Yield	Excellent	Complicated fab process lowers yield
Sensor Window Impact Resistance	Durable plastic window	Fragile crystal.surface Dropping pencil on window fractures surface.
Ability to Image all Fingerprints	Yes	Difficulty w/dry and wet prints – normal only
Special Coating to Aid Dry Print Image Capture	Yes	Not possible
Susceptible to Cost Reduction	Highly susceptible to cost reduction as component volumes increase	Not Susceptible to cost reduction. Silicon cannot be reduced below size of print

Based to the above comparison table, optical sensor is shown to be more reliable and effective way to use as image acquisition mechanism.

Based to the above comparison table, optical sensor is shown to be more reliable and effective way to use as image acquisition mechanism.

3.2.2 Comparison of Supervise Artificial Neural Network (ANN)

Before the discussion of the ANN implemented in this thesis, a brief discussion of the available supervise ANN and the table of comparison is carried out here. The following discussion is a brief explanation of Perceptron, Multilayer Perceptrons, Radial Basis Function Network and Boltzmann Machine.

1) Perceptron

- Feed forward architecture.
- Single layer network with a single neuron.
- Classes of pattern would have to be linearly separable.
- A hard limiter at its output end represents nonlinear element.
- Limited to performing pattern classification.
- Can perform pattern classification only on linearly separable patterns.

2) Multilayer Perceptrons (MLP) trained by Back Propagation algorithm

- Feed forward architecture.
- Performs stochastic gradient descent technique in weight space (for pattern-by-pattern updating of synaptic weights) and not an optimization technique.
- Layers - input layer which is a set of sensory units (source nodes).
    - one or more hidden layers of computational nodes.
    - Nodes.
    - output layer of computational nodes.

- hidden and output layers used as classifier are usually nonlinear.

- Each neuron includes nonlinearity at the output end. The nonlinearity is differentiable everywhere as opposed to the hard limiting used in perceptron.
- Contains one or more layers of *hidden neurons* that are not part of the input or output. These hidden neurons enable the network to learn complex tasks by extracting progressively more meaningful features from the input patterns (vectors).
- Exhibits a high degree of connectivity and it is determined by the synapses of the network. A change in the connectivity of the network requires a change in the population of synaptic connections or their weight.
- Fully connected network means that neuron in any layer is connected to all nodes/neurons in the previous layer. The signal flow is in a forward direction and on a layer-by-layer basis.
- Have different layers of network, which is a forward pass and a backward pass. A *forward pass* is where the input pattern is applied to the sensory nodes and its effect propagates through the network layer by layer and finally a set of outputs is produced as the actual response of the network. In forward pass the synaptic weights of the network are all fixed. During the *backward pass* the synaptic weights are all adjusted in accordance with the error-correction rule. Local computation use parallel architectures as an efficient method for implementation of ANN.
- Simple to compute locally means that the network relies on local computations to discover the information processing.

- Computational restriction is referred to as locally constraint meaning that the computation performed by a neuron is influenced solely by those neurons that are in physical contact with it.
- It is a *universal approximator* in a sense that a MLP can approximate any continuous multivariate function to any desired degree of accuracy, provided that sufficiently many hidden neurons are available.
- Construct *global* approximations to nonlinear input-output mapping. They are capable of generalization in regions of the input space where little or no training data are available.
- Back-propagation is a hill-climbing technique; it is easily get trapped in local minimum and in every small change in synaptic weights increases the cost function. Cost function is a cost associated with each potential solution and the best solution is the one that minimizes the cost while still fulfilling all the requirements for an acceptable solution.
- Some of the area where back-propagation algorithm has been applied successfully.
    - NETalk (Neural networks that learn to pronounce English text). (Sejnowkski and Rosenberg, 1987) [70]
    - Speech Recognition. (Cohen et. Al. [15], 1993; Renals et.al, 1992a [65])
    - Optical character recognition.(Sackinger et. Al., 1992 [68]; Le Cun et. Al., 1990[a] [50])
    - On-line handwritten character recognition.(Guyon, 1990 [26])
    - System identification. (Narendra and Parthasarathy, 1990 [55])

- Control. (Werbos, 1989 [78], 1992 [79]; Haykin and Deng, 1991 [43])
- Steering of an autonomous vehicle. (Pomerleau, 1992 [59])
- Radar target detection and classification. (Haykin and Bhattacharya, 1992 [28]; Haykin and Deng, 1991 [29])
- Passive sonar detection and classification with low alarm rate. (Casselman et. al., 1991 [11])
- Medical diagnosis of heart attack. (Baxt, 1993 [8]; Harrison et. al., 1991 [27])
- Modeling of the control of eye movements. (Robinson, 1992 [66])

3) Radial-Basis Function (RBF) Networks

- Design as a *curve-fitting (approximation) problem* in a high-dimensional space.
- Vector input pattern.
- Three different layers
  - Source nodes (sensory unit)
  - Hidden layer which provides a set of 'function' that constitute an arbitrary 'basis" for the input patterns when they are expanded into the hidden-unit space).
  - Output layer that supplies the response of the network to the activation patterns applied to the input layer.
- In short, the computation nodes in the hidden layer are quiet different and serve a different purpose from those in the output layer of the network.

- The hidden layer is nonlinear whereas the output layer is linear.
- Learning is equivalent to finding a surface that provides a 'best fit' to the training data. 'Best fit' means being measured in some statistical sense.
- The argument of the activation function of each hidden unit computes the *Euclidean norm (distance)* between the input vector and the center of that unit.
- Used exponentially decaying localized nonlinearities to construct local approximations to nonlinear input-output mapping. This will result in fast learning and reduced sensitivity to the order of presentation of training data. However, in order to represent a mapping to some desired degree of smoothness the number of radial-basis function required to span the input space adequately may have to be very large.
- Capable of implementing arbitrary nonlinear transformations of the input space, for example the XOR problem, which cannot be solved by any linear perceptron.
- Suffer from the so-called *curse of dimensionality.* Curse-of-dimensionality means the exponential increase in the number of hidden units with the dimension of the input space. Consequently it becomes particularly acute in trying to solve large-scale problem such as image recognition.
- Some of the area where RBF Network has been applied successfully;
    - Image processing. (Saha et. al., 1991 [69]; Poggio and Edelman, 1990 [58])
    - Speech recognition. (Ng and Lapedes, 1991 [56]; Niranjan and Fallside, 1990 [57])

- Time-series analysis. (He and Lapedes, 1991 [30]; Kadirkamanathan et. al., 1991 [46])
- Adaptive equalization. (Chen et. al., 1992a [12],b [13]; Cid-Sueiro and Figueiras-Vidal, 1993 [14])
- Radar point-source location.(Webb, 1993 [77])
- Medical diagnosis. (Lowe and Webb, 1990 [51])

4) Boltzmann Machine

- Recurrent architecture.
- Not a layer network, however have
  - Visible neurons which provide an interface between the network and the environment.
  - Hidden neurons which operate freely.
  - Output neurons which report the outcome to the end user.
- Closely related to Hopfield. The different is in the property of having hidden neurons. Hopfield does not have hidden neurons.
- Hidden neurons are known to learn internal representations of training patterns and thereby enhancing the performance of the network.
- Hidden and visible neuron is in the stochastic form and binary-state units.
- Uses stochastic neurons with a probabilistic firing mechanism.
- Links simulated annealing algorithm with neural network and exploit the beautiful properties of the Boltzmann Machine.
- Through training, the probability distribution of the network is matched to that of the environment. This is achieved by using the relative entropy that is rooted in information theory and is used as a measure of performance.

- The network offers a generalized approach that is applicable to the basic issues of search, representation and learning (Hinton, 1987).
- The network is guaranteed to find the global minimum of the energy function provided that the annealing schedule in learning process is performed slowly enough (Geman and Geman, 1984).
- The model can be used in different problems area and it is generally applicable.
- It has a sound mathematical background which facilitates the analysis of the model.
- Relatively easy to implement.
- Comprise the characteristic of sequential and paralel Boltzmann Machine. The sequential Boltzmann Machine means the elements in the network can change their states one at a time. Whereas parallel Boltzmann Machine means the elements in the network can change states simultaneously.
- Have property as synchronous parallelism and asynchronous parallelism. In synchronous parallelism the sets of states transition are evaluated consecutively while in asynchronous parallelism it is evaluated simultaneously.
- May be viewed as a nonlinear associative memory or content-addressable memory. The important property of content-addressable memory is the ability to retrieve the stored pattern even though an incomplete or noisy version of that pattern is presented to the network.

Table 3.2 below is the summary of the comparison discussed above:

Table 3.2: Comparison table of supervise ANN

ANN	Architecture	Layers	Performance Enhancement	Image Recognition Ability	Image Recognition Accuracy	Reliability
Perceptron	Feed-forward	Single Layer	No hidden units	Very Limited	Low	Low
MLP	Feed-forward	3 Basic Layers	Have hidden units	High	High	High
RBF	Curve-fitting	3 Different Layers	Have hidden units	Limited	Medium	High
BM	Recurrent	No Layer	Have hidden units	High	High	High

In Table 3.2, a brief study of four supervise artificial neural networks namely Perceptron, Multilayer Perceptron (with back-propagation), Radial Basis Function Network and Boltzmann Machine is presented. From the brief discussion, it can be concluded that back-propagation learning algorithm is one of the powerful technique be implemented in the area of neural network. Back-propagation has been successfully put into operation in various areas of applications. However, its application specifically in fingerprint recognition and classification has yet to be developed. For perceptron neural network, it is stated that the application of it in pattern classification is very limited since it can only be performed on linearly separable patterns. Therefore, it is not a good candidate for the verification and classification system proposed in this thesis.

The discussion for radial basis function network shows that this type of neural network has a great potential in certain type of application. However, its application in

image recognition will be less accurate since it is suffered from *curse of dimensionality* that has been discussed in the comparison table, Table 3.2.

Whereas for Boltzmann Machine neural network, it application in image recognition and classification seems to be more promising. Its ability to have hidden units enhance it performance compared to it 'predecessor' Hopfield neural network. The characteristic of Boltzmann Machine that can be sequential or parallel added an extra property to its ability to perform pattern classification and verification. With parallel Boltzmann Machine we can change the state of the elements simultaneously. In the application of the software, it means that we can tune the parameters for training, classification and verification simultaneously to get a better result. Another interesting characteristic of Boltzmann Machine is having a property of content-addressable memory. As previously discussed, this property will enable the network to retrieve the stored pattern even though an incomplete or noisy version of that pattern is presented to the network. This ability makes Boltzmann Machine a good candidate to have a robust system of image recognition and verification.

Boltzmann Machine, using the simulated annealing learning process is guaranteed to find global minimum, which is the lowest minimum and therefore the preferred solution. This is in contrast with the Back-propagation algorithm, which implements a hill-climbing technique resulting it to easily get trapped in local minimum. Detailed discussion on this topic will be discussed in section 3.2.4. Due to these entire characteristics, in this thesis we proposed the use of Boltzmann Machine as the heart of the fingerprint verification system.

3.2.3 Boltzmann Machine based ANN

In this thesis, Neural Network training is based on Boltzmann Machine which uses Attrasoft software for its implementation and it is referred to as Attrasoft Boltzmann Machine (ABM). It initially comes with a blank neural net, meaning that there is nothing stored in the network. Therefore, the first task is to train the network. A neural network is characterized by:

- Network topology,
- Connection strength between neurons,
- Node properties,
- Internal controls, and
- The updating rules.

Every time the network looks at a training pattern, the network stores the information by modifying the neuron synaptic connections. Modifying the values of the connections represents a learning process: the neural networks learn their environment by changing their internal connections. After a while, these synaptic connections hold certain values. These values represent the neural network's memory and it can be used to perform certain tasks (A brief explanation on the concept of neural network is presented in Appendix A).

A neural network system consists of the front-end subsystem and a neural network subsystem. Front-end subsystem's functions are to convert application data to neural input data (data encoding) and to convert neural output data back to application data (data decoding). Some examples include converting JPEG or GIF images to

neural data, fax data, various databases, handwritten words, scanned documents, sound data, and stock Market.

Neural network subsystem's functions can be classified into two parts, *Classification* and *Pattern Fix*, where in the former patterns are given to the network, and the network decides the classification of the patterns. While for the latter, some parts of a pattern and a classification are given, or just a part of a pattern is given, the network is asked to complete the pattern.

In summary, the neural computation process has three stages: Data encoding, Neural Computation, and Data Decoding. Data encoding and decoding are application-dependent, while the neural network is not application-dependent. ABM is a neural network simulator that is not application-dependent. It can be interfaced with any front-end systems to solve any problem.

Ackley, Hinton and Sejnowski [17] set the design of Boltzmann Machine or network in 1985. The Boltzmann Machine is closely related to the Hopfield model. The Boltzmann Machine and the Hopfield network share the following common features:

- Their processing units are binary values (± 1, say) for their states
- All the synaptic connections between their units are symmetric
- The units are picked at random and one at a time for updating
- They have no self feedback

However they are different in three important respects:

- The Boltzmann Machine permits the use of *hidden neurons*, whereas no such neurons exists in the Hopfield network
- The Boltzmann machines use *stochastic neurons* with a probabilistic firing mechanism, whereas the standard Hopfield network uses neurons based on the *McCulloch-Pitts model* with a *deterministic* firing mechanism. (For example, the rule for operating Hopfield networks in the high gain limit was simply: (1) Pick a unit. (2) Compute its net input as the sum of connection weights to other *active* units. If this net input is positive, make the unit active (state = +1), if this net input is negative or zero make the unit inactive (state= -1)).
- The Hopfield network operates in an unsupervised manner, whereas the Boltzmann Machine may also be trained by supervision of a probabilistic form.

The stochastic neurons of Boltzmann Machine can be divided into two functional groups: *visible,* and *hidden,* as depicted in Figure 3.7.

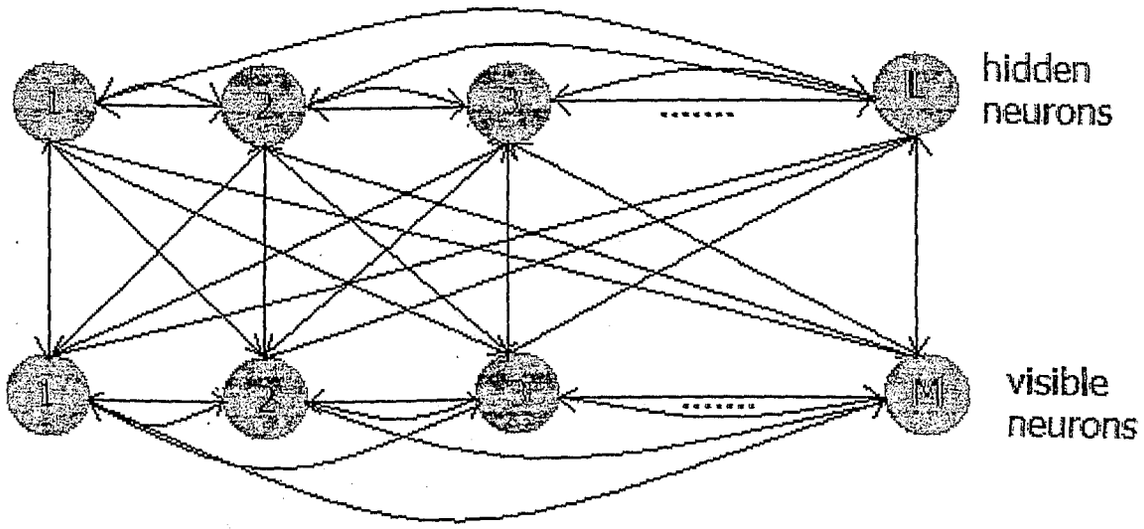


Figure 3.7: Boltzmann Machine with visible and hidden unit neurons

The visible neurons provide an interface between the network and the environment in which it operates. During the training phase of the network, the visible neurons are all *clamped* onto specific states determined by the environment. The hidden neurons, on the other hand, always operate freely; they are used to explain underlying constraints contained in the environmental input vectors.

The hidden neurons accomplish this task by capturing higher-order statistical correlations in the clamping vectors. This property represents a special case of Boltzmann Machine. It may be viewed as an "unsupervised" learning procedure for modelling a probabilistic distribution that is specified by clamping patterns onto the visible neurons with appropriate probabilities. By doing so, the network can perform *pattern completion.* Specifically, when a partial information vector is clamped onto a subset of the visible neurons, the network performs completion on the remaining visible neurons, provided that it has learned the training distribution properly.

The visible neurons may be further subdivided into *input* and *output* neurons, as shown in Figure 3.8. In this second configuration, the Boltzmann Machine performs *association* under the supervision of a teacher, with the inputs neurons receiving information from the environment and the output neurons reporting the outcome of the computation to an end user. In particular the information of the "correct" output pattern is for each input pattern may be probabilistic in nature.

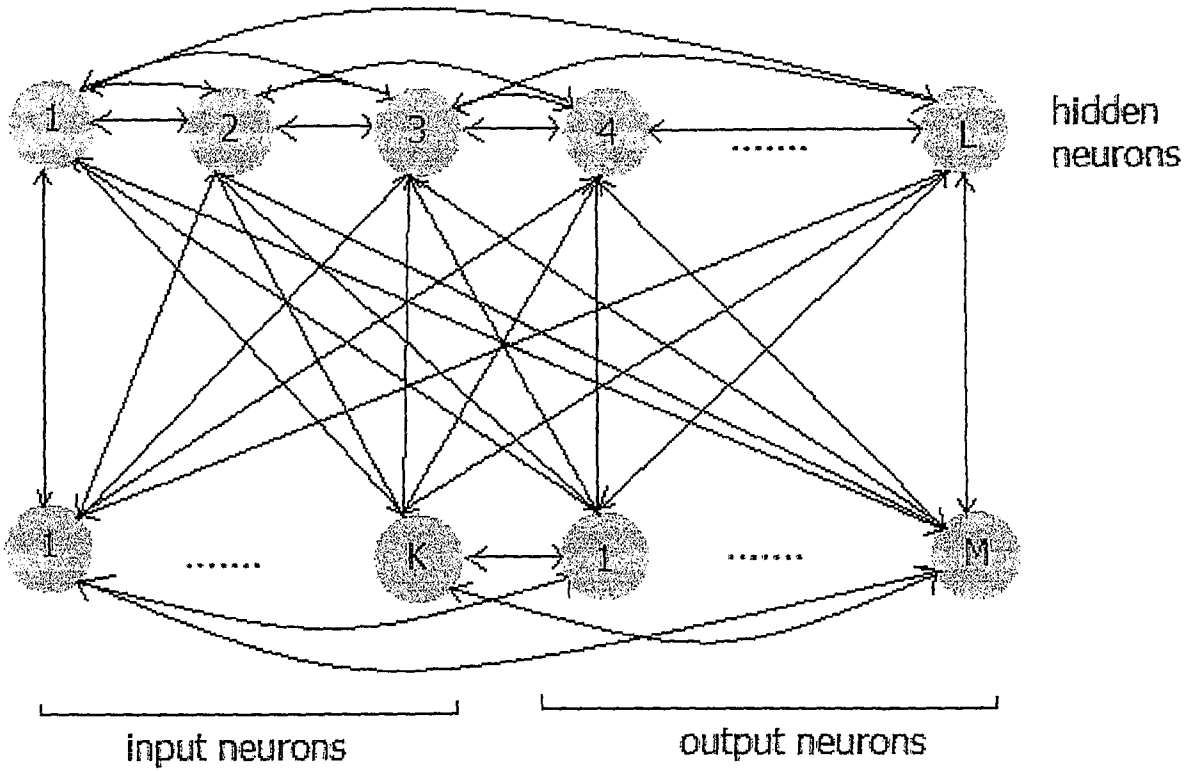


Figure 3.8: Boltzmann Machine with input, output and hidden neurons

The goal of Boltzmann Machine is to produce a neural network that correctly categorizes input patterns, according to a Boltzmann distribution. In applying this form of learning, two assumptions are made:

- Each environmental input vector (pattern) persists long enough to permit the network to reach *thermal equilibrium.*
- There is *no* structure in the chronological order in which environmental vectors are clamped into the visible units of the network.

If this particular set of synaptic weights leads to exactly the same probabilistic distribution of the states of the visible units (when the network is running freely) as that when these units are clamped by the environmental input vectors; it is said to represent a perfect model of the environmental structure.

Furthermore, the Boltzmann Machine also possesses attractive properties for solving problems from various areas such as pattern recognition, combinatorial

optimization and learning. The characteristics feature of Boltzmann Machine is a combination of massive parallelism from neural computing and simulated annealing. These will result in a promising computational tool. Some of the potential benefits of the Boltzmann Machine are:

- The model can be used in different problem areas; it is generally applicable.
- There is a sound mathematical background available which facilitates the analysis of the model.
- It is relatively easy to implement.

In short, Boltzmann Machine represents another important example of a neural network that relies on a stochastic (probabilistic) form of learning. Boltzmann Machine use the idea of simulated annealing as its basic operation. Simulated annealing process is explained in section 3.2.4.

3.2.4 Probabilistic Model

The Boltzmann Machine derives its name from the Boltzmann distribution of molecular velocities, whose mathematical form it borrows. In molecular physics, the Boltzmann distribution provides the probability density function for the kinetic energies of particles in a gas at absolute temperature, T. The probability that any given particle has energy between E and $E + dE$ is proportional to $e^{-E/kT}\, dE$, where k, the Boltzmann constant, is a universal physical constant.

David Ackley, Geoffrey Hinton, and Terrence Sejnowski in 1985 carried over this statistical function into neural network. They hypothesized that a fully connected

network of units (neurons) in binary states +1 (*on*) and −1 (*off*), with the k^{th} neuron having a probability p_k of being in the *on* state, where,

$$p_k = 1/(1+e^{\Delta E_k/T}) \tag{3.1}$$

where ΔE_k is the energy gap between the *on* and *off* states of the unit, and T is analogous to a system temperature. The energy gap ΔE_k for the k^{th} unit, is shown to be equal to the unit's activation function that is the weighted sum of all inputs from other nodes.

$$\Delta E_k = (E_{x_k=0} - E_{x_k=1}) = \sum_i^n w_{ki} x_i \tag{3.2}$$

where n is the total number of units in the network. The summation of the above equation is identical to the usual definition of the net-input value to unit k, so that

$$\Delta E_k = \text{net} \tag{3.3}$$

The motivation that led to the development of this model was based on the limitations of the Hopfield network in converging to a local rather than a global minimum in most learning cases. If a local minimum is reached, the error at the network outputs may still be unacceptably high. If the network stops learning before reaching an acceptable solution, a change in the number of hidden nodes or in the learning parameters will often fix the problem; or we can simply start over with a different set of initial weights.

When a network reaches an acceptable solution, there is no guarantee that it has reached the global minimum rather than a local one. If the solution is acceptable from an error standpoint, it does not matter whether the minimum is global or local, or even whether the training was brought to a standstill at some point before a true minimum has been reached. For some applications, this is not an issue especially if the goal is to

converge to the stored vector closest to the input vector. The *potential well* (local minimum) that is wanted is the nearest, not the deepest one (global minimum). For most optimization problems, however, the global minimum is what is sought for.

Firstly it is necessary to be able to escape from local minimum. As an analogy, consider Figure 3.9. A ball rolling on a surface, if its energy is sufficiently low, it can easily get trapped in a local minimum. That is not only higher than global minimum but also perhaps higher than other minima of interest. The way to free the ball and make it roll uphill is to "kick" it, so that it is ejected from the trap like a pinball hit by a spring mechanism. In neural terms, noise needs to be added to this situation that is something to make the ball oscillate randomly in the local trap until it makes an excursion over a nearby maximum and then downward along the next down slope.

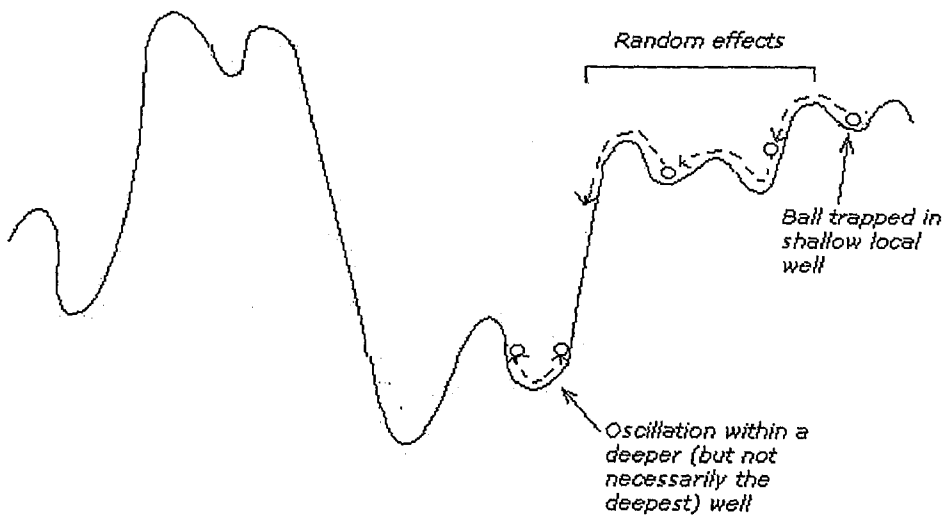


Figure 3.9: *Simulated annealing* analogy

Similarly, the stable state reached in a feedback network may represent a local minimum. Adjusting a parameter that corresponds to temperature (in the molecular parallel) moves the network state out of the local well. Later the network is "cooled" again to allow it to converge ideally into a deeper potential well. This entire cycle is

called *simulated annealing*, because of its resemblance to the temperature cycles that metals undergo as they are strengthened. Simulated annealing is particularly well suited for solving combinatorial optimization problems. The objective of combinatorial optimization problem is to minimize the cost function of a finite, discrete system characterized by a large number of possible solutions. Solution in a combinatorial optimization problem is equivalent to the states of a physical system and the cost solution is equivalent to the energy of a state.

From the equation of the probability, p_k, and the energy gap, ΔE_k, it is established that each node will oscillate between the *on* and *off* states, depending on the input it is receiving. The system as a whole will also oscillate. This oscillation occurs about a dynamic global equilibrium. The relative probabilities of occurrence of any two global systems states, A and B, having energies E_A and E_B are

$$P_A / P_B = e^{-(E_A - E_B)/T} \tag{3.4}$$

This equation fulfills the necessary requirement that when the two energies are equal, the relative probability is 1. As energy level E_A becomes much larger than E_B, the probability of P_A becomes much smaller then the probability of P_B. A nice property about this equation is that the difference of logs of the probabilities for two global states becomes (with T=1) the difference in their energies.

It is clear that a high temperature will make the system quickly escape the local minimum. Even in a deeper well, however, it will tend to be unstable because of the continuously strong random effects. Conversely, with the temperature parameter low, the system is more likely to settle toward the nearest local solution and stay there.

Having control over the temperature parameter, the network can easily be tuned by the user. This control parameter is gradually reduced as the net searches for a maximal consensus. The use of probabilistic update procedure for the activations coupled with the control parameter decreasing as the net searches for the optimal solution to the problem presented by its weights reduces the chances of the "net" getting stuck in a local minimum.

3.2.5 Different Types of Boltzmann Machine

There are two kinds of Boltzmann Machine, namely;

- The sequential Boltzmann Machine whose elements can change their states one at a time
- The parallel Boltzmann Machine whose elements can change their states simultaneously.

In this thesis, the Boltzmann Machine of the second type is used. To give an exact description one should distinguish between the so-called synchronous parallelism and asynchronous parallelism.

In the synchronous parallelism, sets of state transition are evaluated consecutively while in asynchronous parallelism set of state transitions are evaluated simultaneously. The accepted state transitions of a particular set are then communicated through the Boltzmann Machine. This implies that for the next set of state transitions, the exact configuration of the Boltzmann Machine is known. Here elements continuously generate state transitions, which are evaluated on the basis of

not necessarily up-to-date information as the state of connected elements may have changed meanwhile.

Another important characteristic of a parallel Boltzmann Machine is whether there is limited or unlimited parallelism. In limited parallelism only unconnected elements may change state in parallel. This restriction does not apply to unlimited parallelism.

3.2.6 How a simple Boltzmann Machine Works

This section explains how a simple Boltzmann Machine works. The Boltzmann Machine denoted as B represents a set of elements V (the neurons) and a set C of pairs of elements of V (the connections). All connections of the form (v, v), with $v \in V$, called loops, are assumed to be elements of C, that is $\{(v, v) | v \in V\} \subset C$.

To each element $v \in V$, it is associated with one of the two values $\{0, 1\}$. This corresponds to element being *on* (it is associated with 1) or "off" (it is associated with 0). This is also equivalent to using +1 (*on*) and -1 (*off*) as the two states. A configuration k of a Boltzmann Machine is determined by a $(0 - 1)$ vector of length $|V|$, such that the v^{th} component of the vector, $k(v)$, represents the state of element v in this configuration. Thus, for each $v \in V$ we have

$$k(v) = 1 \text{ or } k(v) = 0 \qquad (3.5)$$

Each connection $(v_1, v_2) \in C$ has a certain weight or connection strength denoted by w_{v1v2}. Connection with positive weight are called *excitatory*, those with negative weight are called *inhibitory*. A unit may also have a self-connection, w_{vv},

which is called the bias of element V (or equivalently, there may be a bias unit, which always *on* and connected to every other unit; in this interpretation, the self-connection weight would be replaced by the bias weight). A Boltzmann Machine is bi-directional, that is

$$w_{v1v2} = w_{v2v1}, \text{ for all } (v_1, v_2) \in C \tag{3.6}$$

Now, let (v_1, v_2) be connection on C. We define (v_1, v_2) to be activated in a given configuration k if both elements v_1 and v_2 are *"on"* or, if

$$k(v_1) = 1 \text{ and } k(v2) = 1. \tag{3.7}$$

Finally, let us define a function $F(k)$ that gives each configuration of the Boltzmann Machine a certain value. This value can be interpreted as a measure of the quality of that particular configuration of the Boltzmann Machine that is,

$$F(k) = \sum_{(v1,v2)\in C} w_{v1v2}.\, k(v_1)\,.\, k(v2) \tag{3.8}$$

whose value is considered the consensus. The Boltzmann Machine strives to maximize the consensus function, or in other words, it wants to find a configuration with maximal consensus. It then follows from the definition of $F(k)$ that in the Boltzmann Machine excitatory connections will tend to be activated while activation of inhibitory connections tends to be avoided.

Let us consider a small example, where

$B = (V, C)$

$V = \{v_1, v_2, v_3, v_4\}$

$C = \{(v_i, v_i) : i = 1, \ldots..,4\} \cup \{(v_1, v_2), (v_1, v_3), (v_1, v_4), (v_2, v_3), (v_3, v_4)\}$

With $w_{vivi} = 1$ for all $i = 1,..., 4$ and with all other weights indicated as in Figure 3.10.

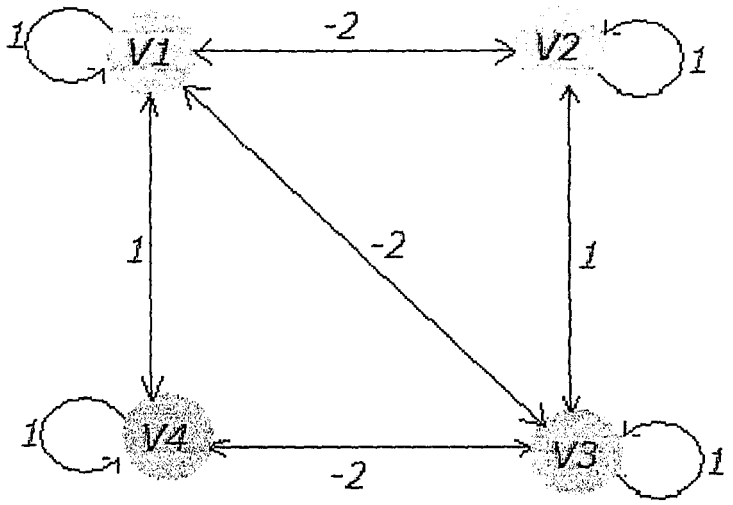


Figure 3.10: Example of a simple Boltzmann Machine

The formula for determining the number of possible configuration is as below:

$$Number\ of\ possible\ configuration = 2^n$$

where n is the number of elements (neurons). For the above example, as there are 4 elements, the number of possible configurations equals $2^4 = 16$. For instance, if all elements are *on* (implying that $k = (k(v_1),\ k(v_2),\ k(v_3),\ k(v_4)) = (1, 1, 1, 1)$), the consensus equals to the sum of the weights of all connections, which in this particular case it turns out to be 0. In this example, there are two configurations reaching a maximal consensus namely (1, 0, 0, 1) and (0, 1, 1, 0) with consensus equal to 3. The calculations for configuration (1, 0, 0, 1) are as shown below.

$$\begin{aligned} F(k) = {} & w_{v1v1}.\ k(v_1).\ k(v_1) + w_{v1v2}.\ k(v_1).\ k(v_2) + \\ & w_{v2v2}.\ k(v_2).\ k(v_2) + w_{v2v3}.\ k(v_2).\ k(v_3) + \\ & w_{v3v3}.\ k(v_3).\ k(v_3) + w_{v3v4}.\ k(v_3).\ k(v_4) + \\ & w_{v4v4}.\ k(v_4).\ k(v_4) + w_{v4v1}.\ k(v_4).\ k(v_1) + \\ & w_{v1v3}.\ k(v_1).\ k(v_3) \end{aligned} \tag{3.9}$$

So, $F(k) = (1)(1)(1) + (-2)(1)(0) +$

$$
\begin{aligned}
&(1)(0)(0) + (1)(0)(0) + \\
&(-2)(0)(1) + (1)(1)(1) + \\
&(1)(1)(1) + (-2)(1)(0) \\
&= 3
\end{aligned}
\tag{3.10}
$$

The result will be the same if the calculation is done with configuration (0, 1, 1, 0). The configuration (0, 1, 0, 1) has the following property: if the state of exactly one of the elements is changed, giving rise to a configuration l, the consensus of that configuration l will not be larger than the consensus of (0, 1, 0, 1). This property will turn out to be important in the continuation. The calculation for the consensus of this configuration can be done in the same way as mentioned above. The result for this configuration will turn out to be $F(k) = 2$.

3.3 Learning in Boltzmann Machine

Learning in Boltzmann Machine is accomplished using a simulated annealing technique. This technique has fundamental differences from other learning techniques of artificial neural network because of its stochastic nature. The output values of the processing elements in Boltzmann Machine have a probabilistic character. Thus, the output of the network as a whole also has a probabilistic character.

Procedure to perform simulated annealing technique is described by the following algorithm. Let us assume that the vector, $\boldsymbol{x} = (0, 1, 1, 0, 0, 1, 0)^t$, is one of the vectors learned by the network. The network will be able to recall this vector given only partial knowledge, for example the vector, $\boldsymbol{x}' = (0, u, 1, 0, u, 1, 0)^t$. The recall procedure will be performed using a simulated annealing technique.

1. Force the outputs of all known visible units to the values specified by the initial input vectors, x'
2. Assign all unknown visible units, and all hidden units, random output values from the set $\{1, 0\}$.
3. Select a unit, x_k, at random and calculate its net-input value, net_k
4. Regardless of the current value of unit, assign the output value, $x_k = 1$, with probability $p_k = 1/(1 + e^{-net_k/T})$. This stochastic choice can be implemented by comparing the value of p_k to that of number, z, selected randomly from a uniform distribution between zero and one. If $z \leq p_k$, then set $x_k = 1$. The parameter, T, acts as the *temperature* of the system. However, T is not the physical temperature of neural network, be that a biological or an artificial network. Rather, T is a parameter that controls the thermal fluctuation representing the effects of synaptic wise. However, T eill be referred simply as *temperature* in the context of neural network.
5. Repeat steps 3 and 4 until all units have had some probability of being selected have been updated. This number of unit-updates defines a processing cycle. For example, in a 10-unit network, 10 random unit selections would be a processing cycle. Completing a single processing cycle does not guarantee that every unit has been updated.
6. Repeat step 5 for several processing cycles, until *thermal equilibrium* has been reached at the given *temperature*, T. The number of processing cycles required to reach equilibrium is not easy to specify. Usually, we guess the number of processing cycles required to reach equilibrium.
7. Lower the temperature T, and repeat steps 3 through 7.

The *temperature* mentioned in step 4 is reduced according to the studies performed on simulated annealing by Donald Geman and Stuart Geman [24][79]. The temperature should be reduced in proportion to the inverse log of the temperature:

$$T(t_n) = \frac{T_0}{1 + \ln t_n} \tag{3.11}$$

where T_0 is the starting temperature and the discrete-time variable, t_n represents the nth processing cycle.

As stated in (3.1), the system will reach a low temperature which is needed by the network to reach equilibrium. The probability of being in any global state depends only on its energy (divided by the temperature). At high temperature the equilibrium is reached quickly, but low energy states are only slightly more than higher energy states. At low temperature, the probability of low energy states is significantly higher but it may take forever to get to equilibrium. Simulated annealing, which reduces the temperature as the network runs, is a fast way to achieve a low temperature equilibrium.

When training the Boltzmann Machine, the supplied examples are the representative of the entire population of possible input vectors. The learning algorithm that has been employed must cause the network to form a model of the entire population of input patterns based on these examples. However, there are often many different models that are consistent with the examples.

One method of choosing among different models is to insist that the model of the population produced by the network will result in the most homogeneous

distribution of input patterns consistent with the example supplied. An example [42] [80] to illustrate this phenomenon is described as follows. Suppose we know that the first component of a three-component input vector would have a value of +1 in 40 percent of all vectors in the population. Four out of eight possible three-component vectors have their first component as +1. There are an infinite number of ways that the probabilities of occurrences of those four vectors can combine to yield a total probability of 40 percent for the occurrence of +1 in the first position. One example is $P\{1,0,0\} = 10\%$, $P\{1,0,1\} = 4\%$, $P\{1,1,0\} = 8\%$, $P\{1,1,1\} = 18\%$. The most homogeneous distribution would be to assign equal probabilities to each of the four vectors, such that $P\{1,0,0\} = 10\%$, $P\{1,0,1\} = 10\%$, $P\{1,1,0\} = 10\%$, $P\{1,1,1\} = 10\%$. The rationale for this choice is that the information available gives us no reason to assign a higher probability of occurrence to any one of the vectors.

If a Boltzmann completion network learns the most homogeneous distribution, then repeated trials with an input vector of $P\{u,0,0\}$, where u is unknown, should result in a final output of $P\{1,0,0\}$ in approximately 10 trials out of every 100 (the more trials, the closer the results will be to 10 out of 100).

A simple algorithm for training a Boltzmann Machine is as follows:

1. Artificially raise the temperature of the neural network to some infinite value.
2. Anneal the system until equilibrium is reached at some low temperature value (This minimum temperature should not be zero)
3. Adjust the weights of the network so that the difference between the observed probability distribution and the canonical distribution is reduced (i.e. change the weight to reduce G)

4. Repeat steps 1 through 3 until the weight has no longer changed.

Consider a set of vectors α that we would like a Boltzmann completion network to learn. These vectors would appear as the outputs of the visible units in the network. Define γ as the set of vectors appearing on the hidden units. Clamp the output of each the visible units to each α vector. (Clamping means that the output values are fixed and do not change, even though other units may be changing according to the stochastic model). The probability that the visible units will be clamped to the vector α is P_α^+. Where the "+" indicates that the visible units have been clamped. The probability that vector α is clamped to the visible units, and that vector γ appears on the hidden units, is $P^+(\alpha\Lambda\gamma)$, and

$$P_\alpha^+ = \sum_b P^+(\alpha\Lambda\gamma) \tag{3.12}$$

Note that P_α^+ is independent of w_{ij} because the visible units are clamped to α and do not vary with changes in the w_{ij}.

The hidden layer vector need to be included since the energy of the system depends on all of the units in the network, not just on the visible units. The network's global energy, summed over all nodes, is

$$E_{ab} = -\sum_{i<j} w_{ij}^{ab} x_i^{ab} x_j^{ab} \tag{3.13}$$

where w_{ij} is the weight leading into node i from node j, x^{ab} refer to either a visible units or a hidden units.

When none of the visible units is clamped, the probability that α will appear on the visible units is given by

$$P_\alpha^- = \sum_b P^-((\alpha \Lambda \gamma) \tag{3.14}$$

where the superscript "-" indicates that the visible units are not clamped. Since this distribution represents an unclamped (*free-running*) system in equilibrium at some temperature, we can explicitly identify the probability as the canonical probabilities.

$$P^-(\alpha \Lambda \gamma) = \frac{e^{-E_{ab}/T}}{\sum_{m,n} e^{-E_{mn}/T}}$$

$$= \frac{e^{-E_{ab}/T}}{Z}$$

Then

$$P_\alpha^- = \frac{\sum_b e^{-E_{ab}/T}}{\sum_{m,n} e^{-E_{m,n}/T}} \tag{3.15}$$

We will use G to represent the distribution between P^+ and P^-.

$$G = G(P_\alpha^+ \| P_\alpha^-) = \sum_\alpha P_\alpha^+ \ln\left(\frac{P_\alpha^+}{P_\alpha^-}\right) \tag{3.16}$$

where G is similar to energy function that we have used in the previous section and we would like to minimize it so that

$$\frac{\partial G}{\partial w_{ij}} = -\sum \frac{P_\alpha^+ \partial P_\alpha^-}{P_\alpha^- \partial w_{ij}} \tag{3.17}$$

Differentiating (3.13)

$$\frac{\partial P_\alpha^-}{w_{ij}} = -\frac{1}{T}\sum_b \frac{e^{-E_{ab}/T}}{Z}\frac{\partial E_{ab}}{w_{ij}} - \sum_b \frac{e^{-E_{ab}/T}}{Z^2}\frac{\partial Z}{w_{ij}} \tag{3.18}$$

where Z is the *partition function*

$$Z = \sum_{ab} e^{-\beta E_{ab}}$$

and the factor β is related to *absolute* temperature (temperature in Kelvins) that is

$$\beta = (k_B T)^{-1} \tag{3.19}$$

in which k_B is Boltzmann constant. The derivative of the energy is

$$\frac{\partial E_{ab}}{w_{ij}} = -x_i^{ab} x_j^{ab} \tag{3.20}$$

where as the derivative of the partition function is

$$\frac{\partial Z}{w_{ij}} = \sum_{m,n} \left(-\frac{1}{T} \frac{\partial E_{mn}}{w_{ij}} e^{-E_{mn}/T}\right)$$

$$= \frac{1}{T} \sum_{m,n} x_i^{m,n} x_j^{m,n} e^{-E_{m,n}/T} \tag{3.21}$$

Substituting (3.18) and (3.19) into (3.17) yields

$$\frac{\partial P_\alpha^-}{wi_j} = \frac{1}{T} \sum_b P^-(\alpha \Lambda \gamma) x_i^{ab} x_j^{ab} - \frac{P_\alpha^-}{T} \frac{1}{T} \sum_{mn} P^-(\alpha_m \Lambda \gamma_n) x_i^{mn} x_j^{mn} \tag{3.22}$$

Substituting (3.20) into (3.15)

$$\frac{\partial G}{w_{ij}} = -\frac{1}{T} \sum_{a,b} \frac{P_\alpha^+}{P_\alpha^-} P^-(\alpha \Lambda \gamma) x_i^{ab} x_j^{ab} + \frac{\sum_a P_\alpha^+}{T} \sum_{mn} P^-(\alpha_m \Lambda \gamma_n) x_i^{mn} x_j^{mn} \tag{3.23}$$

This equation can be simplified, first by noting that $\sum_a P_\alpha^+ = 1$ and from probability theory we have

$$P^+(\alpha \Lambda \gamma) = P^+(\gamma | \alpha) P_\alpha^+ \tag{3.24}$$

This latter expression implies that the probability of having α on the visible layer and γ on the hidden layer is equal to the probability of having γ on the hidden layer given that α was on the visible layer, times the probability that α is on the visible layer. An analogous definition and statement can be made for P_α^- :

$$P^-(\alpha\Lambda\gamma) = P^-(\gamma|\alpha)P_\alpha^- \tag{3.25}$$

If α is on the visible layer, then the probability that γ will occur on the hidden layer should not depend on whether α got there by being clamped to that state or by free-running to that state. Therefore, it must be true that

$$P^+(\gamma|\alpha) = P^-(\gamma|\alpha) \tag{3.26}$$

so that

$$\frac{P^-(\alpha\Lambda\gamma)}{P^+(\alpha\Lambda\gamma)} = \frac{P_\alpha^-}{P_\alpha^+} \tag{3.27}$$

and

$$P^-(\alpha\Lambda\lambda)\frac{P_\alpha^+}{P_\alpha^-} = P^+(\alpha\Lambda\gamma)$$

which leads to

$$\frac{\partial G}{\partial w_{ij}} = -\frac{1}{T}(p_{ij}^- - p_{ij}^+) \tag{3.28}$$

where

$$p_{ij}^- = \sum_{a,b} P^-(\alpha\Lambda\gamma)x_i^{ab}x_j^{ab} \tag{3.29}$$

and

$$p_{ij}^+ = \sum_{a,b} P^+(\alpha\Lambda\gamma)x_i^{ab}x_j^{ab} \tag{3.30}$$

The weight changes occur in the direction of the negative gradient of G. Weight updates are calculated according to

$$\Delta w_{ij} = \varepsilon(p_{ij}^{+} - p_{ij}^{-}) \tag{3.31}$$

where, ε is a constant.

The quantities, p_{ij}^{+} and p_{ij}^{-} are called *co-occurrence probabilities* because they compute the frequency that x_i^{ab} and x_j^{ab} are both active (an output value of 1) averaged over all possible combinations of the patterns, α and γ. Thus, p_{ij}^{+} is the co-occurrence probability when the patterns are being clamped in the visible units, and p_{ij}^{-} is the co-occurrence probability when the network is free-running. As seen in equation 3.24, the weights will continue to change as long as the two co-occurrence probabilities differ.

This simple algorithm for Boltzmann Machine training can be expanded to include the method for determining the weight update values as follows,

1) Clamp one training vector to the visible units of the network
2) Anneal the network according to the annealing schedule until equilibrium is reached at the desired minimum temperature
3) Continue to run the network for several more processing cycles. After each processing cycle, determine which pairs of connected units are on simultaneously.
4) Average the co-occurrence results from step 3.
5) Repeat steps 1 through 4 for all training vectors, and average the co-occurrence results to get an estimated of P_{α}^{+} for each pair of connected units.

6) Unclamp the visible units, and anneal the network until equilibrium is reached at the desired minimum temperature.

7) Continue to run the network for several more processing cycles. After each processing cycle, determine which pairs of connected units are on simultaneously.

8) Average the co-occurrence results from step 7.

9) Repeat steps 6 through 8 for the same number of times as was done in step 5, and average the co-occurrence results to get an estimated P_α^- for each pair of connected units.

10) Calculate and apply the appropriate weight changes (The entire sequences from step 1 to 10 define as a sweep).

11) Repeat steps 1 through 10 until P_α^+ - P_α^- is sufficiently small.

An alternative way to decide when to stop training is to perform test procedure after each sweep. Clamp partial or noisy input vectors to the visible units, anneal the network, and see how well the network reproduces the correct vector. When the performance is adequate, training can be stopped.

In a stochastic network we could design networks, which always moved from state to state without settling down into a stable configuration by making individual neurons to behave probabilistically. By measuring the fraction of time that these networks spent in each of their states when they reached thermal equilibrium, we could use such network to generate probabilistic distributions over the various states. By changing the connection weights in the network we could encourage this equilibrium distribution to be similar to our world distribution. However, the derivatives of state

probabilities with respect to the weights included only terms involving the activation states of *pairs* of unit $x_i x_j$.

This is the limitation of a stochastic network with no hidden units and only pair wise connection. If we were to use the activations of only a certain *subset* of units as our patterns then our networks would be able to capture higher order regularities in the distributions because some of the units would be free to represent these regularities. However, hidden units introduce a new complication. It tells only the probability distribution of the states of the *visible* units while the hidden units are unknown. Hence we do not know the full probability distribution of the entire network, which is needed to calculate our weight derivatives to train our connection. But the question is, how then will the connection weights be set? Boltzmann Machine provides a learning algorithm, which adapts all the connections weight in the network given only the probability distribution over the visible units.

3.3.1 Merits and demerits

Boltzmann Machine has been found to give excellent performance on many statistical decision tasks, greatly outstripping simple back propagation network. It is also can be viewed as a nonlinear associative memory or content addressable memory. An important property of a content-addressable memory is the ability to retrieve a stored pattern even though an incomplete or noisy version of the pattern is present to the network. This property makes them powerful enough to be use in image recognition or verification system. They also provide a convenient Bayesian measure of how well a particular model or internal representation is. In this sense they incorporate the maximum likelihood principle directly into their structure. However, they are apparently slow. This is due to the many nested loops involved in the learning

procedure. But it is also largely due to the fact that Boltzmann Machine represents probabilities *directly*: their units are actively turned on and off to represent a certain activity level, not simply holding a value which encodes that level of activation.

3.4 Summary

The system designed and developed in this thesis is to increase the security of certain areas where higher security is required. In this chapter, the design of the system is discussed. In the first part, the discussion is about the taxonomy of biometrics system and the concept of biometrics for security system. The diagram of the system is presented on the system design section. The system generally consists of four main levels. In this section also a detail explanation on the type of Artificial Neural Network used in this thesis is discussed.

The next chapter will further discuss on the scanner, the door access system and the interface program.

CHAPTER 4: INTEGRATION OF SECURITY SYSTEM

4.1 Introduction

Both the hardware and software components used in the implementation of the biometrics-based security system are discussed in this chapter. Basically, the hardware components comprise of scanner (optical), and security access control system (which include the access door) whereas the software components, the ANN and interface programs, reside inside the computer. The general diagram for the integration is as illustrated in Figure 4.1.

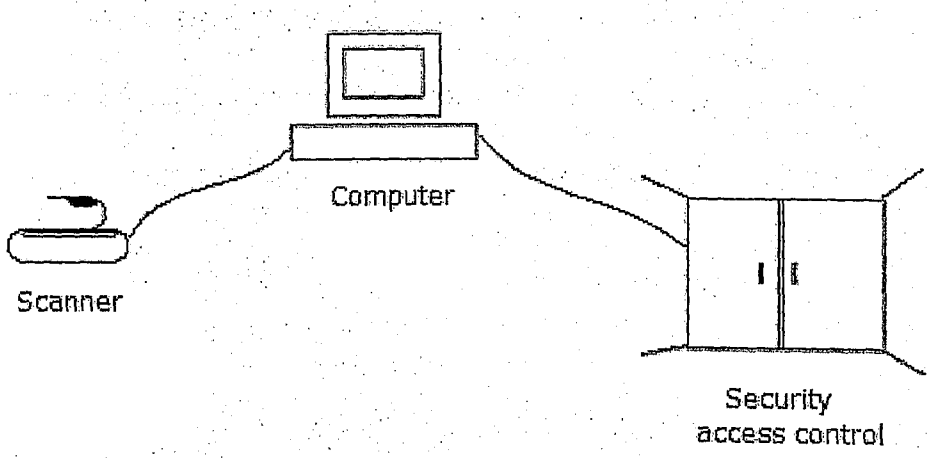


Figure 4.1: General diagram for the integration

4.2 Hardware Features

4.2.1 Fingerprint Scanner

The scanner is used to acquire the fingerprint image of each user. The finger will be scanned and the image is then stored into a directory which is called the enrollment directory. The type of scanner that is used in this thesis development is

DigitalPersona U.are.U 2000 fingerprint sensor. (The specification of the scanner is in Appendix B). The diagram of the scanner is shown below.

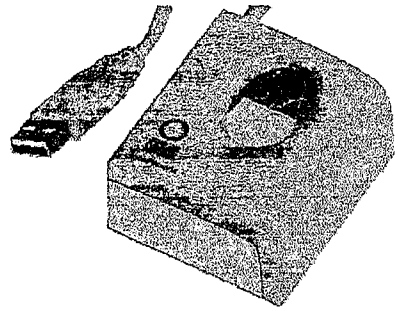


Figure 4.2: DigitalPersona U.are.U Fingerprint Sensor

The fingerprint output image from the scanner is in bitmap form. Since the ANN can only accept image in jpeg or gif format, the bitmap output image is converted using an image converter before being stored and used for training and later for verification. The trained and stored image will be acquired later to verify the life fingerprint captured by the scanner. "DigitalPersona U.are.U" fingerprint scanner uses optical technology in their fingerprint scanner. This scanner is selected for the use in this thesis for the following reason:

1) Reliability

Reliability is particularly crucial in fingerprint recognition. According to the various studies carried by the company, optical technology is the most reliable system over time based on factors that

- Susceptibility of the surface window to damage from objects being dropped on it.
- Wear and tear on the sensor window surface over time
- Susceptibility to electrostatic discharge and other electrical noise
- Contamination from dirt

2) Fragility

Optical components have been used in computer mice and desktop scanners for many years and they have proven to be extremely reliable. Therefore, fingerprint recognition hardware components could have a similar level of durability if it used optical technology. Based on the long-term experience and in-depth study of computer input devices, it is believed that optical technology is highly stable under conditions of day-to-day wear. The scanner has the added advantage of having been proven in the retail marketplace where the product has been used over a prolonged period in a widely varied market.

3) Durability of Sensor Window Surface

The sensor window is durable and not subjected to scratching. The company provides an added coating over its sensor window to improve performance with certain fingerprints.

4) Susceptibility to Electro-Static Discharge (ESD)

ESD can permanently destroy the circuitry of a silicon sensor. Major OEM customers require fingerprint components to withstand ESD ratings of 10KV of electricity. (A user walking across a carpet in winter can easily build up a charge of 10KV). The scanner components can withstand an electrical discharge of over 15KV with no performance failure. The plastic cover and case around the scanner components provide full protection from ESD.

5) Dirt Contamination/Wear and Tear

The fingerprint scanner has also the ability to withstand the rigors of the retail marketplace with customers as evidence in it use in highly "contaminated" environments that range from auto repair shops to standard office settings. While it is always the case that extreme dirt conditions will compromise performance over time, it is believes that the optical components the company has adopted for the scanner line are far more likely to perform well over time. The surfaces and case of the scanner device protect the internal components from wear, and any accumulations of dirt on the sensor window are easily removed.

6) Size

For embedded fingerprint recognition solutions there are two critical size criteria:

- Size of the sensor window
- Overall thickness of the components.

DigitalPersona's new fingerprint sensors for embedded applications are approximately 1mm thicker than typical silicon sensors. Mechanical specification of DigitalPesona U.are.U fingerprint sensor is shown in Figure 4.3.

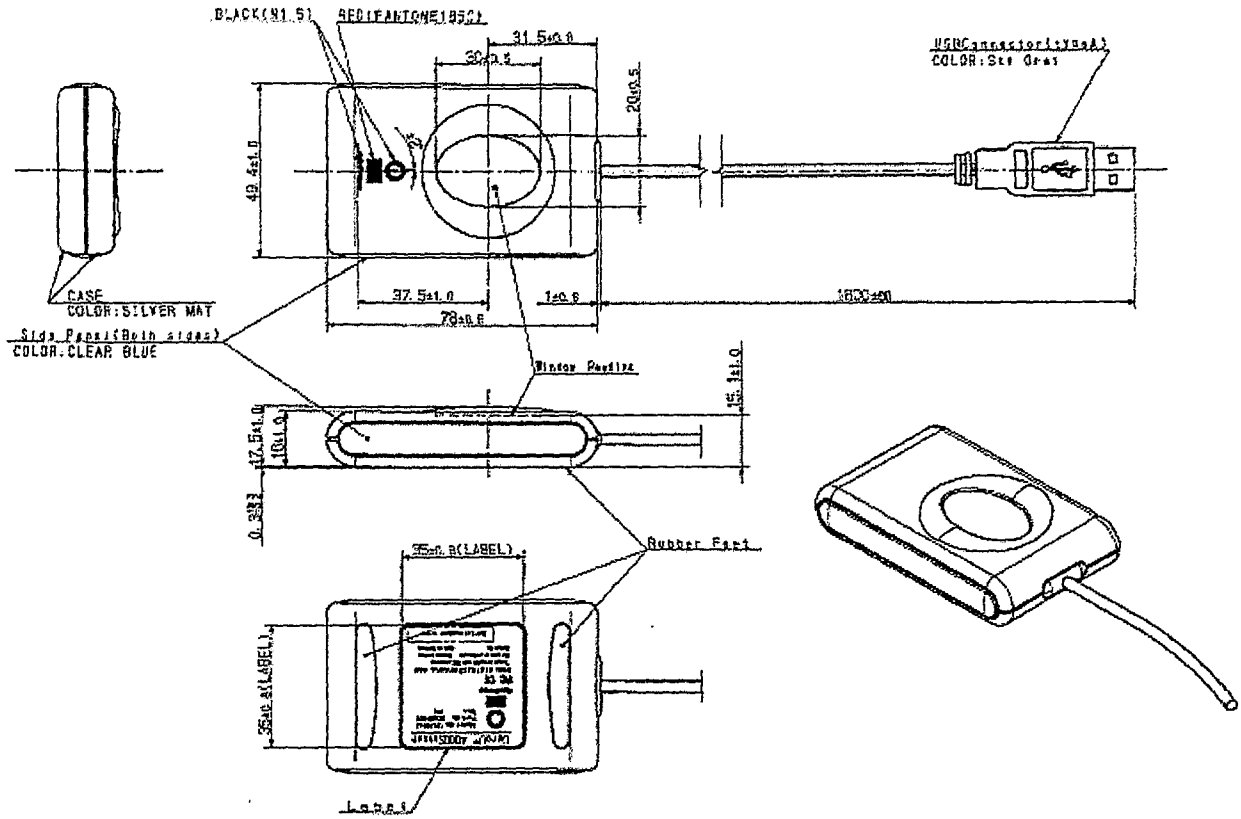


Figure 4.3: Mechanical Specification

7) Image Quality

Those new to issues surrounding fingerprint recognition often assume that the greatest difficulty in obtaining a satisfactory fingerprint image is moisture. In fact the opposite is true. The most difficult fingerprint to image is a dry print. This is because it is more difficult to make the ridges on a dry print sit flush on the sensor surface. Tiny areas of the ridge surface remain out of contact with the sensor, and the result is a poor image..

The company has added a thin, tough silicone coating to the sensor window of its devices to ensure that an optimal fingerprint image is always obtained. The conformal silicone coating creates a superior optical interference between the imaging window and the fingerprint ridges of either dry or moist skin. Such a solution cannot

be adopted for use with capacitive sensors as these sensors have less ability to recognize dry fingerprints. An example of the image scanned from the scanner fingerprint sensor and how it will be stored in the template is as shown in Figure 4.4,

Figure 4.4: Fingerprint image and minutia representation

8) Cost

The cost of the scanner is considered not just from the cost of the components today but also the price of the components as volume increase. As unit volume increase, the potential for cost reduction on the optical components is higher than the solid-state solutions. The silicon imagers in the optical sensors are significantly smaller and less costly than that of a capacitive sensor. This optical technology uses inexpensive off-the-shelf sensor arrays and LED light sources. This however does not compromise on the image quality. The explanation about the optical sensor technique and the comparison between the optical and capacitive sensor has been carried out in chapter 3.

4.2.2 Door Access Control

Door lock system can be classified into two major types: electric strike and electromagnetic lock.

Electric strike

Generally, electric door strike is the most economical mechanism for door access control. However it is more difficult to install and may require a competent locksmith. The specifications of electric strike are:

1) Fail locked (fail-secure): means the strike will be locked when the power is removed.
2) The voltage used is direct current (DC) voltage, 24V and the current is 250mA, maximum.
3) Continuous duty: these specifications correspond to the requirements of the access control.

On double doors, the door strike will be installed on the inactive door, which should be bolted down. A door cord or electric-conductive hinge will carry the power to the strike. Electric door strike also is a good choice for a *free exit* door. *Free exit* door means that a reader or keypad is installed outside the controlled area identifying users as they come in but no control is made during exit of the area. In a special situation like a full glass doors it may not be possible to install door strike and the only alternative may be to install an electro-magnetic lock.

Electromagnetic locks

The electromagnetic lock can be used in *free exit* or *controlled exit* door. *Controlled exit* means that readers or keypad are installed on both sides of the controlled door. The users are required to be authorized on upon entering as well as and exiting the area. Sometimes, the *controlled exit* is enforced only outside of normal working hours.

The installation of electromagnetic locks is less costly compare to the installation of door strike. The vast majority of electromagnetic locks require clean, stable DC voltage. The range of voltage is 12V or 24V. The clean and stable DC voltage is obtained from a filtered regulated power supply, as these are often virtually free from the noise associated with AC voltage or "ripple". A suitable filter for electromagnetic lock should have a means of adjusting the output voltage. This will allow the installer to ensure the connected devices are receiving appropriate voltage according to their specifications. A lock connected to an improperly filtered power supply can overheat, and often will vibrate enough to hum or buzz when locked. These factors can drastically reduce the holding force of the lock.

A common concern with electromagnetic locks is the condition of the lock bonding surfaces. The surface of the lock and the armature plate to which it bonds should be smooth and clean. The diagram of the lock and the armature is illustrated in Figure 4.5.

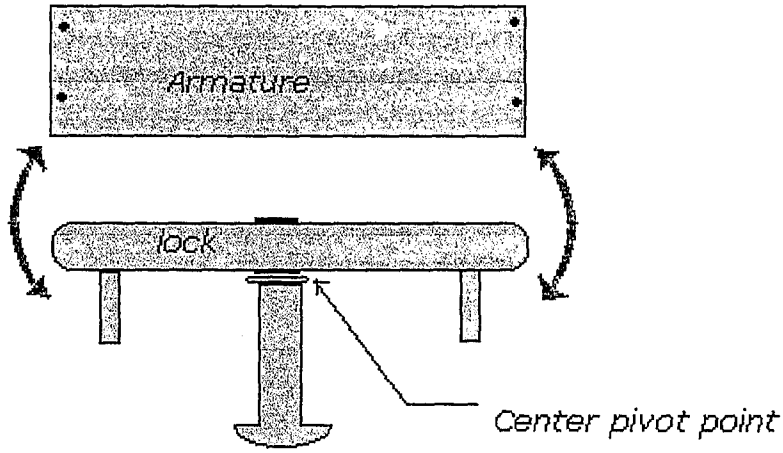


Figure 4.5: The diagram of electromagnetic lock

Many units are plated and then polished to ensure a solid bond when mated correctly. These surfaces need to be checked to make sure they can perform correctly since a piece of cellophane tape on the surface of the lock can seriously affect the holding force. The most important of all is the proper alignment of the armature plate. The armature plate mounted on the surface of the door is the only thing that an electromagnetic lock has to bond to when locked. The diagram showed how the lock and the armature plate are mounted on the door.

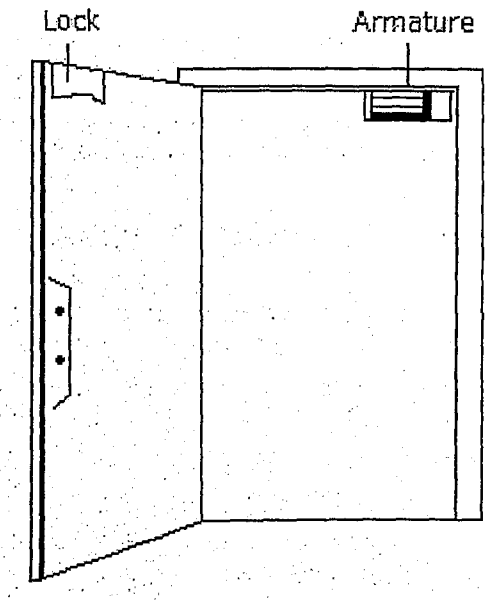


Figure 4.6: A diagram of electromagnetic lock mounted on the door

Different manufacturers have a slightly different method of mounting an armature plate, but they all have one thing in common. An electromagnetic lock's armature plate must be mounted with a certain degree of flexibility so that the lock is able to bond with the maximum amount of surface area when engaged. These plates should pivot on some sort of mounting hardware and be easy to push with just the strength of one hand.

This free range of motion is left in the installation to ensure that the maximum amount of surface area comes in contact with the surface of the magnet. If the plate is too stiff and cannot move freely, then it may not seat properly with the magnet. When the lock is activated, the armature should be pulled tightly to the magnet's surface such that even the air between them is pressed out, forming a complete bond. If the plate is only partially comes in contact with the magnet, the holding force may seem to be non-existent.

To troubleshoot lock problem, place a screwdriver or a pair of pliers to the lock. If the lock is capable of holding a screwdriver, even a little, then it is most likely working perfectly and any problems lie with the alignment of the armature. If this is the case, the problem can generally be remedied by loosening the armature in small amounts until the door locks properly. Although an armature plate needs to be capable of movement when installed, it should not be dangerously loose.

With the above discussion, the electromagnetic door lock is proposed as the implementation of the door access control system.

4.3 Interface Features

The interfaces of the system are created using Visual Basic 6.0 programming language. Visual Basic is one of the most popular and widely used programming languages available today. It provides a set of easy-to-use controls that can be used to develop impressive interactive windows-based applications easily and quickly (A brief discussion about Visual Basic is in Appendix C). The proposed window for interface is as illustrated in Figure 4.7.

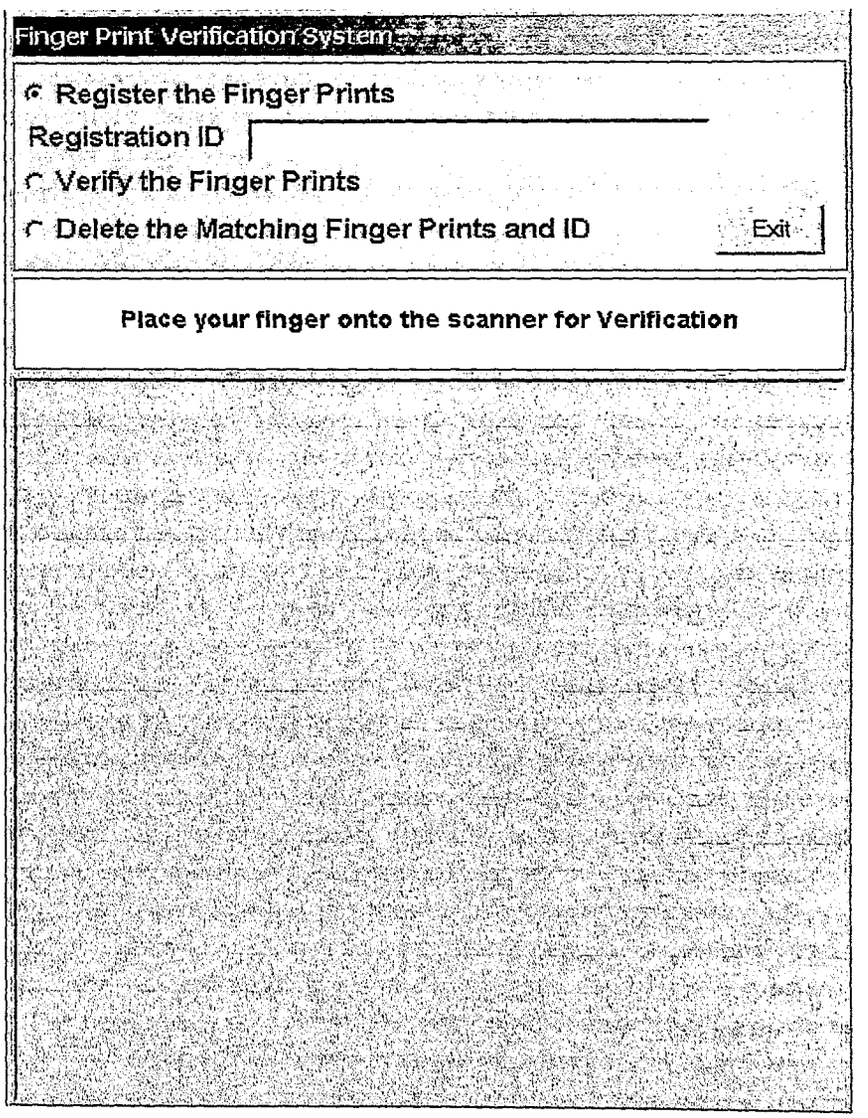


Figure 4.7: The interface window

The scanner is connected to the computer via USB communication port. Before a user places her or his finger on the scanner, an ID in enter on the "Registration ID" box shown in the diagram above. When the user scans his or her finger a newly

captured image is stored in the ID folder. Each user will scan his or her finger three times. Due to the limited ability of the system at this particular moment, the user just can scan one finger at one time. What this meant is that for each ID the image that is stored in that particular ID comes from one finger only. This first step is called enrollment procedure. The scanner communicates with the computer through the visual basic program. As mention in the previous chapter, the scanning fingerprint image will first be converted to Jpeg image by using an image converter program and then stored in the enrollment template with its ID.

During the enrollment procedure also the Image Matching Engine will be called. The Image Matching Engine, which is the ANN in this thesis, learned the content of all the scanning images. The image is classified using the ID associated with it when it is stored in the template.

In the verification procedure, the user clicks on the "Verify the Finger Prints" and place his or her finger on the scanner. The captured image is compare with the stored image. A verification signal is sent to the door lock system. If permission is granted to the person a signal is sent to the electromagnetic control door lock to unlock it. Otherwise the person is not permitted to enter the area.

This whole integration is done by the interface. The following diagram shows the system parameters, which are the software path, the scanner working directory, the data directory, the verification directory and the match files. The match file is where the ImageFinder for DOS (abm48.txt) and the verification parameters (match.txt) reside. The sample program of the interface is given in the Appendix D.

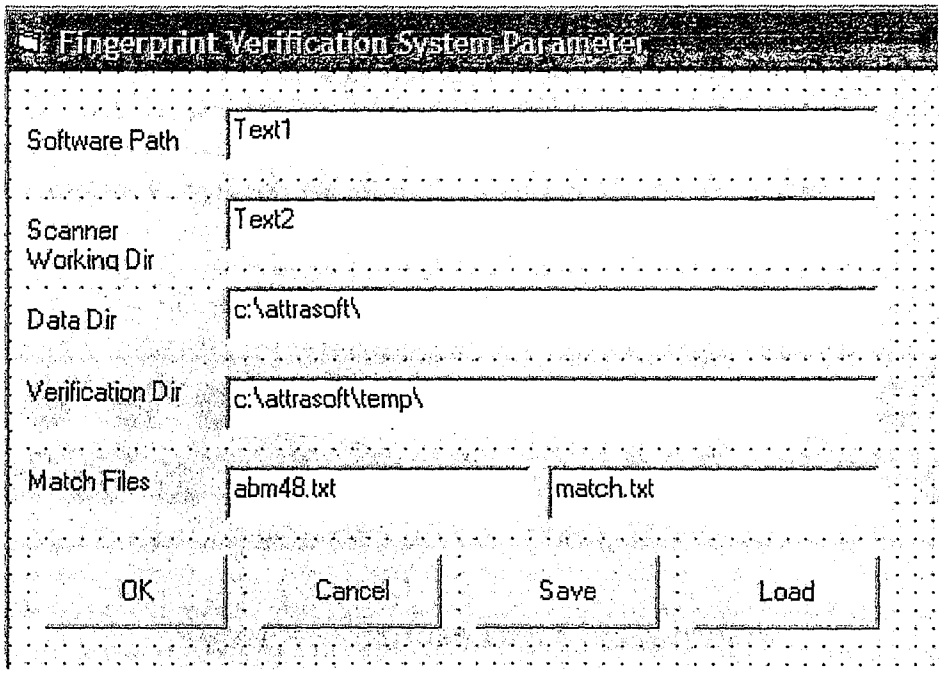


Figure 4.8: The interface system parameters.

In this thesis the implementation and the development of the system is accomplished until the resulting signal of the verification procedure. The interface to the door lock system is put on hold due to the time and budget constraints. However, for the purpose of presentation on how the computer communicate with the physical system using Visual Basic programming language as the interface program, the following techniques is shown. The communication is done through the computer parallel printer port.

4.3.1 Control of Parallel Port using Visual Basic.

Visual Basic has no built-in way to access ports. A solution is to use a dynamic link library (DLL). DLL is a file that contains a library of functions and other information that can be accessed by a Windows program. Some basics explanation of parallel port is as follows. Leven in his article write about the parallel port configuration [35].[81] What is a port? A port contains a set of signal lines that the CPU

sends or receives data with other components. The port is used to communicate via modem, printer, keyboard, mouse etc. In signaling, open signals are "1" and close signals are "0" so it is like binary system. A parallel port sends 8 bits and receives 5 bits at a time. Figure 4.9 shows the configuration of the parallel port.

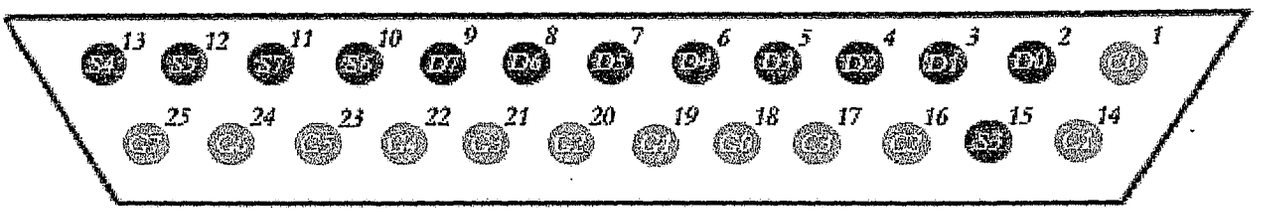


Figure 4.9: Parallel Port Configuration

There are four ports in the diagram, data ports, status ports, control ports and ground pins. The data port is the range of D0 – D7. The status port is in the range of S0 – S7. However S0, S1, S2 are invisible in the connector. This port is for reading signals but S0 is different, this bit is for timeout flag in EPP (Enhanced Parallel Port) compatible ports. The address of this status port is 0 x 379. This will always be referring to "DATA+1" and it can send 5 numeric data from the 10 - 11 - 12 - 13 - 15th pins. The following specification is for printer application.

- S0: This bit becomes higher (1) if a timeout operation occurs in EPP mode.
- S1: Not used (Maybe for decoration :))
- S2: Mostly not used but sometime this bit shows the cut condition (PIRQ) of the port
- S3: If the printer determines an error it becomes lower (0). Which is called nError or nFault
- S4: It is high (1) when the data inputs are active. Which is called Select

- S5: It is high(1) when there is no paper in printer. Which is called PaperEnd, PaperEmpty or PError
- S6: It sends low impact signaling when the printer gets a one byte data. Which is called nAck or nAcknowledge
- S7: This is the only reversed pin on the connector (see my table in the article). If the printer is busy and it cannot get any additional data this pin becomes lower, which is called Busy.

The control port is labeled C0 to C7 but C4, C5, C6, C7 are invisible in connector. The specification for these ports for the printer application is as follows.

- C0: This pin is reversed. It sends a command to read D0 to D7 on the port. When the computer starts it is high in the connector, which is called nStrobe
- C1: This pin is reversed. It sends a command to the printer to feed the next line. It is high in the connector after the machine starts, which is called Auto LF
- C2: This pin is for reset the printer and clears the buffer. Which is called nInit, nInitialize
- C3: This pin is reversed. Sends a high indicate by binary number 1 for opening data inputs. It is low after the machine starts, which is called nSelectIn
- C4: Opens the cut operation for the printer. Not visible in the connector
- C5: Sets the direction control in multidirectional ports. Not visible in the connector
- C6: Not used and also Not visible in the connector
- C7: Mostly not used but it is used as a C5 in some ports. Not visible in the connector.

The last pin configuration is the ground pins that range from G0 - G7, and these pins are mostly used for completing the circuit. The connection of the circuit is shown in Figure 4.10,

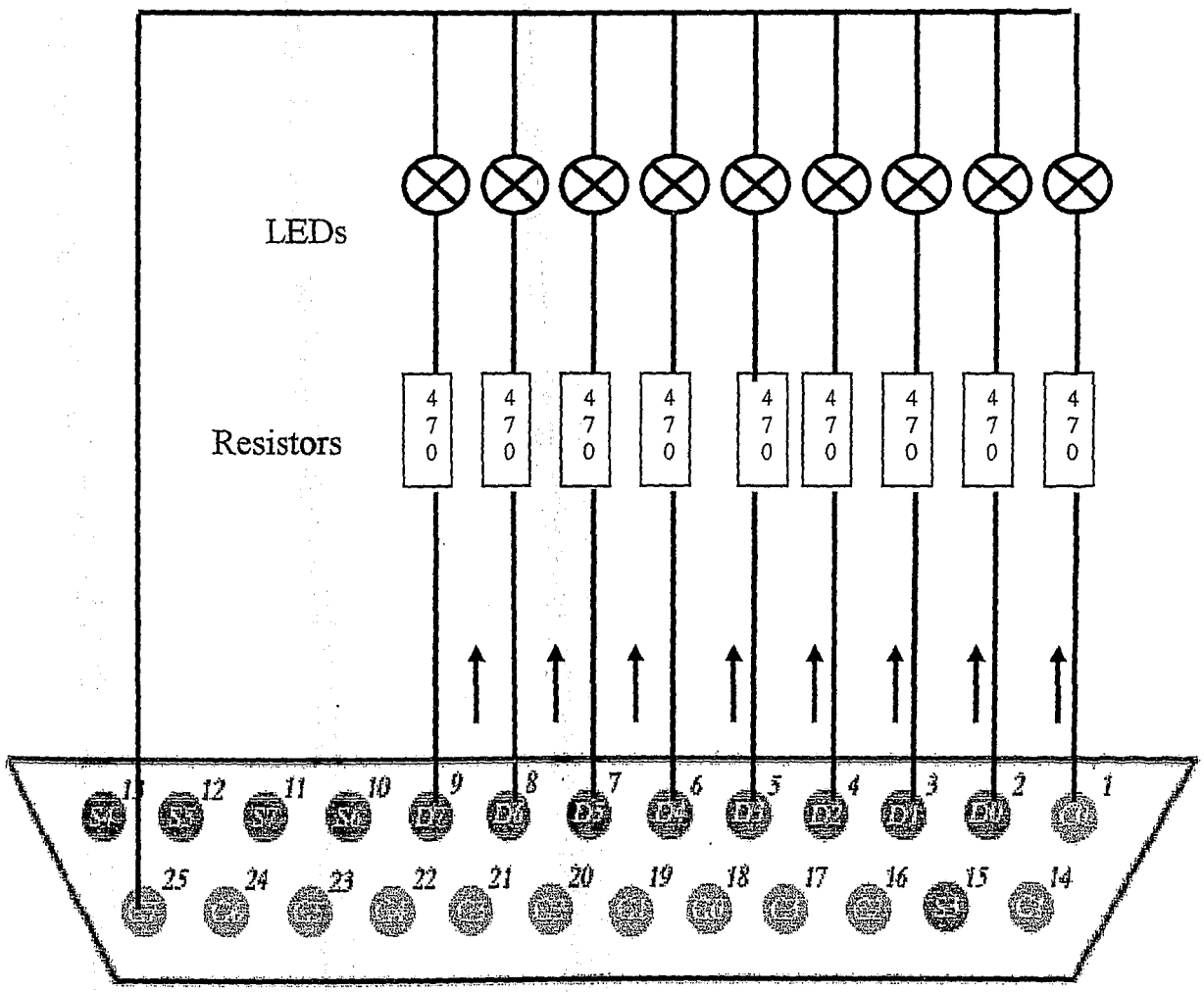


Figure 4.10: Connection of the parallel port to LED

The DLL that is use in the visual basic programming is IO.DLL. IO.DLL allows seamless port I/O operations for Windows 95/98/NT/2000/XP using the same library. IO.DLL need to be copied into C:\WINDOWS\SYSTEM directory. The IO.DLL can be downloaded from http://www.geekhideout.com [36] [82]. However, if you are a good programmer you can develop your own IO.DLL. To used it in the

application, a module need to be inserted in the existing visual basic program. The module code is shown as in Table 4.1,

Table 4.1: Visual Basic I/O Module Code

```
Public Declare Sub PortOut Lib "io.dll" (ByVal Port As Integer, ByVal Value As Byte)
Public Declare Function PortIn Lib "io.dll" (ByVal Port AS Integer) As Byte
```

4.4 Software Selection and Adaptation

The software referred in this section is the Image Matching Engine system. This image-matching engine is the ANN which resides in the ImageFinder 5.4 DOS version software. The ImageFinder uses Boltzmann Machine techniques for the implementation of the image classification and verification. As previously discussed Boltzman Macine is a very powerful algorithm for image classification and verification.

4.4.1 Attrasoft Technology

Attrasoft software is revolutionary, robust, real-time, extremely accurate, versatile, and capable of searching millions of images (easily handles an overwhelming terabyte database). Attrasoft Inc. has specialized in image recognition & pattern recognition since 1995 and has developed a technology which sole purpose is to solve various pattern recognition problems, including all types of image recognition problems, and data mining of very large databases. Attrasoft image recognition technology is having the following characteristics:

- Extremely Accurate.

- Real-Time Results.
- Can Search Millions of Images (easily handles overwhelming terabyte databases).
- Versatile. Works with any type of image including trademarks, palm prints, finger prints, passports, satellite, microscopic, forensic, X-ray, infrared, chemical signatures.

Attrasoft ImageFinder for DOS

In our development and implementation of a biometric-based security system, Attrasoft ImageFinder software is selected. Attrasoft ImageFinder use Boltzmann Machine for the verification procedure. As previously discussed, verification is one-to-one matching while identification will be one-to-many matching basis. Verification will use up a small amount of the computer's memory and also it will take up a little time to do the verification. While in identification we need to have a database compilation and also it will take a longer time for the system to get the matching result due to the time taking for searching the database.

The Attrasoft ImageFinder looks at a sample image or several images and will match all similar images from a directory. It will also classify images based on sample images. Attrasoft's technology can do the following:

- Has real-time image recognition capability (image-based not key-word base)
- Can be used for any images – stamp, satellite, face, x-ray, color photos, black and white, video recognition, fingerprint/palms prints, microscopic, forensic related images, dental images etc.

When the Attrasoft ImageFinder learns an image(s) and goes to a directory to retrieve similar images, it does not deal with keywords. This software learns the content of an image or several images directly from the image(s) and retrieves all similar images based on the content.

Attrasoft ImageFinder can match images (jpg or gif), which either look like an image (called key-image) or a segment of an image (called key-segment); or look for several key-images or key-segments.

Key-images or key-segments are used to tell the ImageFinder what to look for. This is called training. After training, the ImageFinder is ready to retrieve all similar images. This includes all translated, rotated, and scaled images. The detail discussion of Attrasoft software is attached in Appendix E.

4.5 System Integration

The integration of the whole system is as shown in Figure 4.11,

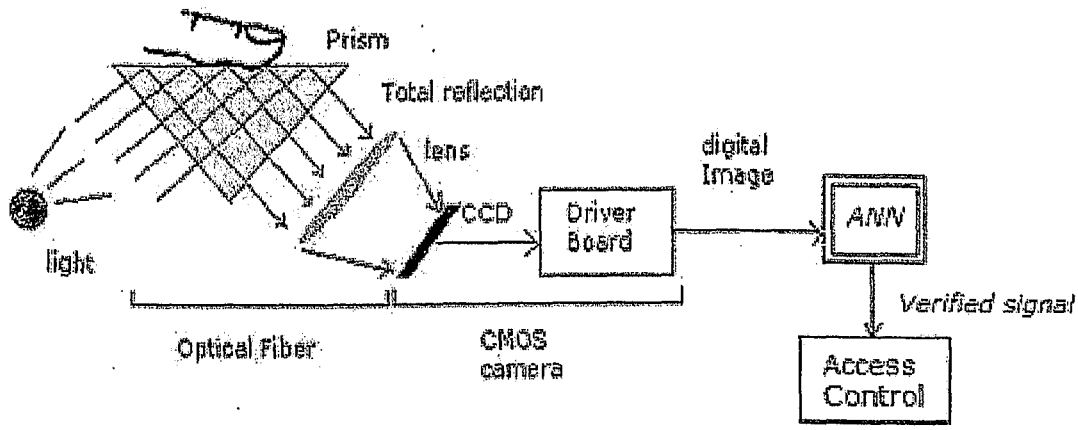


Figure 4.11: Integration of the whole system

As discuss above, interfacing the entire component with visual basic program does the integration. The testing of the system is done and the result will be discussed in chapter 5.

4.6 Summary

In this chapter, the discussion is based on the integration of the whole system. The system can be classified into two parts; the hardware and the software. The detail discussion of the two parts is as mention in the chapter. Besides the discussion of the whole integration, a detail explanation of the scanner and the door access control is also presented. In the last part of the chapter is the discussion of the software uses in this thesis and also the interface program. The interface program is to integrate between the scanner to the software, and the software to the door lock system. From the explanation and integration in this chapter, the testing and result are discussed in the next chapter.

In every development and design of any project the integration of each part is very important, since the integration will make it or break it. In this particular thesis the integration of the scanner to the ANN is a little bit complicated since the scanner output and the software input is not in the same format. As mentioned previously the output from the scanner is in bitmap form whereas the software processes an image in jpeg or gif formats. Although bitmap image will give a sharp and quality image but for the purpose of this thesis it is not needed since it will take up more computer memory. This will caused more times consumed in the process of training and verifying the image. Applying an image converter solves the problem. After successfully integrating the whole system, the system is ready for testing and compiling the result.

CHAPTER 5: TESTING AND RESULT

5.1 Introduction

In this chapter, the discussion of the testing done and result taken from the design and development of the verification system is presented. As discuss in the previous chapter the design of the system consist of two major components i.e. the hardware and the software. The hardware components are the scanner and the door lock access system. The components of the software are the Image Matching Engine and the interface program. The discussion in this chapter is based on the scanner output, Image Matching Engine and the interface program.

5.1.1 Procedure

The procedure to enroll into the system is shown in a flow form as below:

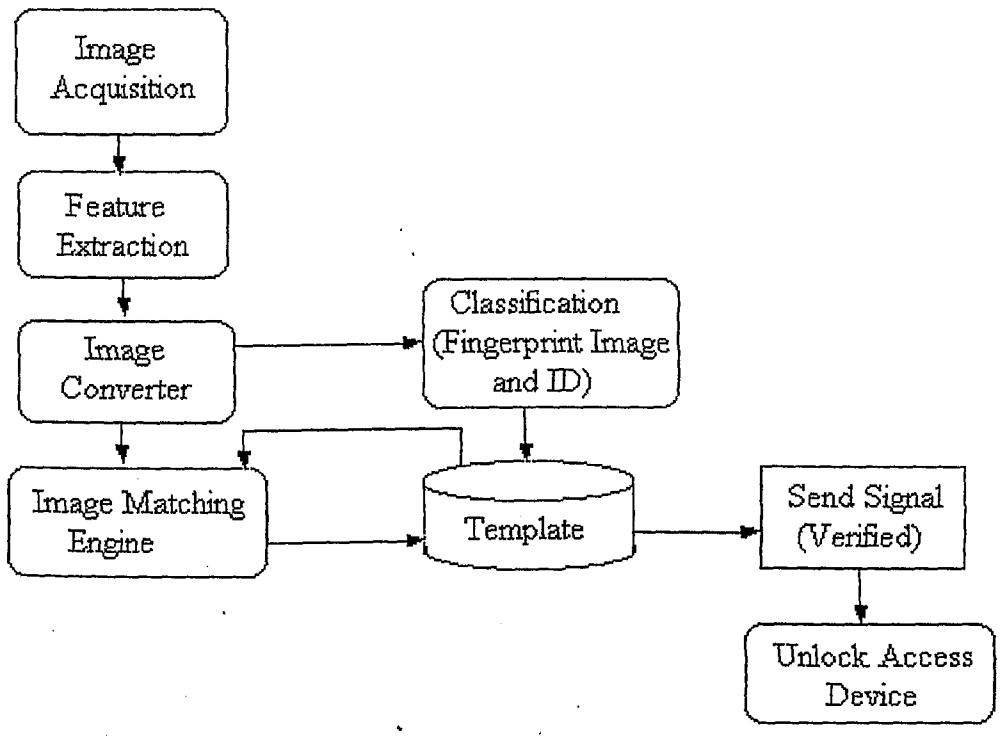


Figure 5.1: Presentation of the procedure in a flow diagram

5.1.2 Fingerprint image data

The first module in this thesis is the fingerprint image acquisition by using U.are.U fingerprint sensor. The data is taken from ten different persons. From each person five fingerprint images will be scanned. The fingerprint image is captured from thumb, index and middle fingers. This is shown in Figure 5.2 below. To complete the five fingerprints sample, two of the fingers will be scan twice with some degree of transition and rotation.

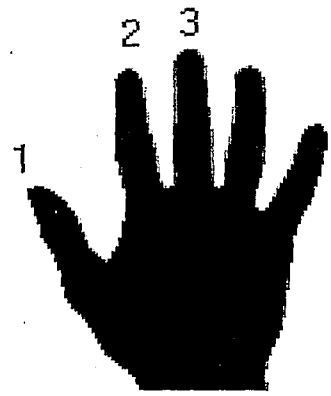


Figure 5.2: Fingers that is used as the input data

The example of the scan fingerprint images is as shown in Figure 5.3 below.

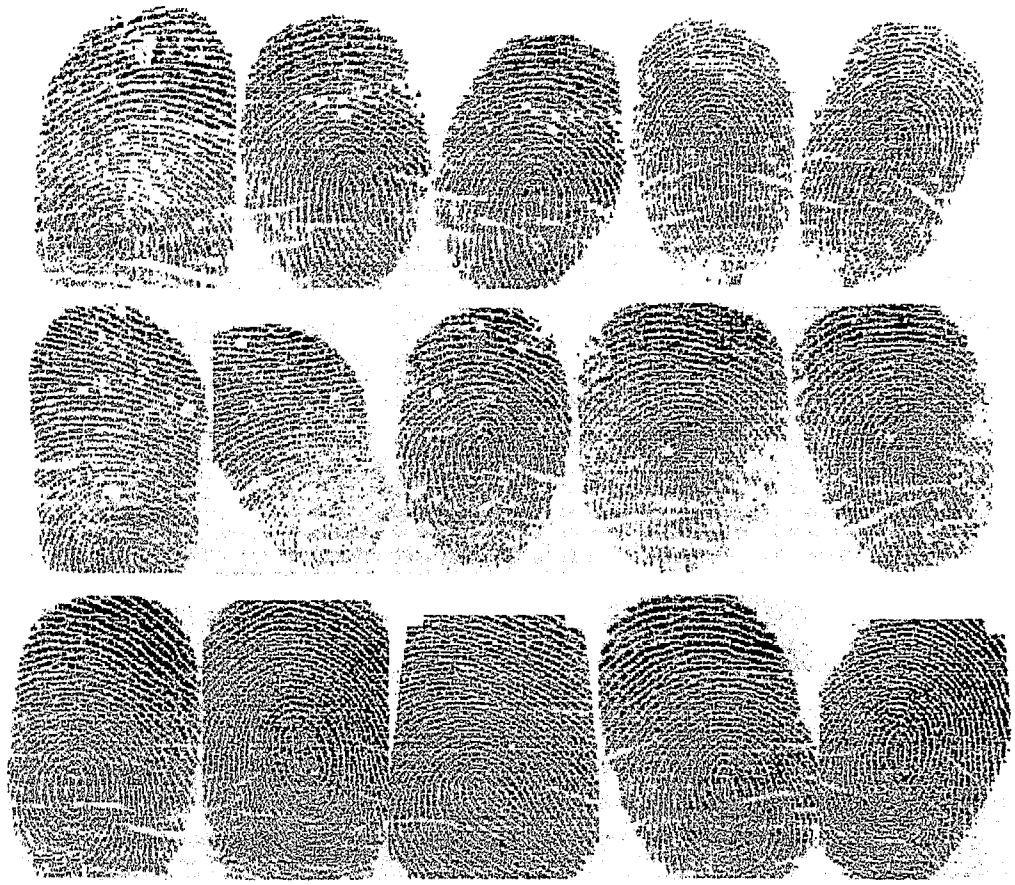


Figure 5.3: Sample of the fingerprint images

The images are saved in the enrollment template with its associated ID. Since the Image Matching Engine recognizes raw image, there is no process of feature extraction, feature thinning or feature enhancing done during this image acquisition. The image taken is in the bmp image format. However the Image Matching Engine recognize only image in jpg or gif format therefore the fingerprint image taken needed to be converted from bmp to jpg image. The following program does the image conversion.

The image is then converted into a neural data and ready to be used for the verification process. The neural data is represented in binary number, i.e. 1 and 0. One of the examples is as shown in Figure 5.4 (Other related example will be provided in Appendix F).

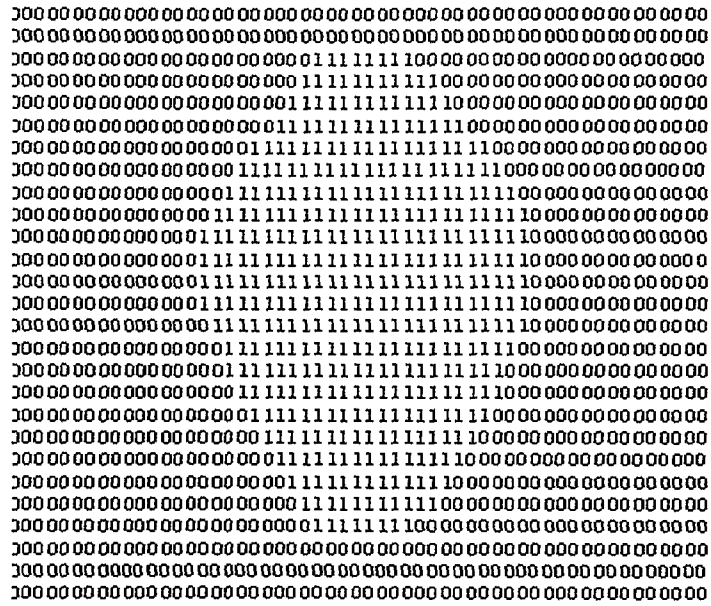


Figure 5.4: Neural data for fingerprint images

5.2 Fingerprint Verification processes

The fingerprint image is sent to the verification system that uses Boltzmann Machine technology. This data will be the input to the neural network Boltzmann Machine to be trained. The test parameter is used in the hidden layer of the Boltzmann Machine.

5.2.1 Test parameters

The parameters for this particular test are in Table 5.1;

Table 5.1: Training Test Parameters

Parameters	Value	Parameters	Value
Segment	X = 40 Y = 40 W = 155 H = 155	Internal Cut	50
Background	30	Short/Long	0
Rotation Type	0	Segment Cut	0
Reduction Type	0	Batch File Type	0
Symmetry	3	Image Dimension	0
Sensitivity	50	Translation Type	0
Blurring	25	Scaling Type	0
External Cut	100000	Shape Cut II	6
Image Type	1	Shape Type	0
Segment Size	1	Border Cut	0
Representation	0	Edge Filter	1
Shape Cut	50	Look-At-Window	x = 0 y = 0 w = 0 h = 0

5.2.2 Batch File

Batch file for the matching is as follows

Table 5.2: Matching batch file

```
1
C:\Program Files\Attrasoft\ImageFinder 5.4\Temp\
1
C:\Program Files\Attrasoft\ImageFinder 5.4\temp\00001FA.GIF
        40 40 220 220 30 0 0 3 50 25
        100000 1 1 0 50 50 0 0 0 0
        0 0 6 0 0 1 0 0 0 0
        160 255 1 192 255 2 192 255 2
end
```

And the batch file to call the ImageFinder 5.4 for DOS is as follows

Table 5.3: ImageFinder for DOS batch file

```
1
C:\attrasoft\
1
C:\attrasoft\temp\
        40 40 220 220 30 0 0 0 30 50 25
        100000 1 1 10 50 50 0 0 0 0
        0 0 6 0 0 1 0 0 0 0
        160 255 1 192 255 2 192 255 2
end
```

The proposed interface for the fingerprint verification system is as shown in Figure 5.5 below.

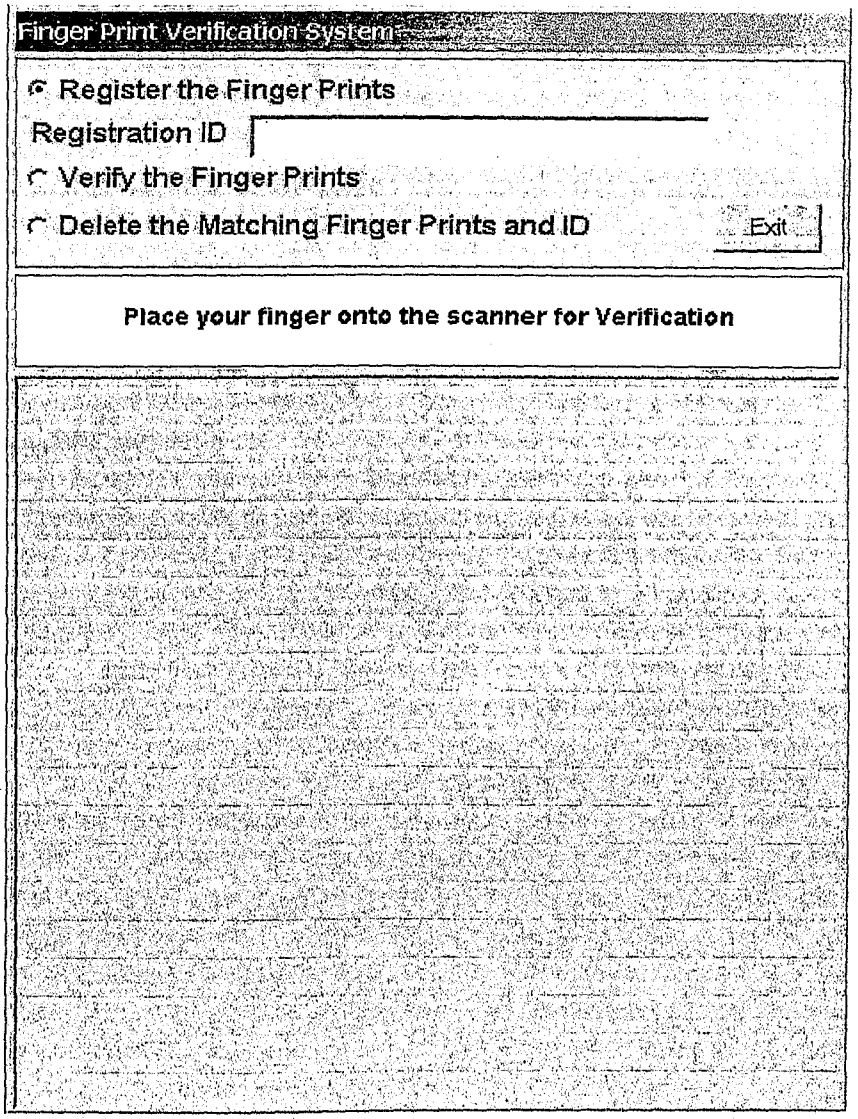


Figure 5.5: Screen shot of the proposed fingerprint verification system

Enrollment

- Then the user input an ID in the 'Registration ID' box.
- The user places his/her finger on the fingerprint scanner.
- The image will appear on the screen.
- The user needs to scan his/her finger two more times.

Verification

- Click the "Verify the Finger Print" button
- The user places his/her finger on the fingerprint scanner.
- The green or red light will appear to indicate successful matching (green light) and unsuccessful matching (red light).

Due to budget and time constraint, the development and implementation of the system is done only up to this stage.

5.3 Performance Result

After doing the test for a few images, the following results were obtained. This result is based on the data input to the ImageFinder only not for the integration of the whole system.

- There are 2 positive identification errors for user 3
- There are 3 negative identification errors for user number 2, 4, and 8

Data:

There are 50 images

10 objects x 5 images per object = 205 images

Object = 10 objects

Each object of the 50 images will be compared with its own class and 9 other classes. The numbers of tests are:

50 x 1 = 50 positive verifications

50 x 9 = 450 negative verifications

There are 500 tests total.

Single Pass Test

Total number of test = 500

Table 5.4: Table of Calculation

Positive Verification: Total positive verification = 50 From the result, there are 2 errors in positive verification = 48 Thus, **Positive Verification = 48/50 = 96%**
Negative Verification: The meaning of negative verification rate is the percentage of times the system recognizes you as using the wrong ID. Total positive verification = 450 From the result, there are 3 errors in negative verification = 447 Thus, **Negative Verification = 447/450 = 99.3%**
False Rejection Rate The meaning of False Rejection Rate is the percentage the system will reject the right person. **False Rejection = 2/50 = 4%**
False Acceptance Rate The meaning of False Acceptance Rate is the percentage of time the system will identify the wrong person as being the right person. **False Acceptance = 3/450 = 0.7%**

5.4 Result for the Control of Parallel Port

The result for the interface program to the physical control which is the LED using the parallel port is not as complete as it supposed to be. At this particular moment the result is not consistent. The LED is ON when the program is run. Supposedly the LED is ON only when the person scans his or her finger gets verified. However, in this thesis when the integration is done the result is not as expected. Therefore, further development need to be done in this particular area since the parallel port can be control perfectly before it is integrated to the existing system.

Overall conclusion, I can say that the performance for the whole system is not very good. The percentage that I can give to this work is around 30%. It need to be improved a lot more in term of the interface program.

5.5 Summary

The testing and result of the design and development of the system is discussed in this chapter. The performance result is based on the data trained to the Boltzmann Machine that reside in the Attrasoft ImageFinder software. The interface program and its interface windows are also discussed in this chapter. The discussion of the integration of the whole system is also explained.

Related to the result in this chapter, a conclusion and recommendation for the future work is discussed in chapter 6.

CHAPTER 6: CONCLUSION AND RECOMMENDATION

6.1 Conclusion

The implementation of biometrics-based security system has become one of the major developments in the area where higher security level is needed. Biometrics applications are increasingly broad-based, rapidly expanding and internationally accepted. G.Rothenbaugh in his paper "Biometrics: A Global Perspective", said that, "The influence of biometrics technology has spread to all continents on the globe". Nowadays there are a lot of applications using biometrics-based system for security purposes. Some of the areas are Law Enforcement, Prison Management, Licensing, National Identity Card, Banking and Financial Services, Access Control and Information System Management.

The great advantage of the proposed biometrics-based security system is that it uses its identification or verification on an intrinsic aspect of a human being. Identification/Verification systems that are based on something other than an intrinsic aspect of a human being are not always secure. The biometrics based security system guarantees a higher security level as compared to that of the traditional system.

In this thesis a study of biometrics-based system has been carried out. The study includes the available biometrics-based security system, their advantages and disadvantages. Based on this study, fingerprint verification system is proposed due to its maturity, acceptability, and ease of use. Furthermore, fingerprint technology has

been used in various areas for security purposes since the era of pyramid. Verification has been proposed rather than identification since verification is one-to-one (1: 1) fingerprint matching system whereas identification is one-to-many (1: N) fingerprint matching system. This property makes verification system the simplest fingerprint matching system as compared to identification. In verification, a person is identified by comparing live finger to the same fingerprint image previously stored in a template. While in identification the newly supplied fingerprint is compared to all other fingerprints in the database. In other words, verification system will speed up the time to search and verify the person's identity to be granted or not the access to certain area.

The system that has been developed and implemented in this thesis can be classified into two major components which are the hardware and the software. The hardware can be divided into two other parts which is the scanner and the door lock access system. Among the scanner technology the proposed scanner system is the optical scanner. This type of scanner technology is selected due to it durability, reliability, size, image quality, cost and other characteristic.

For the door lock access system a study has been made and the result from it shows that there are two system of door lock access that is widely used nowadays. One is the electric-strike system and the other one is the electromagnetic system. Between the two systems the electromagnetic door lock system is proposed. This system is selected due to it easy installation, safe and its wide range of usage. Even though, in this thesis the development and implementation the system did not include the door lock system but the proposed door lock system is presented.

The software part in this thesis is the image matching engine or the image verification software that uses Boltzmann Machine in its implementation. Boltzmann Machine neural network is selected because of it robustness, real time results, versatile and accuracy.

Despite of all the implementation of security system using biometrics-based however the development of biometrics based security system using neural network application is still undergoing some major development. Neural network will provide a more robust and excellent alternative for a complicated network that needs to be exact and accurate in it application. The ability of neural network to operate as supervised learning and unsupervised learning will add *intelligence* to the system especially in the security area system.

There are some problems encountered in the design, implementation and construction of this system especially in the interface stages. To have a good interface programming is crucial since the working of the system as one working unit depends on it. The automation of the system is dependent on the interface programming. The interface is carried out using Visual Basic 6 programming language. Visual basic is preferred because of it easy-to-use control. Furthermore, it is one of the most popular and widely used programming languages available today. Despite all the problems, I was able to implemented test for the feasibility and practicality of this system.

Based on the study and results, the development of prototype of a biometrics based security system using the Boltzmann Machine is found to be feasible with

potential application in high level security system. However the system can be improved and expanded for extra security purposes.

The recommendation is discussed as follows. For future work in this design and development of biometrics based security system, the developer can explore on how to improve the system. The developer can explore the software in this thesis and upgrade it to accept .bmp image format. This will surely increase the quality of the fingerprint image taken.

Another area that can be explored is to design and build a door lock system to integrate with the system that can enhance the security level, for example, to design and develop a door lock system using a fuzzy control system. This will ensure a tighter security system to the access control area especially to the area where higher security level is needed.

A thorough study of the capability and the application of the Boltzmann Machine will give a lot of benefit in terms of the development of the existing system since Botzmann Machine Neural Network is the current interest of the designers and developers.

BIBLIOGRAPHY

1. A.Jain, R.Bolle and S.Pakanti. 1999. *Biometrics: Personal Identification in Networked Society.* Kluwer Academic Publishers. London.

2. A.K.Jain, H.Lin, P.Harath and R.Bolle. 1997. An Identity-Authentication System Using Fingerprints. *Proc. IEEE Special Issues on Automated Biometrics.* 85(9). 1365-1388.

3. A.R.Roddy and J.D Stosz. 1997. Fingerprint Pattern Classification. *Pattern Recognition.* 85(9). 1390-1421.

4. Ashbourn, Julian. 2000. *Biometrics: Advanced Identity Verification, the Complete Guide.* Springer-Verlag London.

5. B. Carter. 1997. Biometrics Technologies: What they Are and How They Work. *Proceedings CTST'97.* Orlando, EL.

6. B.M. Mehtre. 1993. Fingerprint Image Analysis for Automatic Identification. *Machine Vision and Application.*

7. Baldi, P. & Chauvin, Y. 1993. Neural networks for fingerprint recognition. *Neural Computing,* 5(3). 402–418.

8. Baxt, W.1993. The Applications of Artificial Neural Network to Clinical Decision Making. *Conference on Neural Information Processing Systems - Natural and Synthetic.* Denver, CO.

9. C. Watson and C. Wilson. 1992. NIST special database 4: Fingerprint database. *Technical report, National of Standards and Technolog.*

10. C.Quek, K.B. Tan, V.K. Sagar. 2001. Pseudo-outer product Based Fuzzy Neural Network Fingerprint Verification System. *Elsvier Science Ltd.*

11. Casselman, F.L., D.F. Freeman, D.A. Kerringan, Se.E. Lane, N.H. Millstrom, and W.G. Nichols, Jr. 1991. A Neural Network-based Passive Sonar Detection and Classification Design with a Low False Alarm Rate. *IEEE Conference on Neural Networks for Ocean Engineering.* Washington DC. 49–55,

12. Chen, S., B.Mulgrew, and S.McLaughlin. 1992a. Adaptive Bayesian Feedback Equalizer Based on Radial Basis Function Network. *IEEE International Conference on Communication 3.* Chicago, Illinois. 1267–1271,

13. Chen, S., B.Mulgrew, S.McLaughlin, and P.M. Grant. 1992a. Adaptive Bayesian Equalizer with Feedback for Mobile Radio Channels. *Workshop on Adaptive Algorithms in Communication.* Bordeaux, France.

14. Cid-Sueiro, J., and A.R. Figueiras-Vidal. 1993. Improving Conventional Equalizers with Neural Network. *In Applications of Neural Networks to Telecommunications.* Hillsdale, NJ. 20–26.

15. Cohen, M., H. Franco, N. Morgan, D. Rumelhart, and V.Abrash. 1993. Context-dependent Multiple Distribution Phonetic Modeling with MLPs. *Advances in Neural Information Processing.* San Mateo, CS. 649-657.

16. D.R. Richards. 1995. Rules of Thumb For Biometricss Systems. *Security Manage.*

17. David H. Ackley, Geoffrey E. Hinton, and J. Sejnowski. 1985. *A Learning Algorithm for Boltzmann Machine.* Neurocomputing. MIT Press, Cambridge.

18. David Zhang. 2000. *Automated Biometrics: Technologies and Systems.* Kluwer Academic Publishers. London.

19. Dechman, G.H. 1996. Fingerprint identification standards for emerging application." *Fingerprint.* USA.

20. E. Newham. 1995. *The Biometrics Report.* SJB Services. New York.

21. E.M Johansson, F.U. Dowla, and D.M.Goodman. 1991. Backpropagation learning for multiplayer feed forward neural network using the conjugate gradient method. *IEEE Transactions on Neural Networks.*

22. G. Lawton. 1998. Biometrics: A New Era in Security. *IEEE Computer.* 16-18,

23. G.T. Candela, R. Chellappa. 1993. Comparative Performance of Classification Methods for Fingerprints. *U.S. Department of Commerce, National Institute of Standards and Technology.* Gaithersvurg.

24. Geman, Stuart and Geman, Donald, 1988. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *Neurocomputing.* MIT Press, Cambridge. 638 – 650,

25. Gonzalez, R.C. & Woods, R.E. 1992. *Digital image processing.* Addison-Wesley Publishing Company, Massacusset.

26. Guyon, I. 1990. Neural Networks and Applications. *Computer Physics Reports, Elsevier.* Amsterdam.

27. Harrison, R., S. Marshall, and R. Kennedy. 1991. The Early Diagnosisof Heart Attacks: A Neuro Computational Approach. *International Joint Conference on Neural Networks.* Seatle, WA. 1. 1–5.

28. Haykin, S. , And T.K. Bhattacharya. 1992. Adaptive Radar Detection using Supervised Learning Networks. *Computational Neuroscience Symposium, Indiana University-Purdue University.* Indianapolis. 35–51,

29. Haykin, S., C. Deng. 1991. Classification of Radar Clutter using Neural Networks. *IEEE Transaction on Neural Networks 2.* 589–600.

30. He, X., and A. Lapedes. 1991. Nonlinear Modeling and Prediction by Successive Approximation Using Radial Basis Function. *Technical Report LA-UR-91-1375, Los Alamos National Laboratory.* Los Alamos, NM.

31. Hrecha, A.K. & McHugh, J.A. 1990. Automated fingerprint recognition using structural matching. *Pattern Recognition.* 23(8), 893-904.

32. http://computer.howstuffworks.com/fingerprint-scanner2.htm .

33. http://computer.howstuffworks.com/fingerprint-scanner3.htm

34. http://www.biometrika.it/eng/wp_fingintro.html.

35. http://www.codeproject.com/csharp

36. http://www.geekhideout.com

37. http://www.icsa.net/services/consortia/cbd-c/sec4.html.

38. J.D Woodward. 1997. Biometrics: Privacy's Foe or Privacy's Friend?. *Proc. IEEE Special Issues on Automated Biometrics.* 85(9). 1480-1492,

39. J.G. Daugman and G.O.Williams. 1996. A Proposed Standard for Biometrics Decidability. *Proceedings CardTech, SecureTech Conference.* Atlanta, GA. 223-234.

40. J.G. Daugman. 1993. High confidence Visual Recognition of Persons by a Test of Statistical Independence. *IEEE Transaction on PAMI.* 15.

41. J.L.Blue and P.J.Gother. 1992. Training Feed Forward Network Using Conjugate Gradients. *Conference Character Recognition and Digitizer Technologies, SPIE.* San Jose California, 1661. 179 – 190.

42. James A. Freeman, David M. Skapura. 1992. *Neural Networks: Algorithms, Applications, and Programming Techniques.* Addison-Wesley Publishing Company, New York.

43. Jordon, M.I., and R.A. Jacobs. 1990. Learning to Control and Unstable System with Forward Modeling. *Advances in Neural Information Processing Systems 2.* San Mateo, CA. 324 – 331.

44. K.C. Yow and R.Clipolla. 1997. Feature-based Human Face Detection. *Image and Vision Computing.* 15. 713-735.

45. K.Huang and H.Yan. 1997. Off-line Signature Verification Based in Geometric Feature Extraction and Neural Network Classification. *Pattern Recognition.* 30(1). 9-17.

46. Kadirkamanathan, V., M. Niranjan, and R. Fallside. 1991. Sequential Adaptation of Radial Basis Function Neural Networks. *In Advances in Neural Information Processing Systems .* San Mateo, CA. 721 – 727.

47. Karu , K. Jain, A.K. 1996. Fingerprint classification. *Pattern Recognition.* 29(3), 389–404.

48. Kot, Alex. 2003. An Overview of Recent Biometric Technologies. *IMAGE 2003*. Kuching, Malaysia.

49. L.R. Rabiner and B.H. Juang. 1986. An Introduction to hidden Markov Models. *IEEE ASSP magazine*. 3(1). 4–16.

50. LeCun, Y., B. Boser, J.S. Denker, D. Henderson, R.E.Howard, W. Hubbard, and L.D. Jackel. 1990a. Handwritten Digit Recognition with a Back Propogation Network." *Advanced in Neural Information Processing 2*. San Mateo. 396-404.

51. Lowe, D., A.R.Webb. 1990. Exploiting Prior Knowledge in Network Optimization: An Illustration from Medical Prognosis. *Network 1*. 299–323.

52. M.Golfarelli, D.Maio and D.Maltoni. 1997. On the Error-Reject Trade-Off in Biometrics Verification Systems. *IEEE Trans. Pattern Analysis and Machine intelligence*.19(7). 786–796.

53. Mantaras, R. L. 1990. *Approximate reasoning models*. Ellis Horwood Limited. Chichester, UK.

54. Moayer, B & Fu.K.S. 1990. A syntactic approach to fingerprint recognition. *Pattern Recognition*.7(1), 1-23.

55. Narendra, K.S., and K.Parthasarathy. 1990. Identification and Control of Dynamical Systems using Neural Networks. *IEEE Transactions on Neural Network 1*. 4-27,

56. Ng, K., and R.P Lippmann. 1991. Practical Characteristics of Neural Network and Conventional Pattern Classifiers. *In Advances in Neural Information Processing Systems 3* . San Mateo, CA. 970–976.

57. Niranjan, M., and E.F.Fallside. 1990. Neural Networks and Radial Basis Functions in Classifying Static Speech Patterns. *Computer Speech and Language*. 4. 275–289.

58. Poggio, T., and S.Edelman. 1990. *A Network that Learns to Recognize Three-Dimensional Objects*. Nature. London.

59. Pomerleau, D.A. 1992. Neural Network Perception for Mobile Robot Guidance. *PhD Dissertation, School of Computer Science*. Carnegie Mellon University, Pittsburgh, PA.

60. R. Chandrasekaran. 1997. Brave New Whorl: ID System Using The Human Body Are Here, but Privacy Issues Persist. *Washington Post*.

61. R. Clarke. 1994. Human Identification in Information System: Management Challenges and Public Policy Issues. *Info. Technology People*. 7(4). 6-37.

62. R.B. Hill. 1998. Apparatus and method for identifying individuals through their retinal vasculature patterns. *US Pattern*. 4.

63. R.Mandelbaum. 1994. Vital Signs of Identity. *IEEE Spectrum*. 22-30.

64. R.P. Wildes. 1997. Iris Recognition: An Emerging Biometrics Technology. *Proc. IEEE Special Issue on Automated Biometrics.* 85(9). 1348-1363.

65. Renals, S., N. Morgan, H. Bourland, H.Franco and M.Cohen. 1992a. Connectionist Optimization of Tied Mixture Hidden Markov Models. *Advanced in Information Processing System.* San Mateo, CA. 167-174.

66. Robinson, D.A. 1992. Signal Processing by Neural Networks in The Control of Eye Movements. *Computational Neuroscience Symposium, Indiana University-Purdue University.* Indianapolis. 73–78.

67. Rosenfeld, A. & Kak, A.C. 1979. Digital picture processing. *Academic Press.* New York.

68. Sackinger, E., B.E. Baser, J. Bromley, Y. LeCun, and L.D. Jackel. 1992. Application of the ANNA Neural Network Chip to High-Speed Character Recognition. *IEEE Transactions on Neural Networks 3.* 498–505,

69. Saha, A., J. Christian, D.S. Tang, and C.L. Wu. 1991.Oriented Non-Radial Basis Functions for Image Coding and Analysis. *In Advances in Neural Information Processing Systems 3.* San Mateo, CA. 728–734,

70. Sejnowksi, T.J., and C.r. Rosenberg. 1987. Parallel Networks that Learn to Pronounce English Text. *Complex System 1.* 145-168,

71. Senior ,Andrew. 1998. A Hidden Markov Model Fingeprint Classifier.*Watson Research Center.* New York, USA.

72. Senior, Andrew. 1998. A Hidden Markov Model Fingerprint Classifier. *IEEE ASSP Magazine.* 3(1). 306-310.

73. Subramaniam, R. & Mehrotra, R. 1988. Automatic threshold selection based on information gain. *Piece Recognition and Image Processing, SPIE.* 956. 2-5.

74. T.M Cover and P.E Hart. 1995. Nearest neighbor pattern classification. *IEEE Transaction on Information Processing.* 13. 21-27.

75. W. She, M. Surette and R. Khanna. 1997. Evaluation of Automated Biometrics-Based Identification And Verification Systems. *Proc. IEEE Special Issues on Automated Biometricss.* 85(9). 1464 – 1478.

76. W.Shu and D.Zhang. 1998. Palm print Verification: An Implementation of Biometrics Technology. *Proceeding of ICPR'98.* Brisbane, Australia. 1. 219–221.

77. Webb, A.R. 1993. Functional Approximation by Feed-Forward Networks: A Least-Squares Approach to Generalisation. *IEEE Transactions on Neural Networks 5.*

78. Werbos, P.J. 1989. Backpropagation and Neurocontrol: A review and Prospectus. *International Joint Conference on Neural Networks.* Washington DC. 1. 209-216,

79. Werbos, P.J. 1992. Neural Networks and the Human Mind: New Mathematics Fits Humanistic Insight. *IEEE International Conference on Systems, Man, and Cybernetic.* Chicago, IL. 1. 78-83.

80. Zhou, R.W. and Quek, C. 1996. AF: An Automatic Fuzzy Neural Network Driven Signature Verification System. *In Proceedings of the International Conference on Neural Networks. ICNN '96.* Washington DC. 2. 1156-1161.

81. Zhou, R.W., Quek, C., & Ng, G.S. 1995. A novel single-pass thinning algorithm and an effective set of performance criteria. *Pattern Recognition Letters.* 16(12). 1267-1275.

APPENDIX A

What is a Neural Network?

An Artificial Neural Network (ANN) is an information-processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. ANNs, like people, learn by example. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons. This is true of ANNs as well.

Historical background

Neural network simulations appear to be a recent development. However, this field was established before the advent of computers, and has survived at least one major setback and several eras.

Many important advances have been boosted by the use of inexpensive computer emulations. Following an initial period of enthusiasm, the field survived a period of frustration and disrepute. During this period when funding and professional support was minimal, relatively few researchers made important advances. These pioneers were able to develop convincing technology, which surpassed the limitations identified by Minsky and Papert. Minsky and Papert, published a book (in 1969) in which they summed up a general feeling of frustration (against neural networks)

among researchers, and was thus accepted by most without further analysis. Currently, the neural network field enjoys a resurgence of interest and a corresponding increase in funding.

The first artificial neuron was produced in 1943 by the neuro-physiologist Warren McCulloch and the logician Walter Pits. But the technology available at that time did not allow them to do too much.

Why use neural networks?

Either humans or other computer techniques can use neural networks, with their remarkable ability to derive meaning from complicated or imprecise data, to extract patterns and detect trends that are too complex to be noticed. A trained neural network can be thought of as an "expert" in the category of information it has been given to analyze. This expert can then be used to provide projections given new situations of interest and answer "what if" questions.

Other advantages include:

- ***Adaptive learning***: An ability to learn how to do tasks based on the data given for training or initial experience.
- **Self-Organization**: An ANN can create its own organization or representation of the information it receives during learning time.
- **Real Time Operation**: ANN computations may be carried out in parallel, and special hardware devices are being designed and manufactured which take advantage of this capability.

- **Fault Tolerance via Redundant Information Coding**: Partial destruction of a network leads to the corresponding degradation of performance. However, some network capabilities may be retained even with major network damage.

Neural networks versus conventional computers

Neural networks take a different approach to problem solving than that of conventional computers. Conventional computers use an algorithmic approach i.e. the computer follows a set of instructions in order to solve a problem. Unless the specific steps that the computer needs to follow are known the computer cannot solve the problem. That restricts the problem solving capability of conventional computers to problems that we already understand and know how to solve. But computers would be so much more useful if they could do things that we don't exactly know how to do.

Neural networks process information in a similar way the human brain does. The network is composed of a large number of highly interconnected processing elements (neurons) working in parallel to solve a specific problem. Neural networks learn by example. They cannot be programmed to perform a specific task. The examples must be selected carefully otherwise useful time is wasted or even worse the network might be functioning incorrectly. The disadvantage is that because the network finds out how to solve the problem by itself, its operation can be unpredictable.

On the other hand, conventional computers use a cognitive approach to problem solving; the way the problem is to solve must be known and stated in small unambiguous instructions. These instructions are then converted to a high-level language program and then into machine code that the computer can understand. These

machines are totally predictable; if anything goes wrong is due to a software or hardware fault.

Neural networks and conventional algorithmic computers are not in competition but complement each other. There are tasks are more suited to an algorithmic approach like arithmetic operations and tasks that are more suited to neural networks. Even more, a large number of tasks require systems that use a combination of the two approaches (normally a conventional computer is used to supervise the neural network) in order to perform at maximum efficiency.

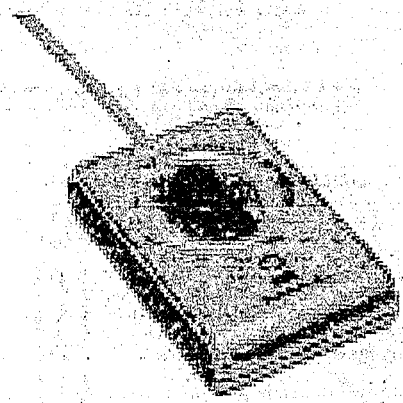
APPENDIX B

Data specification of U.are.U 4000 Fingerprint Sensor, U.are.U 2000 Fingerprint Sensor is the same family as U.are.U 4000 Fingerprint Sensor

U.are.U® 4000

USB Fingerprint Sensor

digitalPersona



Applications

- Desktop PC security
- Mobile PCs
- Custom applications
- Home and Office Use

Features

- Small form factor
- Excellent image quality
- Encrypted image data
- Latent print rejection
- Counterfeit image rejection
- Scratch resistant
- Rugged
- Works well with dry, moist, or rough fingerprints
- Challenge response link
- Compatible with all U.are.U applications
- Drivers support Windows 98, Me, NT 4.0, 2000, XP

Product Description

The U.are.U 4000 is a USB fingerprint sensor designed for use with DigitalPersona's U.are.U applications and developer tools.

The user simply places his/her finger on the sensing surface, and the device quickly and automatically captures the fingerprint image. On-board electronics calibrate the device and encrypt the image data before sending it over the USB interface.

DigitalPersona™ products utilize optical fingerprint sensing technology for superior image quality and product reliability. The U.are.U 4000 Sensor and DigitalPersona recognition software have an unmatched ability to recognize even the most difficult fingerprints.

Key Specifications

- Pixel resolution
  - 512 dpi (average x,y over the field)
- Image capture area:
  - 14.6 mm (nominal width at center)
  - 18.1 mm (nominal length)
- 8-bit grayscale (256 levels of gray)
- Sensor size: approx. 29 mm x 45 mm x 18 mm
- Compatible with USB specifications 1.0, 1.1, 2.0

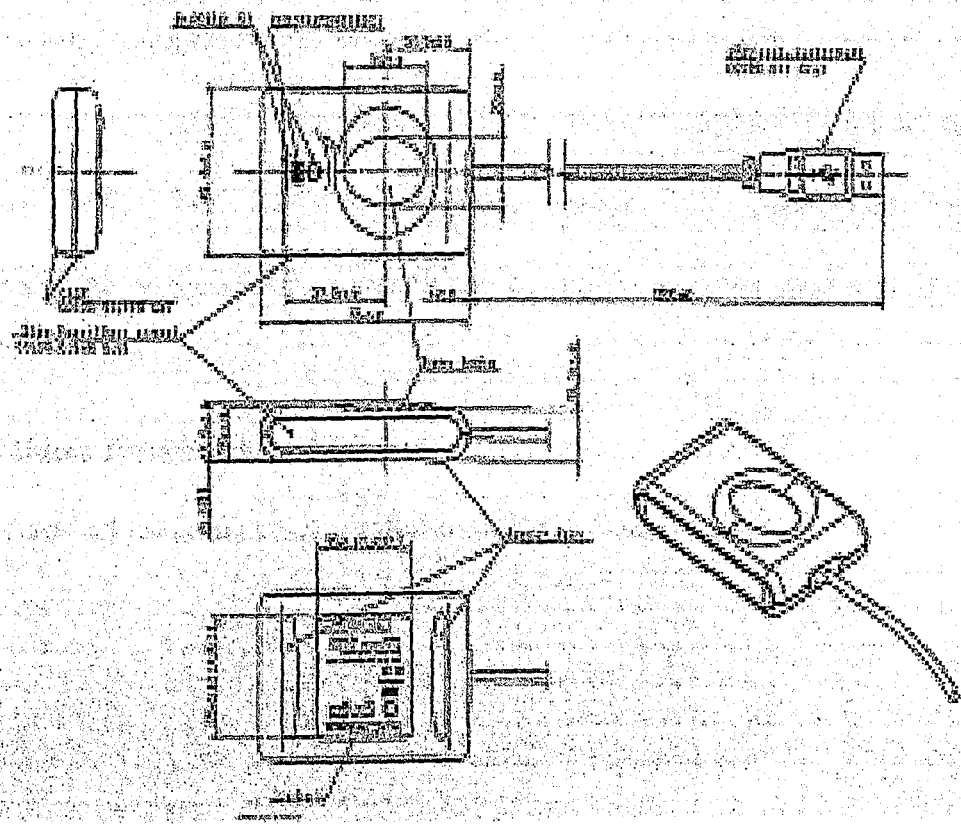
Part Numbering:

Part Number	Description
50256-001	DigitalPersona USB Fingerprint Sensor

Ratings:

Supply Voltage	5.0V +/- .25V supplied by USB
Supply Current – scanning	370 mA (typ.)
Supply Current – idle mode	120 mA (typ.)
Supply Current – suspend mode	2 mA (max.)
ESD Susceptibility	>15 KV
Temperature, Operating	5° - 35° C
Humidity, Operation	20% – 80% non-condensing
Temperature, Storage	-10° - 60° C
Humidity, Storage	20% - 90% non-condensing
Image data	8-bit grayscale
Standards Compliance	FCC Class B, CE, VCCI

Mechanical Specification:



APPENDIX C

A Brief Description of Visual Basic

Visual Basic is a high level programming language evolved from the earlier DOS version called Basic. Basic means Beginners' All-purpose Symbolic Instruction Code. It is a fairly easy programming language to learn. The codes look a bit like English Language.

Visual Basic is a Visual and events driven Programming Language. These are the main divergence from the old Basic. In Basic, programming is done in a text-only environment and the program is executed sequentially. In Visual Basic, programming is done in a graphical environment. Because users may click on a certain object randomly, so each object has to be programmed independently to be able to response to those actions (events). Therefore, a Visual Basic Program is made up of many subprograms, each has its own program codes, and each can be executed independently and at the same time each can be linked together in one way or another.

The Visual Basic Environment

Shown below is the start-up dialog box for Visual Basic.

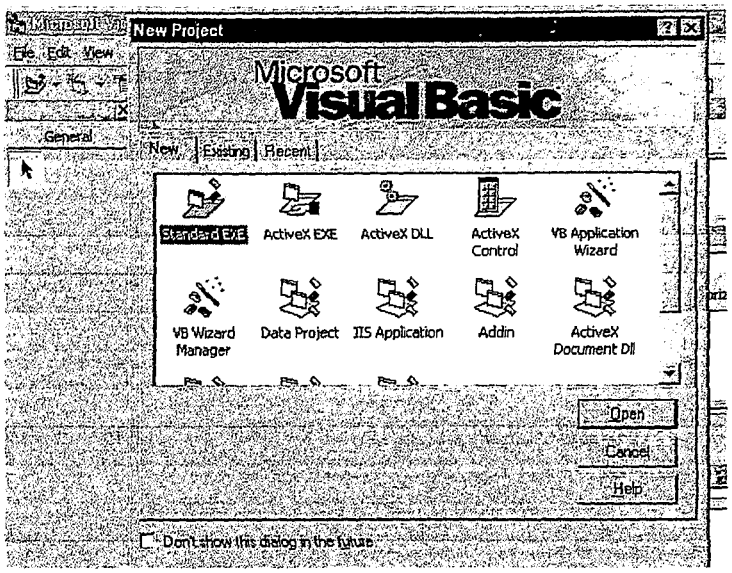


Figure 1: The Visual Basic Start-up Dialog Box

The following figure is the main window of the Visual Basic 6.

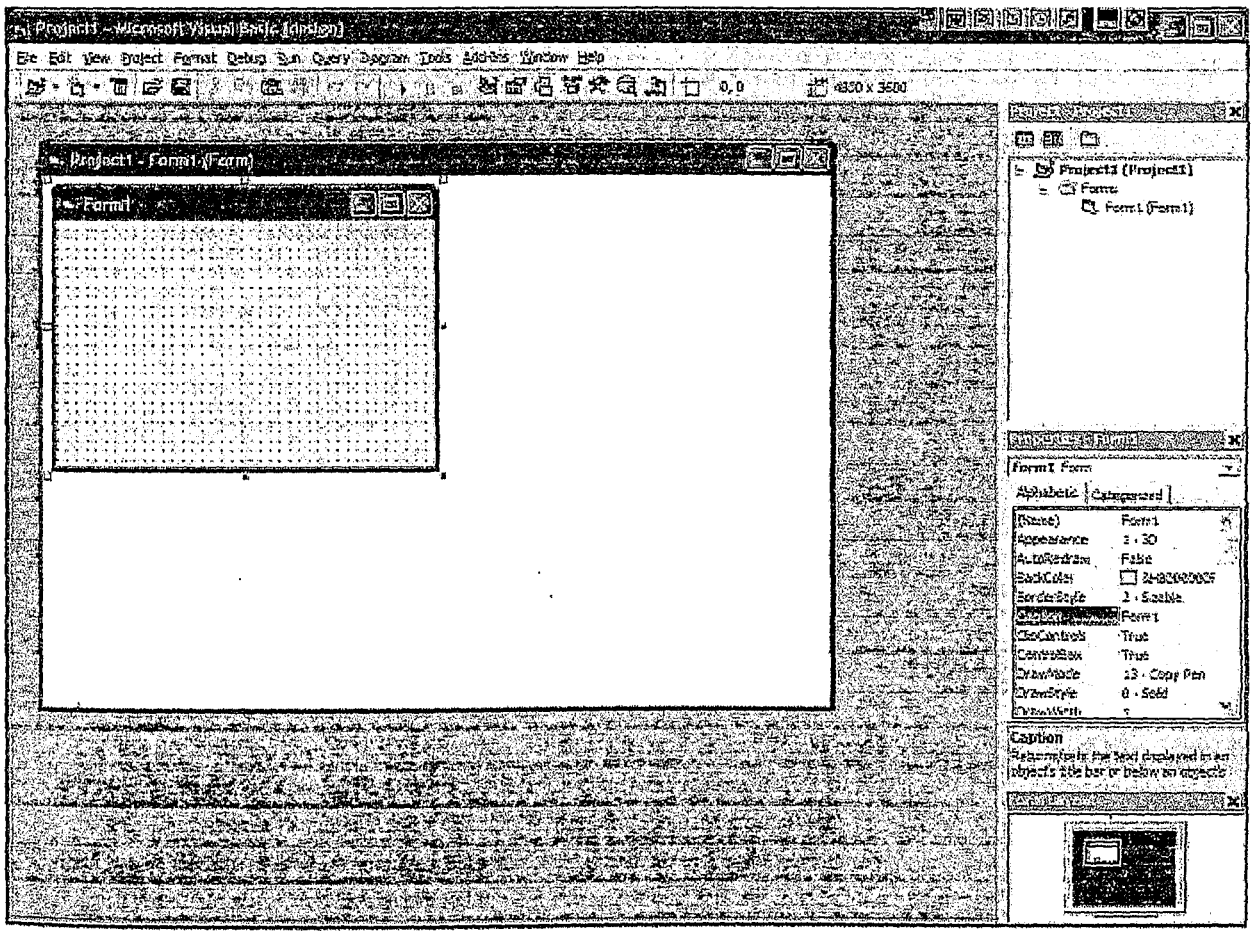


Figure 2: The Visual Basic main window

Visual Basic is a powerful programming language available nowadays. There are a lot of applications that can be created using Visual Basic programming for example interface system, games, data communication etc.

APPENDIX D

Sample of the interface programming language:

1) Form1 Code

```
Public globalcount As Integer, globalcountverify As Integer
Dim d As New Dos
Dim a As New abm54
Dim q As New Scan
Dim fso As New FileSystemObject
Public Is2K As Boolean
Public canABM As Boolean
Public ReturnValue
Option Explicit
' U.are.U dir      --- take images--- file1
' Enrollment dir    --- attrasoft\10001\ --- file2
' Verification dir  --- attrasoft\temp\ --- file3
```

```
Private Sub clear_Click()
Text1.Text = ""
End Sub

Private Sub Command3_Click()
Text1.Text = ""
init_light
a.results

End Sub
```

```
Private Sub Command1_Click()
  File2.Path = Form2.Text3.Text + Text4.Text
```

```
 Call File2.Refresh
 File3.Path = Form2.Text4.Text
 Call File3.Refresh
 q.loadImage1 (q.newlycaptured)
End Sub
```

```
Private Sub Command2_Click()
File2.Path = Form2.Text3.Text + Text4.Text
Call File2.Refresh
'Text1.Text = Text1.Text + Str(File2.ListCount)
 If File2.ListCount >= 1 Then
    q.loadImage2 (Form2.Text3.Text + Text4.Text + "\" + Form1.File2.List(0))
     End If
End Sub
```

```
Private Sub Command4_Click()
'Call a.verify_input(q.newlycaptured)
If canABM Then
   Call a.verify_Click(q.newlycaptured)
End If
End Sub
```

```
Private Sub Command5_Click()
Call verify_Click
Call Command4_Click
End Sub
```

```
Private Sub Command6_Click()
'd.enroll_Click
'q.enroll_Click2
q.verify_Click1
'd.envAD
End Sub
```

```
Private Sub end_Click()
q.closeQuickCam
End
End Sub
```

```
Private Sub enroll_Click()
d.enroll_Click
q.enroll_Click1
enroll.SetFocus
init_light
End Sub
```

```
Private Sub Form_Load()
'File1 is for the scanner working dir for delete purpose
'get program files
Is2K = False
If Is2K Then
'Call d.envPF
 Call d.envAD
End If
Call Form2.init
Call init_Click
ID.SetFocus
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)
Call Unload(Form2)
End
End Sub
```

```
Private Sub ID_Click()
  Call d.createFolderEnroll(Form2.Text3.Text + Text4.Text)
  init_light
```

```
End Sub

---

Private Sub init_Click()
'Text1.Text = ""
Call Form2.init1
globalcount = 1
globalcountverify = 1
If fso.FolderExists(Form2.Text2.Text) Then
    File1.Path = Form2.Text2.Text
Else
   MsgBox "Cannot find scanner Working Folder " + Form2.Text2.Text + ". You must
click Parameter Button and set the Parameter; then click Initiation button"
End If
Call test_Attrasoft
'If Not fso.FolderExists(Form2.Text4.Text) Then
   Call d.createFolderVerify(Form2.Text4.Text)
'End If
Call q.openScan
Call d.enroll_Click
Call ID_Click
End Sub

---

Private Sub parameter_Click()
Form2.Show
End Sub

---

Private Sub verify_Click()
Call init_light
'1 clean scan working dir
Call d.verify_Click

'2 capture an image
Call q.verify_Click1
```

```
'3 Verify
'Call a.verify_Click(q.newlycaptured)
'4 results
'Call a.results
End Sub

---

Private Sub init_light()
   Form1.red.FillColor = vbWhite
   Form1.green.FillColor = vbWhite
End Sub

---

Private Sub test_Attrasoft()
   '3. ImageFinder working dir
   If Not fso.FolderExists(Form2.Text3.Text) Then

      fso.CreateFolder (Form2.Text3.Text)
      Text1.Text = Text1.Text + "Create FVS Working Folder " + Form2.Text3.Text +
vbCrLf
      'MsgBox "Create FVS Working Folder " + Form2.Text3.Text
   End If

   '6. match.txt
   If Not fso.FileExists(Form2.Text6.Text) Then
      Text1.Text = Text1.Text + Form2.Text6.Text + " does not exist!" + vbCrLf
      Text1.Text = Text1.Text + "Please copy this file from your CD or disk!" +
vbCrLf
   End If

   ' 5. abm48.txt
   If Not fso.FileExists(Form2.Text5.Text) Then
      Call fso.CreateTextFile(Form2.Text5.Text)
      Text1.Text = Text1.Text + "Create " + Form2.Text5.Text + vbCrLf
   End If
```

```
    ' new a.txt
  Call fso.CreateTextFile("a.txt")
End Sub
```

---

2) Form2 Code

```
Dim s1 As String, s2 As String, s3 As String, s4 As String
Dim s5 As String, s6 As String
Dim fso As New FileSystemObject
Dim mFile As File
Dim y As TextStream
Public SPF As String
Public SAppData As String
Option Explicit
```

```
Private Sub Command1_Click()
s1 = Text1.Text
s2 = Text2.Text
s3 = Text3.Text
s4 = Text4.Text
s5 = Text5.Text
s6 = Text6.Text
Call Unload(Form2)
End Sub
```

```
Private Sub Command2_Click()
Call Unload(Form2)
End Sub
```

```
Private Sub Form_Load()
Call init1
End Sub
```

```
Public Sub init()
Dim s As String

s1 = "C:\Program Files\Vfinger\Vfinger.exe"

If Form1.Is2K Then
   s2 = SAppData + "My Documents\Vfinger\Images\"
Else
   s2 = "C:\My My Documents\Vfinger\Images\"
End If

s3 = "C:\attrasoft\"
s4 = "C:\attrasoft\temp\"
s5 = "abm48.txt"
s6 = "match.txt"

Text1.Text = s1
Text2.Text = s2
Text3.Text = s3
Text4.Text = s4
Text5.Text = s5
Text6.Text = s6

' read from init.txt
'Form1.Text1.Text = "Open init.txt " + vbCrLf
Call autoopen_initfile

End Sub
```

```
Public Sub init1()

Text1.Text = s1
```

```
Text2.Text = s2
Text3.Text = s3
Text4.Text = s4
Text5.Text = s5
Text6.Text = s6
End Sub
```

```
Public Sub open_initfile()
Dim s As String
Dim t0 As String, t1 As String, t2 As String, t3 As String, t4 As String
Dim t5 As String, t6 As String

s = "init.txt"

If fso.FileExists(s) Then
   Set mFile = fso.GetFile(s)
   Set y = mFile.OpenAsTextStream(ForReading)
   If Not y.AtEndOfStream Then
      t0 = y.ReadLine
   End If
   If Not y.AtEndOfStream Then
      t1 = y.ReadLine
   End If
   If Not y.AtEndOfStream Then
      t2 = y.ReadLine
   End If
   If Not y.AtEndOfStream Then
      t3 = y.ReadLine
   End If
   If Not y.AtEndOfStream Then
      t4 = y.ReadLine

   End If
```

```
If Not y.AtEndOfStream Then
   t5 = y.ReadLine
End If
If Not y.AtEndOfStream Then
   t6 = y.ReadLine
End If
y.Close
If Val(t0) = 0 Then
   Form1.Text1.Text = s + ": code = 0, file ignored!" + vbCrLf
End If

If Val(t0) = 1 Then
   Form1.Text1.Text = s + ": " + vbCrLf
   Form1.Text1.Text = Form1.Text1.Text + t0 + vbCrLf
   Form1.Text1.Text = Form1.Text1.Text + t1 + vbCrLf
   Form1.Text1.Text = Form1.Text1.Text + t2 + vbCrLf
   Form1.Text1.Text = Form1.Text1.Text + t3 + vbCrLf
   Form1.Text1.Text = Form1.Text1.Text + t4 + vbCrLf
   Form1.Text1.Text = Form1.Text1.Text + t5 + vbCrLf
   Form1.Text1.Text = Form1.Text1.Text + t6 + vbCrLf

   Text1.Text = t1
   Text2.Text = t2
   Text3.Text = t3
   Text4.Text = t4
   Text5.Text = t5
   Text6.Text = t6

   s1 = t1
   s2 = t2
   s3 = t3
   s4 = t4
   s5 = t5
```

```
        s6 = t6

    End If

    Else
        Form1.Text1.Text = Form1.Text1.Text + "Cannot Find init.txt" + vbCrLf
    End If
End Sub

Public Sub autoopen_initfile()
Dim s As String
Dim t0 As String, t1 As String, t2 As String, t3 As String, t4 As String
Dim t5 As String, t6 As String

s = "init.txt"

If fso.FileExists(s) Then
    Set mFile = fso.GetFile(s)
    Set y = mFile.OpenAsTextStream(ForReading)
    If Not y.AtEndOfStream Then
        t0 = y.ReadLine
    End If
    If Not y.AtEndOfStream Then
        t1 = y.ReadLine
    End If
    If Not y.AtEndOfStream Then
        t2 = y.ReadLine
    End If
    If Not y.AtEndOfStream Then
        t3 = y.ReadLine
    End If
    If Not y.AtEndOfStream Then
        t4 = y.ReadLine
```

```
End If
If Not y.AtEndOfStream Then
   t5 = y.ReadLine
End If
If Not y.AtEndOfStream Then
   t6 = y.ReadLine
End If
y.Close
   If Val(t0) = 2 Then
   Form1.Text1.Text = s + ": " + vbCrLf
   Form1.Text1.Text = Form1.Text1.Text + t0 + vbCrLf
   Form1.Text1.Text = Form1.Text1.Text + t1 + vbCrLf
   Form1.Text1.Text = Form1.Text1.Text + t2 + vbCrLf
   Form1.Text1.Text = Form1.Text1.Text + t3 + vbCrLf
   Form1.Text1.Text = Form1.Text1.Text + t4 + vbCrLf
   Form1.Text1.Text = Form1.Text1.Text + t5 + vbCrLf
   Form1.Text1.Text = Form1.Text1.Text + t6 + vbCrLf

   Text1.Text = t1
   Text2.Text = t2
   Text3.Text = t3
   Text4.Text = t4
   Text5.Text = t5
   Text6.Text = t6

   s1 = t1
   s2 = t2
   s3 = t3
   s4 = t4
   s5 = t5
   s6 = t6
End If
```

```
  Else
    Form1.Text1.Text = Form1.Text1.Text + "Cannot Find init.txt" + vbCrLf
  End If
End Sub
```

```
Public Sub save_initfile()
  Dim s As String
  s = "init.txt"
  Form1.Text1.Text = s + ": " + vbCrLf

  Call fso.CreateTextFile(s)
  If fso.FileExists(s) Then

    Set mFile = fso.GetFile(s)
    Set y = mFile.OpenAsTextStream(ForWriting)

    Call y.WriteLine("1")
    Form1.Text1.Text = Form1.Text1.Text + "1" + vbCrLf

    Call y.WriteLine(Text1.Text)
    Form1.Text1.Text = Form1.Text1.Text + Text1.Text + vbCrLf

    Call y.WriteLine(Text2.Text)
    Form1.Text1.Text = Form1.Text1.Text + Text2.Text + vbCrLf

    Call y.WriteLine(Text3.Text)
    Form1.Text1.Text = Form1.Text1.Text + Text3.Text + vbCrLf

  Call y.WriteLine(Text4.Text)
    Form1.Text1.Text = Form1.Text1.Text + Text4.Text + vbCrLf

  Call y.WriteLine(Text5.Text)
```

```
    Form1.Text1.Text = Form1.Text1.Text + Text5.Text + vbCrLf

  Call y.WriteLine(Text6.Text)
    Form1.Text1.Text = Form1.Text1.Text + Text6.Text + vbCrLf
    y.Close
  Else
    Form1.Text1.Text = Form1.Text1.Text + "Cannot Create init.txt" + vbCrLf
  End If
End Sub
```

```
Private Sub load_Click()
  Call open_initfile
End Sub
```

```
Private Sub save_Click()
Call save_initfile
End Sub
```

---

3) ABM52 Code

```
Public fso As New FileSystemObject
Public sLength As Integer
Dim mFile As File
Dim y As TextStream
Option Explicit
```

```
Public Sub verify_input(key As String)
Dim s As String
Dim s5 As String, s6 As String, s7 As String, s8 As String, s9 As String
Dim numOfLines As Integer

If fso.FileExists(Form2.Text6.Text) Then
  Set mFile = fso.GetFile(Form2.Text6.Text)
```

```
Set y = mFile.OpenAsTextStream(ForReading)
If Not y.AtEndOfStream Then
   s = y.ReadLine
End If
If Not y.AtEndOfStream Then
   s = y.ReadLine
End If
If Not y.AtEndOfStream Then
   s = y.ReadLine
End If
If Not y.AtEndOfStream Then
   s = y.ReadLine
End If
If Not y.AtEndOfStream Then
   s5 = y.ReadLine
   numOfLines = 5
End If
If Not y.AtEndOfStream Then
   s6 = y.ReadLine
   numOfLines = 6
End If
If Not y.AtEndOfStream Then
   s7 = y.ReadLine
   numOfLines = 7
End If
If Not y.AtEndOfStream Then
   s8 = y.ReadLine
   numOfLines = 8
End If
If Not y.AtEndOfStream Then
   s9 = y.ReadLine
   numOfLines = 9
End If
y.Close
```

```
'Call fso.CreateTextFile(Form2.Text5.Text)
If fso.FileExists(Form2.Text5.Text) Then

    Set mFile = fso.GetFile(Form2.Text5.Text)
    Set y = mFile.OpenAsTextStream(ForWriting)

    Call y.WriteLine("1")
    Form1.Text1.Text = Form1.Text1.Text + "1" + vbCrLf

    Call y.WriteLine(Form2.Text3.Text + Form1.Text4 + "\")
    Form1.Text1.Text = Form1.Text1.Text + Form2.Text3.Text + Form1.Text4 + "\"
+ vbCrLf
    Call y.WriteLine("1")
    Form1.Text1.Text = Form1.Text1.Text + "1" + vbCrLf

    Call y.WriteLine(key)
    Form1.Text1.Text = Form1.Text1.Text + key + vbCrLf

    Call y.WriteLine(s5)
    Form1.Text1.Text = Form1.Text1.Text + s5 + vbCrLf
    Call y.WriteLine(s6)
    Form1.Text1.Text = Form1.Text1.Text + s6 + vbCrLf
    If numOfLines >= 7 Then
       Call y.WriteLine(s7)
       Form1.Text1.Text = Form1.Text1.Text + s7 + vbCrLf
    End If
    If numOfLines >= 8 Then
       Call y.WriteLine(s8)
       Form1.Text1.Text = Form1.Text1.Text + s8 + vbCrLf
    End If
    If numOfLines >= 9 Then
       Call y.WriteLine(s9)
```

```
        Form1.Text1.Text = Form1.Text1.Text + s9 + vbCrLf
      End If
        y.Close
   End If
End If
End Sub

Public Sub verify_Click(key As String)

Dim x As Long
Dim s As String
Dim i As Integer

Call verify_input(key)

If fso.FileExists(Form2.Text5.Text) Then
   s = "attrasoft54 " + Form2.Text5.Text + " a.txt 1"
   Form1.Text1.Text = Form1.Text1.Text + s + vbCrLf
   x = Shell(s, 1)
   'Call results
End If
End Sub
```

```
Public Sub results()
Dim s As String
Dim i As Integer

sLength = 0
If fso.FileExists("a.txt") Then
   Set mFile = fso.GetFile("a.txt")
   Set y = mFile.OpenAsTextStream
   If Not y.AtEndOfStream Then
      s = y.ReadAll
```

```
    End If
    y.Close
    sLength = Len(s)
    If sLength > 0 Then
        Form1.Text1.Text = Form1.Text1.Text + s + vbCrLf
    End If
    Call analysis
End If
End Sub

Public Sub analysis()
    If sLength < 10 Then
        Form1.Text1.Text = Form1.Text1.Text + "Verification Negative" + vbCrLf
        Form1.red.FillColor = vbRed
    Else
        Form1.Text1.Text = Form1.Text1.Text + "Verification Positive" + vbCrLf
        Form1.green.FillColor = vbGreen
    End If
End Sub
```

---

4) DOS Code

```
Public fso As New FileSystemObject

Option Explicit
```

```
Public Sub openDos()

Dim x As Long
Dim s As String
Dim i As Integer

's = "COMMAND.COM"
'X = Shell(s, 1)
```

```
End Sub
```

```
Public Sub createFolderEnroll(s As String)
  If fso.FolderExists(s) Then
    Form1.Text1.Text = Form1.Text1.Text + s + " already exist!" + vbCrLf
    Form1.File2.Path = s
    Call Form1.File2.Refresh
    Form1.globalcount = Form1.File2.ListCount + 1

    If Form1.File2.ListCount >= 1 Then
      If fso.FileExists(s + "\" + Form1.File2.List(0)) Then
        Form1.Text1.Text = Form1.Text1.Text + "Display 1st: " + s + "\" +
Form1.File2.List(0) + vbCrLf
        Form1.Image2.Image= LoadImage(s + "\" + Form1.File2.List(0))
      Else
        Form1.Text1.Text = Form1.Text1.Text + "Cannot Find:" + s + "\" +
Form1.File2.List(0) + vbCrLf
      End If
    Else
      Form1.Image2.Image = LoadImage("")
    End If
      Exit Sub
  End If

  Call fso.CreateFolder(s)
  Form1.File2.Path = s
  Call Form1.File2.Refresh
  Form1.globalcount = 1
  Form1.Image2.Image = LoadImage("")
  Form1.Text1.Text = Form1.Text1.Text + "Create " + s + vbCrLf
End Sub
```

```
Public Sub createFolderVerify(s As String)

 If fso.FolderExists(s) Then
    Form1.Text1.Text = Form1.Text1.Text + s + " already exist!" + vbCrLf
    Form1.File3.Path = s
    Call Form1.File3.Refresh
    Form1.globalcountverify = Form1.File3.ListCount + 1
   Exit Sub
 End If

 Call fso.CreateFolder(s)
 Form1.File3.Path = s
 Call Form1.File3.Refresh
 Form1.globalcountverify = 1
 Form1.Text1.Text = Form1.Text1.Text + "Create " + s + vbCrLf
 '    Form1.Text1.Text    =    Form1.Text1.Text    +    "createFolderVerify    "    +
Str(Form1.globalcountverify) + vbCrLf
End Sub
```

```
Public Sub enroll_Click()

Dim i As Integer, NumofFiles As Integer, x As Integer
Dim s As String
Call Form1.File1.Refresh
NumofFiles = Form1.File1.ListCount
   For i = 0 To NumofFiles - 1
      s = Form2.Text2.Text & Form1.File1.List(i)

      If fso.FileExists(s) = True Then
         Call fso.DeleteFile(s, True)
         Form1.Text1.Text = Form1.Text1.Text + "Delete " + s + vbCrLf
      End If
   Next i
```

```
End Sub
```

```
Public Sub verify_Click()

Dim i As Integer, NumofFiles As Integer, x As Integer
Dim s As String
Call Form1.File1.Refresh
NumofFiles = Form1.File1.ListCount
  For i = 0 To NumofFiles - 1
    s = Form2.Text2.Text & Form1.File1.List(i)
    Form1.Text1.Text = Form1.Text1.Text + "Delete " + s + vbCrLf
    If fso.FileExists(s) = True Then
      Call fso.DeleteFile(s, True)
    End If
  Next i
End Sub
```

```
Public Sub envPF()
Dim EnvString, Indx, Msg, PathLen
Indx = 1   ' Initialize index to 1, start from 1.
Do
  EnvString = Environ(Indx)
  If Left(EnvString, 13) = "ProgramFiles=" Or Left(EnvString, 13) =
"PROGRAMFILES=" Then   ' Check PATH entry.
    PathLen = Len(Environ("ProgramFiles"))   ' Get length.
    Msg = "PATH entry = " & Indx & " and length = " & PathLen &
Environ("ProgramFiles")
    Form2.SPF = Environ("ProgramFiles") + "\"
    Form1.Text1.Text = Form1.Text1.Text + "Program Files =" +
Environ("ProgramFiles") + "\" + vbCrLf
    Exit Do
  'Else
  '  Indx = Indx + 1   ' Not PATH entry,
```

```
  End If   ' so increment.
  Indx = Indx + 1
  'Form1.Text1.Text = Form1.Text1.Text + EnvString + vbCrLf
Loop Until EnvString = ""
If PathLen <= 0 Then
  MsgBox "This is Windows 2000 Version."
End If
End Sub
```

```
Public Sub envAD()
Dim EnvString, Indx, Msg, PathLen
Indx = 1   ' Initialize index to 1, start from 1.
Do
  EnvString = Environ(Indx)
 If Left(EnvString, 12) = "USERPROFILE=" Then   ' Check PATH entry.
    PathLen = Len(Environ("USERPROFILE"))   ' Get length.
    Msg = "Index = " & Indx & " and length = " & PathLen &
Environ("USERPROFILE")
    Form2.SAppData = Environ("USERPROFILE") + "\"
    Form1.Text1.Text = Form1.Text1.Text + "USER PROFILE =" +
Environ("USERPROFILE") + "\" + vbCrLf
    Exit Do
  'Else
  '  Indx = Indx + 1   ' Not PATH entry,
  End If   ' so increment.
  Indx = Indx + 1
  'Form1.Text1.Text = Form1.Text1.Text + EnvString + vbCrLf
Loop Until EnvString = ""
If PathLen <= 0 Then
  MsgBox "This is Windows 2000 Version."
End If
End Sub
```

```
    idToString = "hello"
End Function
```

---

5) Init.txt File

```
0
C:\Program Files\Logitech\ImageStudio\ImgStud.exe
C:\My Documents\ImageStudio\Album\Pictures and Videos\
C:\attrasoft\
C:\attrasoft\temp\
abm48.txt
match.txt
```

APPENDIX E

Image Matching Engine (ImageFinder 5.4 DOS version)

A brief discussion has been given about Image Matching Engine in the previous section. As stated before ImageFinder 5.4 uses Boltzmann Machine neural network for the implementation of its system. ImageFinder 5.4 can be used for any type of images. Technical specification of the module is:

Finger scanner: DigitalPersona 2000 U.are.U fingerprint scanner

Finger scanner driver: U.are.U Integrator 2.1.0

Interface software: FVS.exe and match.txt (batch code)

Image Matching Engine: Attrasoft54.exe (ImageFinder for DOS)

The operation of the engine consists of two procedures; Enrollment and Verification. The *enrollment* procedure takes the fingerprint image of a person and stores them with an ID. The *verification* procedure verifies the person's identity with his ID. Each procedure involves two steps and the first step for both procedures is exactly the same.

In the enrollment process, a fingerprint is captured and sent to the ImageFinder to be trained and stored by the software in a directory. The parameters for the training of the image are as follows (the number in the bracket represents the batch code):

1) Primary Parameters: Training

- **Background**: The background should be set which make the black area of the image as small as possible, as long as it covers the key-segment.
- **Area of Interest (AOI)**: Use image-segment for searching similar images. Only use the whole image for exact match.

- **Symmetry**: Symmetry of invariance means similarity under certain types of conditions. For example, two images, one with a face in the middle and the other with the face moved to the edge; we say these two images are similar because of the face.

  There are five symmetry setting

  - No symmetry (0)
  - Translation symmetry (3)
  - Scaling symmetry (4)
  - Rotation Symmetry (5)
  - Rotation and scaling symmetry (6)

  However, there is an effect to the performance of the system if the parameter symmetry is used. Symmetry will make the verification job longer. However, there would not be any problem to use symmetry when operated on a computer with a large RAM and fast CPU.

- **Segment Cut:** The range of segment cut is between digit 0 and 12. Parameter segment cut deals with the edges of the segments in the images. The larger this parameter is, the smaller the segment the ImageFinder will use. This parameter is used for both training and searching.

1) Secondary Parameters: Training

- **Internal Representation of Images:** Default representation of the image is 100 x 100. The search speed crucially depends on this setting. If the internal representation is reduced to 50x50 then the underlying neural net size is reduced by a factor of 4; and the neural computation speed will be

decreased by a factor of 16. However throughout the training of the software this parameter can have only one value.

- **Reduction type:** The image will be reduced by integer, real or all images are reduced by a same amount. The default representation is integer + average.
- **Translation type:** Translation type can be defined as the accuracy of the translation symmetry. The settings are
    - Most accurate (0)
    - Accurate (1)
    - Least accurate (2)
- **Scaling type:** Scaling type can be defined as the accuracy of the scaling symmetry. The settings are
    - Least accurate (0)
    - Accurate (1)
    - Accurate (1)
    - Most accurate (3)
- **Rotation Type:** Rotation type can be defined as the accuracy of rotation symmetry. The settings are
    - 360° rotation, least accurate (0)
    - -5 ° to 5 ° rotation (1)
    - -10 ° to 10 ° rotation (2)
    - 360° rotation, accurate (3)
    - 360° rotation, more accurate (4)
    - 360° rotation, most accurate (5)

- **Border Cut:** Border Cut range from 0 (no cut) to 9 (18% border cut). The purpose of this parameter is to get rid of the border section.
- **The Edge Filter:** The purpose of this parameter is to extract and enhance edges and contour in an image by expressing intensity differences, between neighboring pixels as an intensity value.

The parameters for matching are as follows:

1) Primary Parameters: Matching

1. **Sensitivity:** The ranges of the sensitivity parameters are 0(least sensitive) to 100 (most sensitive). This parameter is set by the following rules;
    - To search small segment(s), use HIGH sensitivity,
    - To search large segment(s), use LOW sensitivity
    - When search yields no results, increase sensitivity
    - When search yields too many results, decrease sensitivity

   Default setting for this parameter is 50.
2. **Blurring:** The range of the blurring parameters is 0% to 20%. The "0%" blurring means the exact match. If we increase "Blurring" more similar images will appear. By increasing this parameter also will slow the speed slightly. This parameter is set by the following rules;
    - When search yields no results, increase sensitivity
    - When search yields too many results, decrease sensitivity

   Default setting for this parameter is 5%.

- **Shape Cut:** This parameter deals with the shape of the image. The range of the parameter is 0(shapes of image must exactly match the training image) to 100(any shape of image will be search).

- **Internal/External Weight Cut:** The purpose of this parameter is to list only those retrieved images with scores or weight greater than a certain value (threshold). The range of internal cut is from 0 to 99; the external cut can be any numbers. The internal cut stops the images from coming out where as the external cut can bring the eliminated images back if you set the external cut to 0.

  Example: We want to search for images and all similar images have weights ranging from 1000 to 10,000. It is possible that some other images pop up with weights ranging from 10 to 100. To eliminate these images, you can set the "External Weight Cut" to 1000.

2) Secondary Parameters: Matching

- **Image Type:** There are six types of Image Type
  - Bi-level 1 (0)
  - Bi-level 2 (1)
  - Bi-level 3 (2)
  - Color 1 (3)
  - Color 2 (4)
  - Color 3 (5)

The meaning of some of the type is

Bi-level 1 – like and integration function f(x) [sum search]

Bi-level 2 – like maximum value of f(x) [maximum search]

Bi-level 3 – is in the middle [average search]

Color 1 – produce higher weight than Color 2

Color 3 – is in the middle.

- **Large (L)/Smart (S) Segments:** Currently S-segment only support translation symmetry. For the rotation or/and scaling symmetry an L-segment is used.
- **Short and Long Search:** Limit of the short search is 1000. For the long search there is no technical limit.
- **Shape Cut Type:** There are two shape cut parameters; shape cut and shape cut 2
- **Image Dimension:** This parameter is used to search images of certain dimension and ignore other images.
- **Sample Match:** This parameter deals with training using an image segment. The ImageFinder will use a small segment to train and then search a directory. The sample match will provide a typical match to the training segment. This parameter needs to be set before training the software start.
- **Short Cut Scrollbar:** A quick way to set the parameters. Use scrollbar to select a number between 0 and 99 where 0 means the most accurate and 99 means the least accurate search.

Batch File

In ImageFinder for DOS we will deal with batch file for the interface of the scanner and the Image Matching Engine. The batch file specifies the Image Finder setting. The organization of batch code is as illustrated in Table 4.0

The first integer, N, indicates how many times we will search through the search directory. For the *search, identification and verification* problems which is 1: M matching, N = 1 is used. For the other applications such as classification problems (N: M Matching), N should be less than or equal to 30.

The second line specifies the search directory (parameter 1 of 27). After the second line, you should have N block for N searches. Each block has 3 or more lines to specify other parameters;

- The first line specifies how many sample images are in the current search. If we have 1 sample, there will be 2 more lines in this block; if we have 2 samples; there will be 4 more lines below. Each sample is specified by 2 lines.
- The second line specifies a sample image (parameter 2 of 27)

```
N
Search directory
M
Sample image
        Segment (x,y;w,h); Background
        Rotation Type; Reduction Type; Training symmetry
        Sensitivity; Blurring;

        External Cut; Image Type; Segment Size;
        Representation; Shape Cut;

        Internal Weight Cut; Short/Long Search; Segment Cut;
        Batch File Type; Image Dimension;

        Translation Type; Scaling Type; Shape Cut II; Shape
        Type; Border Cut;

        Edge Filter; Look-at Window (x1, y1; w1, h1)
End
```

Organization of the batch code

- The third line specifies the other 25 parameters;
    - Segment (x, y; w, h) (default is 0,0,0,0)
    - Background;
    - Rotation Type;
    - Reduction Type;
    - Training Symmetry;

- Sensitivity;
- Image Type;
- Segment Size;
- Representation;
- Shape Cut;
- Internal Weight Cut;
- Short/Long Search;
- Segment Cut;
- Batch File Type;
- Image Dimension
- Translation Type;
- Scaling Type;
- Shape Cut II;
- Shape Type;
- Border Cut;
- Edge Filter;
- Look-At Window (x_1, y1; w_1, h1)

The discussion of all this parameters has been done in the earlier section. The sample of the batch file code is shown,

```
1
Engin
1
Engin\Surize.jpg
        75 75 155 155 5 0 0 3 45 10
        24000 1 0 0 40 40 0 0 0 0
        0 0 1 0 0 0 0 0 0 0
```

Example of Batch File Codes

Batch Commands

The ImageFinder for DOS needs to be operated from the C:\ > prompt. Below are four possible commands for the ImageFinder:

C:\> attrasoft54

C:\> attrasoft54 A

C:\> attrasoft54 A B

C:\> attrasoft54 A B C D

Where

A is the input batch file

B is the output file

C id the batch file commands type

D is for debug or starting the ImageFinder for Windows.

Example for the four argument command i.e. C:\> attrasoft54 A B C D is as shown;

Input Batch File: A

Output Batch File: B

Command Type: if C = 0; Search + Sort

C = 1; Batch (N) Command

C = 2; Batch (1) Command

C = 3; Fast Batch (1) Command

C = 4; Multiple-Search Batch Command.

From a different angle, for

Search, Verification and Identification problem use C = 0,1,2

Large Classification Problem use C = 1

Small Classification Problem use C = 2

Fast Classification Problem use C = 3

Long Search use C = 0, 1, 2

Multiple Search C = 4

Debug;

D = 0; No Debug

D = 1; Open the ImageFinder for Windows.

The example of the execution of ImageFinder for DOS program is shown below,

```
C:\Program Files\Attrasoft\ImageFinder 5.4>attrasoft54 Suriza.txt aout.txt 1 1

When the execution stop, we will see
C:\Program Files\Attrasoft\ImageFinder 5.4>attrasoft54 Suriza.txt aout.txt 1 1
Batch File : suriza.txt
Search Start
Search End!
Sort Start
Sort End!
C:\Program Files\Attrasoft\ImageFinder 5.4>
```

Simple operation of ImageFinder for DOS

APPENDIX F

Image is rotated vertically

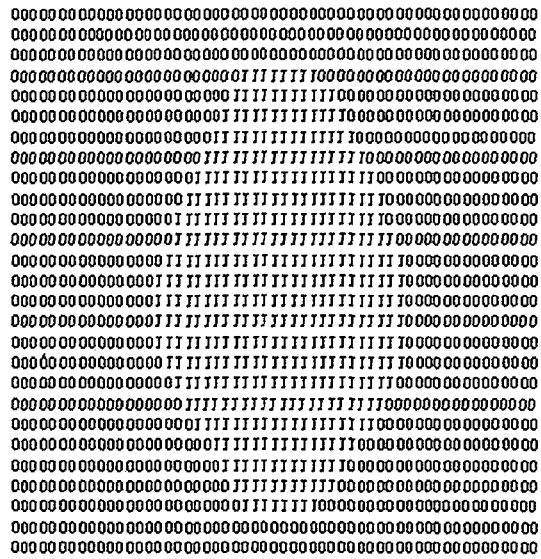


Image is skew 10°

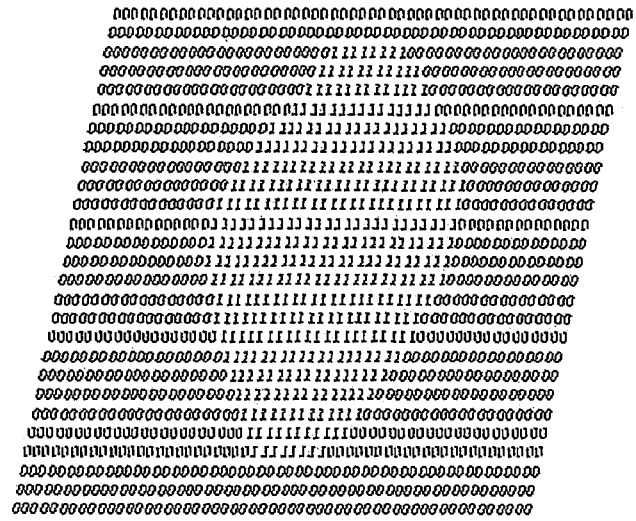


Image is rotate 90°