Khartoum University
Faculty of Mathematical Sciences

Offline Signature Verification for Arabic Language

Howayda Sid Ahmed Mohammed-Ali

Supervisor
Prof. Izzeldin Kamil Amin

Submitted to the Faculty of Gradate Studies in fulfillment for a Ph.D Degree in Computer Sciences

July 2006

Acknowledgements

In humble words I would like to express my profound gratitude to the people and institutions whose help and guidance has resulted in achieving the objectives of this work.

Among those, most prominent in contribution, is my supervisor Dr. Izzeldin Kamil Amin who has been an unfailing source of help and encouragement. Without his considerable support this work could never have emerged.

My gratitude is also extended to my colleagues Hassan A.Malik and Muhanned A.SidAhmed who have both provided me with necessary references unavailable for me, during the course of my study.

I acknowledge the moral support I received from my parents, family and most of all my two kids (Ismat & Rahaf) who, despite their occasional annoyance, they have created relaxing atmosphere.

My thanks go to the colleges of Graduate Studies and to the Faculty of Mathematical sciences, both of the University of Khartoum for giving me the opportunity to do this research.

The full record of debts to friends and colleagues is difficult to compile.

Yet the shortcoming of this work remains mine.

Abstract

Biometrics relies on biological features (e.g. finger print, iris or the retina) or behavioral features (voice, signature). Those features can be used for identity verification for an individual. For this it became one of the most trusted and natural ways to identify a person and controlling access to the systems.

Signature is a behavioral biometric. Signature is not unique like iris or finger print as it can be forged. Automatic signature verification is divided into two areas depending on the way of data capturing: offline and online signature verification. In offline signature verification, the signature is scanned from a document using a scanner to get the image of the signature. In online signature, a digitizing tablet is used to collect the movements during the signing.

In this work we present a system for offline signature verification. In this system the user has to submit a number of signatures which are used to extract two types of features, statistical features and structural features. A vector obtained from each of them is used to train propagation neural net in the verification stage. A test signature is then taken from the user, to compare it with those the net had been trained with.

A test experiment was carried out with two sets of data are collected. One set is used as a training set for the propagation neural net in its verification stage. This set with four signatures form each user is used for the training purpose. The second set consisting of one sample of signature for each of the 20 persons is used as a test set for the system. A negative identification test was carried out using a signature of one person to test others' signatures. The system gave encouraging results.

ملخـــص

يرتكز الإحياء القياسي (Biometrics) على خواص إحيائية (بيولوجية) مثل بصمات الأصابع وقزحية العين وشبكية العين, كما يعتمد على خواص سلوكية مثل نبرات والصوت هذه الخواص يمكن استخدامها كسمات مميزة للفرد وبالتالي أصبحت أكثر الوسائل العملية مصداقية لتمييز الفرد والتحكم في الدخول للنظام.

بحكم هذا التصنيف فان التوقيع ظاهرة سلوكية، غير انه، خلافاً للقزحية و الشبكية فهو ليس خاصية متفردة إذ يمكن تقليده أي تزويره.وتقسم المطابقة التلقائية إلى قسمين استنادا على طريق جمع التوقيعات المطابقة المباشرة(online signature verification) والمطابقة غير المباشرة offline signature verification. في الحالة الأولى يتم استخدام digitizing tablet جهاز لجمع التوقيعات مباشرة واستخلاص الخواص المطلوبة مباشرة خلال عملية التوقيع.أما في الحالة الثانية يتم مسح ضوئي (scanning) للتوقيع الموجود على وثيقة ما لأخذ صورة (image) للتوقيع يتم استخلاص الخواص منها لاستخدامها في عملية المطابقة.

نحاول في هذا البحث تقديم Offline signature verification system في هذا النظام على المستخدم أن يقدم مجموعة من التوقيعات التي تستخدم لاستخلاص نوعين من الخواص: خواص إحصائية (statistical features) وخواص تركيبية (هيكلية) structural.

ويستخدم المتجه(vector) المتحصل عليه من كل من هذه الخواص في تدريب شبكة عصبية اصطناعية propagation neural network لاستخدامها في المطابقة، حيث يؤخذ التوقيع المراد مطابقته لمقارنته بتلك التوقيعات التي تم تدريب الشبكة عليها.

أخذت ثلاث مجموعات من التوقيعات من المستخدمين. شملت المجموعة الأولى 20 مستخدما" وتم أخذ 4 توقيعات من كل مستخدم لاستخدامها في تدريب الشبكة. أما المجموعة الثانية فهي تحوي توقيع واحد من كل مستخدم لاستخدامه لاختبار النظام للتعرف على صاحبه.

المجموعة الثالثة حوت تواقيع مزورة لعرضها على النظام للتأكد من مقدرة النظام على كشفها وقد كانت النتائج مرضية فيما نحسب. قمنا كذلك بعرض تواقيع حقيقية لأفراد آخرين على النظام لنتيقن من مقدرة النظام على تمييزها ورفضها تلقائيا إمعانا منا في التأكد من سلامة النظام.

كما أجري الاختبار للتأكد من تعرف النظام فقط على توقيع الفرد المعني باستخدام توقيع فرد آخر كتوقيع مزيف لقد كانت نتائج النظام مرضية فيما نحسب.

Table of Contents

List of Figures

List of Tables

Introduction

Signature verification is very important in realizing banking and networking systems where signatures can be used to identify the person who signs. An automated verification process would enable banks and other financial institutions to significantly reduce cheque and money order forgeries, which account for a large monetary loss each year. Reliable signature verification can be of great use in many other application areas such as law enforcement, industry security control and so on.

The Motivation:

- That handwritten signature is still needed as identification to identify a person on all official matters and to establish his/her authenticity (in cheques, contracts, certifications…etc).
- Handwritten signature is not an established biological feature; rather it is a behavioral one. Unlike fingerprints, its recognition is not always easy or certain. Therefore it can be forged and hence the identity of the signer can be forged.

The objectives of the research are:

- To develop a computer system to solve or at least to ease the problem of handwritten signature verification.
- Our objectives are confined to offline signature verification as it is easy to forge due to variations found in a genuine signature resulting from behavioral factors.

If somebody is asked to handwrite his signature 100 times, no two of these signatures will be identical unless by mere chance: but all of them will be similar with a varying degree of similarity ranging between 1 to 99%. One of our objectives here is to raise that degree to the maximum.

Handwritten signatures appear on many types of documents such as bank cheques and credit slips etc. The large volume of such documents makes automatic signature verification desirable. A system for signature verification requires high reliability. Lots of efforts have been focused on the investigation of automatic signature verification methods. Signature authentication deals with verifying whether a given signature belongs to a person whose signature characteristics are known in advance in the form of extracted features. These features are fed to neural network that is trained on it and then used as a comparison tool to classify which signature is genuine or fraud.

A system was developed and when tested it proved some degree of efficiency. At least good than manual system of human examiners.

The system may give two types of errors:

- *False rejection Rate* (FRR), in which genuine signature is rejected as a forged signature.
- *False acceptance Rate* (FAR), in which forged signature is accepted as a genuine signature.

The project has some recommendations:

- The system needs to incorporate a filtering techniques to avoid all 'noise' usually appear in offline signatures such as pen size, ink, signer state of mind…etc will contribute towards the betterment of the system.
- The system need a robust method for filtering ,such as a good use of closing and opening operator to solve the problem arise during experiment in which the system delete the line or dot below the word.

- A trail is needed to get the upper and lower contour using another method which uses a grid to take the peaks and valley of the shape of the signature into consideration. In addition, a method to deal with rotated images of a signature is needed.
- Our sets used for the test does not contain a set of real skilled forge data (no way to get this type of data). The result would be more valuable if true skilled forgeries available.

The data presented here to deal with this matter is divided over an introduction and six chapters. Figures and illustrations together with a list of references are also provided

Chapter One deals with the biometrics and its roll in identification and verification.

Chapter Two deals with the signature as identification and describes the different types of the signature systems.

Chapter Three gives an overview of image processing.

Chapter Four deals with neural networks, its types and how it is used in signature verification systems.

Chapter Five describes the system developed for offline signature verification for Arabic handwriting.

Chapter six presents the conclusion and highlights possible future work in form of recommendations.

Chapter One

Biometrics: An overview

This chapter is meant to deal in a very general manner with biometrics as a subfield of pattern recognition, its function and the role it play in identification and verification.

1.1 Introduction

As time goes on the world is getting smaller. Technology, as the art of the study of development of industrial skills, is playing the major, if not the only role, in this phenomenon.

In a growing world of forgery, falsity, fraud and deception; accuracy and precision are becoming a necessity to segregate between true and false and to recognize validity, not to eliminate crime but to reduce it.

One line of thought is biometrics which can measure the degree of variations involved in a given set of characters, whether signature, finger print, voice, iris etc…

Daily in our live, we need to verify an individual identity, in other words, to determine who some one is is. This means we need identification and authentication (verification).

Identification is the process for establishing the identity of the individual. Authentication (verification) is the process by which a system establishes that an identification assertion is valid [10].

Now many systems used to automate the authentication process which brings greater security and convenience to our life. In authentication (verification mechanism) we use 3-techniques to do that:

1. Something you know (e.g. password)
2. Something you have (e.g. cards)
3. Something you are (biometrics)

Passwords systems work effectively as long as it is not guessed or, otherwise, disclosed to potential adversaries through accident, or sharing [13]. May be we can reduce risk of guessing or finding out, but no way to

prevent sharing. To eliminate the weakness happen with passwords they develop the smart cards, which carry a token with passwords to verify the person given that card; but a problem here arises: if the card is stolen no one can verify whether or not this card is used by the right person,

To overcome most of these disadvantages biometric techniques are used.

1.2. What is Biometrics?

The word biometric [34] is derived from the Greek roots – *bios* for life and *metrikos* for measurement. It stands for any characteristic that distinguishes living individuals from one another and can be measured such that a comparison is possible.

English dictionary define biometrics as “the measurements of life for statistical actuarial purpose” and “static applied to problems of biology, particularly variations”.

The international Biometrics Industry Association defines it as "automated methods for verifying or identifying the identity of a living individual based on physiological or behavioral characteristics" [16]

Also biometrics is the science of identifying or verifying the identity of person based on physiological or behavioral features Fig [1.1]. Physiological features include finger-prints and facial image .The behavioral characteristic are signature and voice and they depends on physical characteristic since they are actions done by a person. Others defined it as it is "positive personal identification", because they are claimed to provide greater confidence that the identification is accurate [16].

Biometrics, take the measurement from a person and make such a procedure to compare the given data with that collected before. This technique has two applications, identification and verification. Identification means "The act or process of establishing the identity of a

person, or recognizing him" or "the treating of a thing as identical with another [16].

Why Use Biometrics?

Over thousands of year's biometrics was used for identification. In Egypt's Nile Valley, traders were formally identified based on physical characteristics such as height, eye color and complexion. There are many reasons for popularity of biometrics such as [12]:

1. Increased need for verification: since it offers some method to identify a person other than password and PIN, which it can be stolen or lost.
2. Convenient verification: In biometrics the characteristics are used as identifier, so there is nothing to lose or forget. Also it offers easy and quick verification systems than that using cards or PIN.
3. Decreased Cost: Advancement in computing power, networking and database systems have allowed biometrics geographical and network areas.

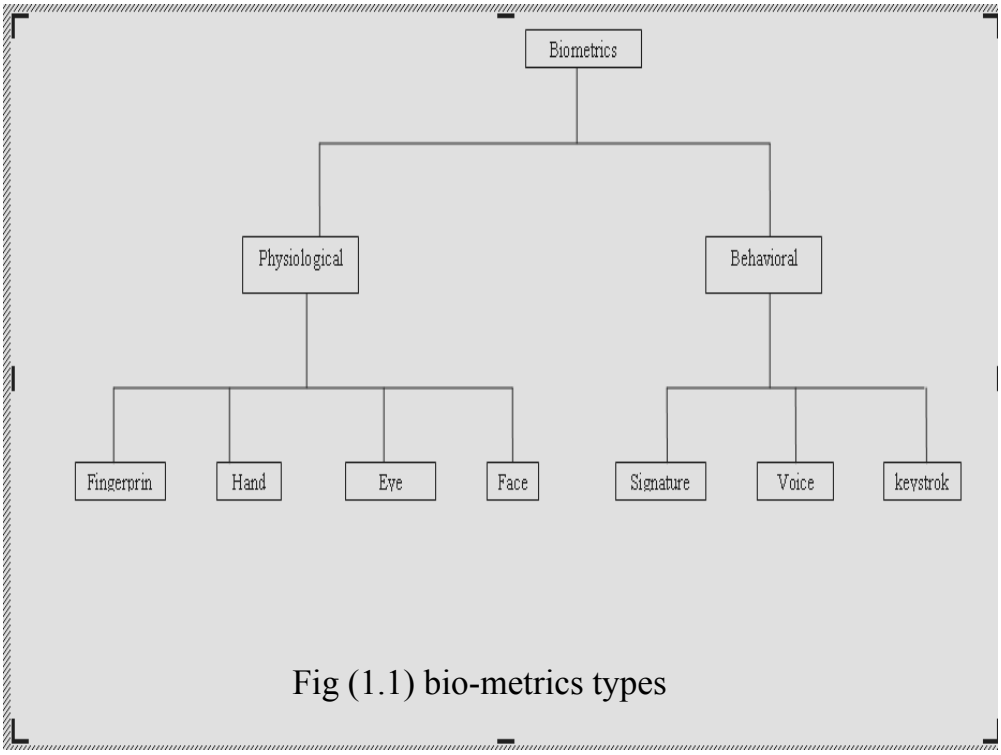


Fig (1.1) bio-metrics types

1.3. Biometric Systems

There are two types of questions that can be posed giving rise to two categories of such problems: Recognition (or identification) and verification. In the recognition problem, we ask the question: “Who is this?” we compare the unknown biometric (say face) with stored templates or models corresponding to the faces of the individuals registered in the database, and selects the one that matches best. If faces that are not in the database are to be permitted, an artificial class of “other” may be created to close the set of faces. In the verification problem, we ask the question: “Is this X?” when presented with the biometric from an unknown person and a claim to be X. A comparison is then made to the template or model for X, and possibly a background (or “other”) template or model.

When choosing a biometric for an application one of the important question must be asked: Does the application need verification or identification? If an application requires an identification of a subject from a large database, it needs a scalable and relatively more distinctive biometric (e.g., fingerprint, iris, or DNA).

A biometrics system is an automatic identification or verification of an individual by using a behavioral or physical characteristic of that person .It may be called either a verification system or an identification system.

- A verification system verify a person identity by comparing the captured biometric characteristic with that already saved in a data base .A verification system may accept or reject these entered data.
- An identification system recognizes a person by searching the data base for a match. In an identification system, the system

establishes a subject's identity without the subject having to claim the identity [3].

A simple biometric system can be designed in the following steps:

- Acquisition
- Enhancement (enrolment)
- Feature extraction
- Verification

Acquisition: In this stage a biometric data is collected using special biometric reader for each type, and stored in a data base. The performance of a biometric system is largely affected by the reliability of the reader used. Further, if the biometric trait or being measured is noisy, the result given by the comparison tool is not reliable

Enhancement (enrolment): some times it is called preprocessing, since the hardware or the medium on which data is captured may introduce a noise, we need to remove it then a reference to get values for the corresponding image this is enhancement (enrolment).

Feature extraction: Here many processes should be done to extract the feature value.

Verification: Here a method of comparison is used, in which the features of the test value is compared against those are stored in the data base. After the comparison we need a decision if the tested value is accepted or rejected. Special measurements are used for this.

1.3.1. Measures of Performance of Biometric Systems

In evaluating a performance of a biometrics system there are two important factors: (a)False Rejection Rate (FRR) which occurs when the system reject an authorized user, and (b) False Acceptance Rate (FAR) which occurs when the system accept an unauthorized user. The probability of these errors varies with the threshold. In general, one type

of error decreases at the cost of an increase in the other. Usually false acceptance and false rejection errors are plotted as percentages or probabilities.

Also we can speak about the point where FAR=FRR which is called Equal Error Rate (EER).

To evaluate the performance we have to calculate each of (FRR) and (FAR), and since they are related lowering of one of them will increase the other's value.

1.3.2. A Comparison of Various Biometrics

A human physiological or behavioral characteristic can be used to identify a person if it satisfies some requirements [3]:

- *universality:* means each person should have the biometrics;
- *distinctiveness*: which indicates that any two persons should be sufficiently different in terms of their biometrics identifiers;
- *permanence:* which means that the biometric should be sufficiently invariant (with respect to the matching criterion) over a period of time.
- *collectability:* which indicates that the biometric can be measured quantitatively?
- *performance :* which refers to achievable recognition accuracy, such as speed, robustness. The resource requirements to achieve the desired recognition accuracy and speed, as well operational or environmental factors that affect the recognition accuracy and speed;
- *acceptability:* which indicates the extent to which people are willing to accept a particular biometric identifier;
- *circumvention:* this refers to how easy it is to fool the system by fraudulent methods.

Biometric identifier	Universality	Distinctiveness	Permanence	Collectability	Performance	Acceptability	Circumvention
DNA	H	H	H	L	H	L	L
Ear	M	M	H	M	M	H	M
Face	H	L	M	H	L	H	H
Facial thermogram	H	H	L	H	M	H	L
Fingerprint	M	H	H	M	H	M	M
Gait	M	L	L	H	L	H	M
Hand geometry	M	M	M	H	M	M	M
Hand vein	M	M	M	M	M	M	L
Iris	H	H	H	M	H	L	L
Key stroke	L	L	L	M	L	M	M
Odor	H	H	H	L	L	M	L
Retina	H	H	M	L	H	L	L
Signature	L	L	L	H	L	H	H
Voice	M	L	L	M	L	H	H

H: High M: Medium L: Low

Table (1.1) Biometrics Comparisons

4.1 Common Used Biometrics

Many types of biometrics are used in identification and verification the most used ones can be summarized as follows:

1.4.1. Fingerprint

Fingerprint is one of the most widely used biometrics. It is the oldest method used successfully in numerous applications. Each person has a unique fingerprint. This uniqueness can be determined by the pattern of ridges and furrows as well as the minutiae points. Minutiae points are local ridge characteristics that occur at either ridge ending [11]. Automatic fingerprint matching depends on the comparison of these local ridge characteristics and their relationship to make personal identification.

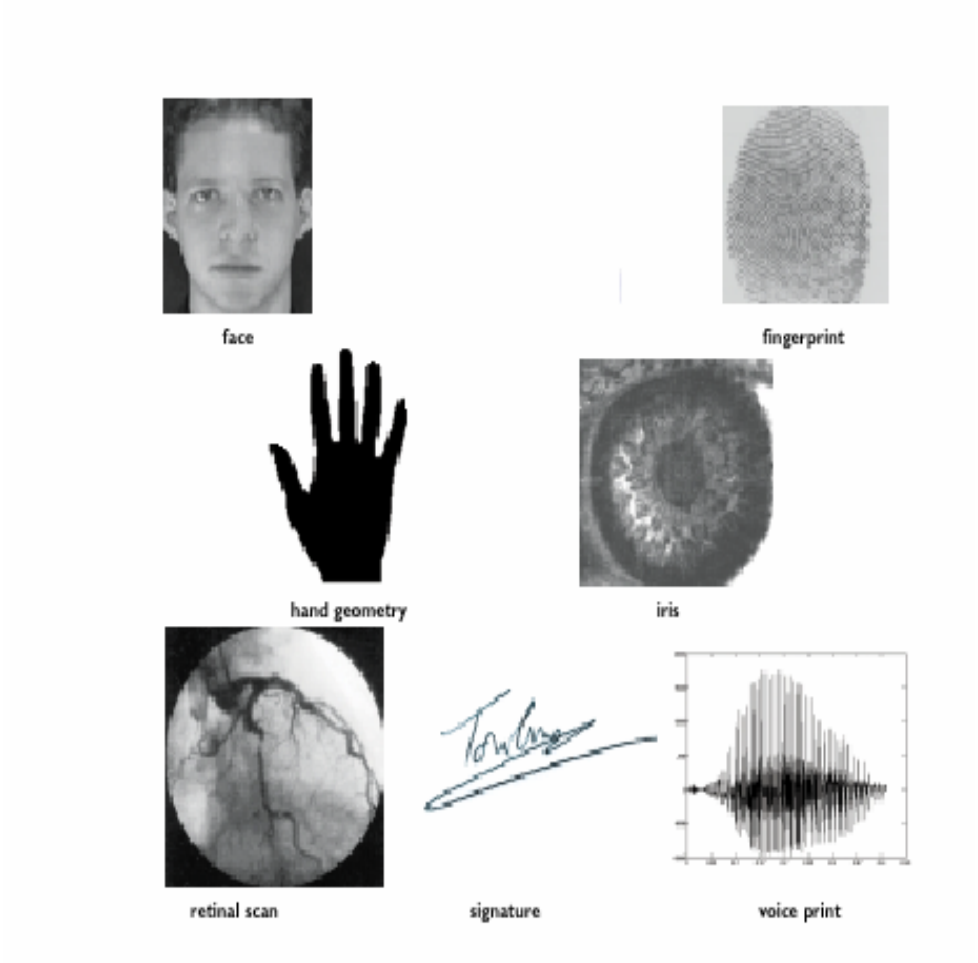


Fig (1.2) Some Different Biometrics

The difficult step in fingerprint matching is to extract minutiae from the fingerprint image automatically.

1.4.2. Hand Geometry

The second most widely used biometric is the hand itself [16]. Every hand Geometry is unique .An image is taken for the hand, it is placed on a highly reflected surface. The size and the location of the hand is determined. A second image is captured using the same camera to measure the thickness profile of the hand. For verification many calculations and comparisons are used.

1.4.3. Iris Recognition

It is the colored part of eye bounded by the pupil and sclera (white of the eye). It is rich of texture. Each iris is, and even irises of identical twins are different. It is easy to detect artificial irises (contact lenses) .Since iris is an internal protected organ, taking an image is a difficult process [15], so few recognition systems had been done and they were expensive.

1.4.4. Voice Recognition

One area where biometrics is tested in one of its sub-fields and proved success is voice recognition. We expect to pick up the phone and to be able to recognize someone by their voice after few words, also if we confuse the brain it becomes very good to narrow the possibilities [3].

Voice identification have many problems such as variation in microphone and transformation channel ,background noise, variation of voice due to illness, emotion of age…etc Voice recognitions systems are categorized depending on the freedom in what is spoken [17].

Voice identification plays an important roll after 11/9/2001. As an example the American C.I.A (central Intelligence Agency) was trying to trace Osama Bin Laden and some of his followers in Afghan-Pakistan

boarders. The area was bombarded and it was thought he was killed as he was already suffering kidney problems and no more signals were hared.

Later a tape was audio cassette (not video cassettes) by al-jazeera network commenting on current issues. If turns to be Bin Laden, then the man is still in business. U.S.A advisors had to match the voice, by a sound spectrograph.

Voice characteristics are distinctive (tone, accent, phrasing, voice) voice recognition methods approved the authenticity of the audiotape [37].

1.4.5. Signature Verification

Signature is a behavioral biometrics it is not based on physiological properties of the individual, such as fingerprint or face, but behavioral ones. Each person has a unique style of handwriting .So, no two signatures are identical, and variation of each one from a typical signature also depends upon the physical or emotional state of the person [17].

Signatures require contact and effort with the writing instrument. They seem to be acceptable in many government, legal, and commercial transactions as a method of verification [4]. Signatures are a behavioral biometric that change over a period of time and are influenced by physical and emotional conditions of the signatories. Signatures of some people vary a lot: even successive impressions of their signature are significantly different. More over, professional forgers can reproduce signatures like the original one of a person.

Usually signature verification is done by visual comparison. Someone compares the appearance of two signatures and accept the given one, if he sees that they are similar. Identification by signature has two important advantages [31]:

a- Identification by the signature is frequently acceptable, as the individuals are used to draw their signature while confirming their identity.
b- Signature can neither be stolen, nor lost or forgotten.

1.4.6. Other Biometrics

- **Face recognition**

Face recognition is a particularly compelling biometric because it is used every day by every one [24]. Because of its naturalness, face recognition is more acceptable than most biometrics, and the fact that cameras can acquire the biometric passively means that it can be very easy to use. There are two approaches to face recognition

1. Appearance-based approach: analyzing the appearance of the face, the universe of the face image domain is represented using a set of basis vector.
2. Geometric approach: analyzing the distance between facial attributes like nose, eyes, etc from the face image and the invariance of geometry properties is used for recognizing features.

Many systems combine the two approaches, but Appearance-based is mostly used, where the whole face is considered as a single entity, where many representations of separate areas of the face are created [17].

- **DNA**

DeoxyriboNucleic Acid (DNA) is the one-dimensional ultimate unique code for one's individuality, except for the fact that identical twins have identical DNA patterns [4].

- **Hand and finger geometry**

Some features related to a human hand (e.g., length of fingers) are relatively invariant and peculiar (although not very distinctive) to an individual. The image acquisition system requires cooperation of the subject and captures frontal and side view images of the palm flatly placed on a panel with outstretched fingers. The representational requirements of the hand are very small (nine bytes in one of the commercially available products), which is an attractive feature for bandwidth- and memory-limited systems.. Finger geometry systems (which measure the geometry of only one or two fingers) may be preferred because of their compact size [11].

- **Ear:**

It is known that the shape of the ear and the structure of the cartilaginous tissue of the pinna are distinctive. The features of an ear are not expected to be unique to an individual. The ear recognition approaches are based on matching the distance of salient points on the pinna from a landmark location on the ear [4].

- **Odor:**

It is known that each object exudes an odor that is characteristic of its chemical composition and could be used for distinguishing various objects. A whiff of air surrounding an object is blown over an array of chemical sensors, each sensitive to a certain group of (aromatic) compounds. A component of the odor emitted by a human (or any animal) body is distinctive to a particular individual [12].

- **Retinal scan**:

The retinal vasculature is rich in structure and is supposed to be a characteristic of each individual and each eye. It is claimed to be the most secure biometric since it is not easy to change or replicate the retinal vasculature. The image capture requires a person to peep into an eyepiece

and focus on a specific spot in the visual field so that a predetermined part of the retinal vasculature may be imaged. Retinal vasculature can reveal some medical conditions (e.g., hypertension), which is another factor standing in the way of public acceptance of retinal scan-based biometrics.

Chapter Two
Signatures: An overview

Signatures are one way of identifying personalities and in a certain areas it is the only way to identify persons. This chapter is devoted to the signatures as an identifying method and cast light on its types

2.1 Introduction

Signatures (from Latin signatura) are defined as mark of a person, written or impressed, with his own hand distinguishing himself or representing his approval. In modern terms a signature is a symbol, a password, or a secret formula of proof of identity.

What is the general definition of what a signature is? According to American Heritage Dictionary signature can be defined as: “the name of a person written with his or her own hand; i.e the act of signing one's name”. The second definition refers to the whole process of signing, and brings us to the assumption that the way the signature is made is a part of this signature [33]. Look at the first definition of the signature; it is possible to describe the signature as a static two dimensional image, which does not contain any time-related information. Both of the definitions of the signature lead to two different approaches of signature verification. The first is based on static characteristics of the signature, which are time invariant. In this sense signature verification becomes a typical pattern recognition task. Knowing that variations in signature patterns are inevitable the task of signature authentication can be narrowed to drawing the threshold of the range of genuine variations [33]. The second definition is based on dynamic characteristics of the process of signing, and is called on-line. The task in this case would be to extract some characteristics from the recorded information of the signing process and further compare them with the characteristics of the reference signature. The question that arises in this case is which kinds of

characteristics should be recorded and extracted in order to identify the person in question in the most efficient and accurate way.

2.2 Signature verification approaches:

Long before the invention of writing (i.e before 3000 B.C.) man thought of a method to identify himself. He invented small objects known as seals made of stone, clay or organic materials with different shapes (circular, square…etc) and with different and wide range of artistic designs(i.e. each seal with a different design). The purpose was to mark ownership of property and authenticity. These seals were to leave impressions when pressed or stamped. At a later date some seals started to carry inscribed letters to identify ownership or owner's name.

Now, Technology gives big and good chances to develop many systems for verification purpose.

Signature verification systems are used to evaluate whether or not a given signature is within acceptable threshold to be accepted as a sign of the person made the signature. It is widely used in automation in the fields of finance and security. The main task of any signature verification process is to detect whether the signature is genuine or forged. The instruments and the results of the verification depend upon the type of the forgery. There are three main types of forgery [33]:

- The first type of a forgery is a random forgery, can normally be represented by a signature sample that belongs to a different writer (meaning that the forger has whatsoever no information about the signature style and the name of the person),
- the second type of simple forgery is a signature with the same shape of the genuine writer's name

- The third type of the forgery is a skilled forgery, which is a suitable imitation of the genuine signature.

Each of the verification approaches (off-line and on-line) deals with different types of forgeries. Off-line methods are normally used with the random and simple forgeries. The reason for that is the fact that this method generally deals with the shape factors of the signatures [33].

The main step of signature verification systems in the two approaches is to convert handwritten word into a digital form to compare it with another word. These systems are also called optical character recognition (OCR) systems. The signature verification systems have some requirements such as:

- Data collection has to be accurate.
- Identification of the signature as correct signature.
- Determination of whether the given signature is accurate or forgery

Signature verification is one of the least accurate biometrics; the matching process is difficult and the user can easily change his signature to generate a false rejection. One of the advantages of signature verification is that the signature has been established as an acceptable form of personal identification method and can be used into the existing business processes requiring signatures. An advantage of online signature verification is that it is impossible for any one to obtain the dynamics information from a written signature for another one.

On-line system work on the dynamic process of generating the signature, a simple online signature verification system can be described by Fig (2.1). In the enrollment process the user has to provide a set of signatures. From these signatures different features can be extracted depending on the method used for verification. This feature extraction phase use other methods to get data vector stored as a template (data

base). For verification a user gives a test signature, same features extracted from it, compared with those in the data base, which is called verification phase.

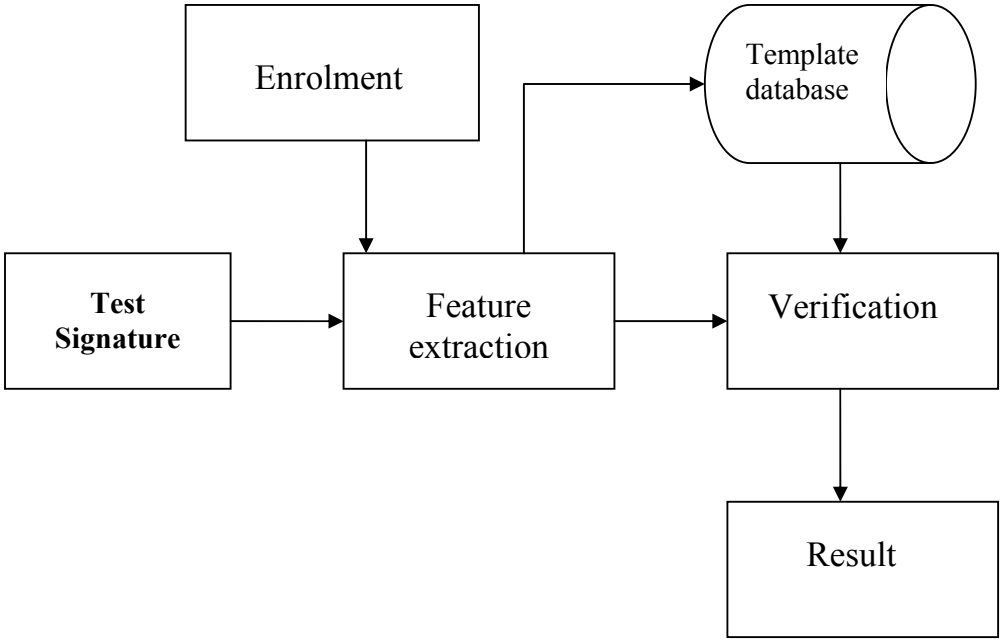


Fig (2.1) online signature verification

Off-line signature (Fig 2.2) deals with a static image of the signature. In the enrollment process an image of the signature captured using a scanner or any other device, a different features can be extracted depending on the method used for verification. For verification phase there are many approaches used, for example *Neural Network* (explained deeply in chapter 3), *Euclidean Distance classifier* and *Dynamic time warping* (DTW).

Each of those systems must give an answer to the question whether this signature is accepted or rejected?

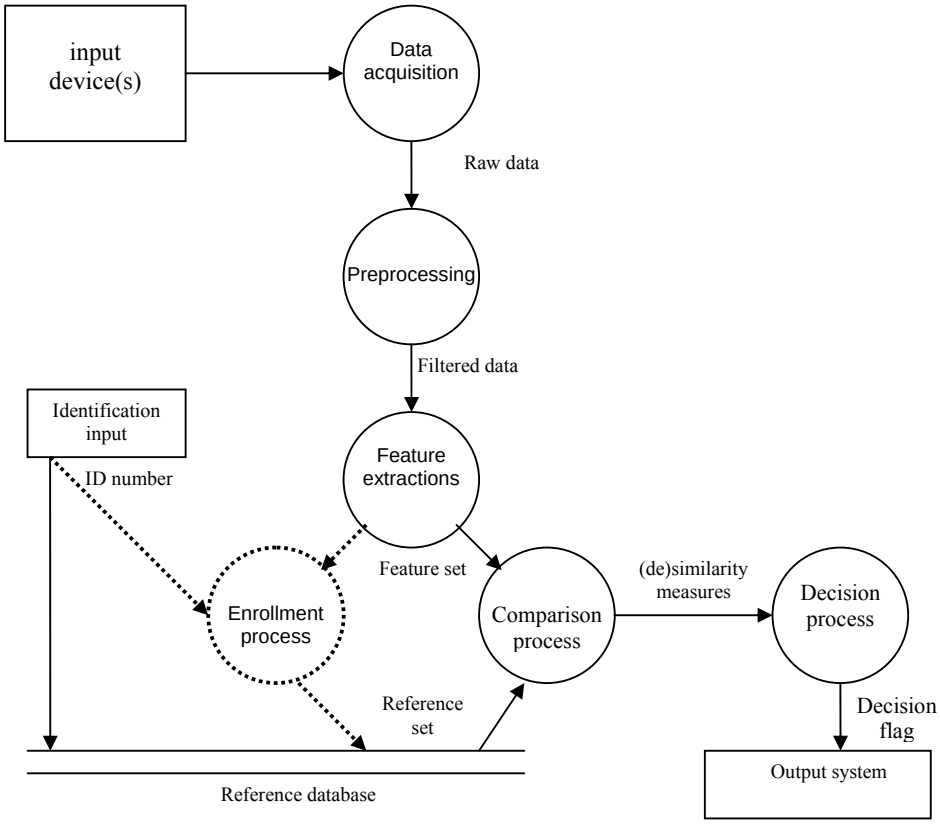


Fig (2.2) Data Flow Diagram Offline Signature Verification System

2.2.1 Online signature verification:

In Online signature verification, the choice of signals that can be processed is fairly large (the x and y coordinates of a pen tip as a function of time, speed, acceleration, pressure, etc.) and therefore, the signal acquisition process is important to the whole verification process. Special equipment is used to support the acquisition step, for example, an instrumented pen or a digitized graphics tablet. There are numerous approaches to dynamic signature verification based on comparison techniques (regional correlation, elastic matching, and tree matching), segmentation or neural networks. There are several systems for online handwriting acquisition available on the market such as:

- I-pen (inductum): it is a pen which works with any paper, the main functions of it are text and drawings capture, it has an interchangeable nib, the data acquisition start when the pen touches the paper.
- E-pen (in motion): it captures the user's handwriting and converts it into digital forms. It is a two part system, a receiver module and the pen itself, no need for special papers, the receiver can be connected to a computer, it convert the hand writing into a digital form.
- Camera –based data Acquisition [31]: in their project discuss the use of Camera –based data Acquisition, they consider that all mentioned systems are bulky and complicated to use, but cameras are smaller and simpler to handle. The user writes on a paper with a regular pen, it uses an optical signal detector to get the position of the nib. The position has given the best match between the signals and detector.
- Tablet-Based Data acquisition device: also called graphic tablets or pads. It is the most used device in this area; it used a physical device as an interactive between the tablet and the pen, known as digitizer. There are some points which determine the quality and possible application area of the tablet these are: size of the active area of the pad, resolution, pressure sensitive levels, and sample rate (table 2.1). Input device determine the quality of signature features, being extracted during the signing sessions, and directly affect the performance of the systems. Digitizer translates the position of the pen into x and y coordinates values, it consists of [17]:
-

- A pen or human finger to generate input data,
- A sensor device to generate x,y coordinates from input data,
- A micro controller to convert the x,y coordinates into, digital data.
- driver software

Model	Active are	Pressure level	Resolution
Genius WizardPen	4"×3"	512	4064lpi
Genius MousePen	4"×3"	512	4064lpi
Interlink ePad-ink	3"×2.20"	512	300dpi
UC-Logic SuperPen 4030	4"×3"	512	1000lpi
UC-Logic SuperPen 8060	8"×6"	1024	1000lpi
Acedad Flair	5"×3.75"	512	2540lpi

Table (2.1) some tablets available in the market

2.2.2 Off-line signature verification:

In this method the signature is written on document which can be scanned and obtain its digital image representation. It has to be mentioned that the signature needed to verify may be genuine or forgery. The only input here is the image of the signature captured by a traditional scanner in most cases, and there are other types of scanners such as:

C-pen which consists of a digital camera, a processor and a memory. The digital camera inside the pen captures the text and saves it in the memory then it can be transferred to a PC using a cable.

Many problems arise here such as variation within genuine signature, and noise introduced by the scanning device, different in pen width, which can make off-line signature verification a defiance problem. Signature verification research, works on random forgery detection and skilled forgery detection. Skilled forgery detection is a much more

difficult task; since differentiation between a forgery and a genuine signature is difficult. Random detection uses only the image based features to classify signatures as genuine or forgeries verification.

2.2.2.1 Feature extraction

Off-line signature verification using scanned images extract many types of features such as:

- *Global features*: The set are the features extracted from every pixel that lies within a rectangle circumscribing the signature. These features do not reflect any local, geometrical, or topological properties of the signature, but include transformations, series expansions, image gradient analysis etc. Although global features are easily extractable and insensitive to noise, they are dependent upon the position alignment and highly sensitive to distortion and style variations. Global features are considered to be classical in the pattern recognition problems. The most used features are:
  - Image area
  - Pure width (the width of the image with horizontal black space removed)
  - Pure height
  - Baseline shift
  - Gravity of the left and the right part of the image.
  - Vertical center of mass
  - Horizontal center of mass
  - Maximum vertical projection
  - Maximum horizontal projection

- Vertical projection peaks. The number of local maximums of the vertical
- Projection function.
- Horizontal projection peaks
- Global slant angle
- Local slant angle
- Number of edge points
- Number of cross points
- Number of closed loops

Some writers used most of these features in on-line systems such as total signature time, time of pen down, average horizontal speed, x,y speed correlations, first moments.

- *Statistical features*: these features are derived from the distribution of pixels of a signature, e.g. statistics of high gray-level pixels to identify pseudo-dynamic characteristics of signatures. This technique includes the extraction of high pressure factors with respect to specific regions (for example, upper, middle and lower envelope) ,also maximum distance between the highest and lowest points, signature length, standard deviation of x/change in x,y/change in y
- *Geometrical and topological features*: these describe the characteristic geometry and topology of a signature and thereby preserve the signatures global and local properties, e.g. local correspondence of stroke segments to trace signature.

2.3 Previous Work

Many methods have been applied to off-line and on-line signature verification. To develop an offline signature system, [22] work on some geometrical features based on the shape and dimension of the signature image. Some features were used. These are:

- Baseline Slant angle which is the imaginary line about which the signature is assumed to rest.
- Aspect Ratio (A): It is the ratio of width to height of the signature.
- Normalized area of the signature (NA): It is the ratio of the area occupied by signature pixels to the area of the bounding box
- Centre of Gravity : Which is the 2-tuple(X,Y)given by specific equation
- Slope of the line joining the centers of Gravity of two halves of signature image.

Those features were extracted from a group of signature images .The mean values and standard deviation for all features are computed and used for verification, the Euclidian Distance given by the following equation is calculated: $\delta = (1/n)\sum_{i=1}^{n}[(Fi - \mu i)/\sigma i]2$

If this distance is less than a defined value then the signature is accepted as a genuine otherwise it is assumed to be forged. When the system was tested it give a result of 8.5% False Reject Rate for original signature, for the forged it gives 13.3% False Acceptance Rate. This system gives bad results if large database is used, and it fails in case of skilled forgeries.

Kholmatov [2] in his offline system he extracts four different feature upper and lower envelopes, vertical and horizontal projections. To extract

upper envelope, each column of the image is traversed from top to bottom. The location of the first non-white pixel is marked as a point of the upper envelope, also to extract the lower envelope each column of the image is traversed from bottom to top. After feature extraction the feature vectors are merged into one vector, and then it is compared using the Euclidian distances which lead to a weak result. As a second way he used autocorrelation which misses the ability to deal with non linear distortions in the signatures, then he used Dynamic Time Warping algorithm (in order to compare two signatures) which is defined by the following equations.

$$c[i, j] = Min\begin{cases} c[i-1] + GapCost \\ c[i, j-1] + GapCost \\ c[i-1, j-1] + Dist(S_1[i], S_2[j]) \end{cases}$$

where $S_1[i]$ denotes i'th signature's j;s point in trajectory and

$$dist(x, y) = \begin{cases} 0 & \text{if} \left\| x - y \right\| < \text{thr} \\ \left\| x - y \right\| - \text{Thr} & \text{otherwise} \end{cases}$$

Where c is matrix filled by the algorithm, $GapCost$ is the constant coefficient.

This algorithm finds the best non-linear alignment of two vectors such that the overall distance between two corresponding vector elements is minimized in least square sense. The overall distance between two signatures S1 and S2 is calculated in linear time using specific equation, Which is mainly based on distances on the signature, the test signature is compared with each reference signature, resulting in a number of distance to the closest, farthest, and the template signatures are all used to classify the test signature as genuine or forgery. Same method was used for offline and online signature verification. It gets low performance in offline signature verification, because it is hard to differentiate between forgery and genuine signature of a writer.

Meenakshi [20] worked on shape matrices and local size distributions. Shape matrices were used as a mixed shape matrices features for signature verification. Mixed shape feature is a global feature, in calculation of which local features are taken into account. In another research she used extended shadow codes as a shape feature to detect random forgeries. These codes incorporate both global and local representation of a signature. To calculate shadow codes the image is projected onto a bar mask array, where each bar is associated with spatially constrained area of a signature. Feature vector of a normalized shadow codes is then used by a k-Nearest neighborhood and a minimum distance classifiers. After testing, the system gave a performance result of 2.16% EER. In another research she used a combination of three types of features (Global features, statistical features and geometrical features) .This combination known as Gradient, Structural and Concavity or GSC features. The gradient features detect local features of the image and provide a great deal of information about stroke shape on small scale. The structural features extend the gradient features to longer distance to give useful information about stroke trajectories. The concavity features are used to detect stroke relation ships at long distance which can span across the image. The system gave result of FAR 34.91% while FRR is 28.33%.

Guo [35] approached the problem by establishing a local correspondence between a model and a questioned signature. The questioned signature was segmented into consecutive stroke segments that were matched to the stroke segment of the model. Stroke segment boundaries were defined by topological features, such as crossings and endings. The cost of the match was determined by comparing a set of geometric properties of the corresponding sub-strokes and computing a weight sum of the property value differences. The least invariant features

of the least invariant sub strokes were given the biggest weight. He reported a 0.13% false accepted rate and 0.39% false reject rate

Murshed [21] and others centralized the image of the signature to divide it into m regions, through the use of an identity grid. Each region was divided in 16-pixel. The geometrical structure was defined with respect to the center of the image area such that, when a signature is centralized on the image, it becomes centralized on the grid. Graphical segments are extracted from each region in binary signature and applied to a back-propagation network which reduced the size of the segment and then applied to the comparison stage, which is composed of m Fuzzy ARTMAP networks, each of which responsible for one region in the signature. This system gives a good result with small data base but it was not tested with a large one, the researchers believe that this approach may provide an efficient solution to very difficult problem in the field of signature verification.

Kalenova [32] in this work uses three different sets of feature, global features, grid information features and texture features. Global features tacked many (signature height, image area, pure height, vertical centre of mass, horizontal center of mass). In grid information features he divided the image to 96 rectangles and the area of each is calculated. In texture feature they used a 2x2 matrices describing the transformation of black and white pixels for given direction and distance. In classification phase a three neural network is used one for each set of features

Woodwards [16] in this research take online signature verification, several local features were extracted, absolute and relative speed, gray value in a neighborhood, differences between two consecutive points, absolute relative speed etc. The only extracted global feature was the number of signature strokes Dynamic programming algorithm was used for the comparison .The dissimilarity value between the test and template

signatures was calculated .Three different criteria were investigated in verification, the minimum, the maximum, and the average dissimilarity values to the test signature. When tested Jain et al reported best results, the performance of the system was a 2.8% false accept rate and 1.6% false reject rate.

Leung [34] used a vertical and horizontal projection of signature, to compare two signatures their corresponding projections were aligned using dynamic programming algorithm. Wrapping function of the test signature was compared to the average warping function of referenced set signatures using the mahalanobis distance. He reported a result of 23.2% false reject rate and 21.4% false accept rate.

Connell [1] tries to combine some methods to get a new method for signature verification, so he takes the methods: holistic approach where the test signature is matched with each one of N reference signature. Another approach is regional matching approach in which the test signature is split into segments as well as reference signature then matching the corresponding segments.

Papamarkos [31] develops a system for signatures captured by camera-based acquisition system. The system uses computer vision techniques and estimation theory to track the position of the pen tip in image plane, and then use an algorithm to compare the 2D shape of the signatures. He uses Dynamic Time Warping (DTW) method in the comparison algorithm.

Jambi [5] used some features that can be easily extracted or computed: Total Time, Pen-up Time, total path length, number of sign changes in the x and y velocities and x and y accelerations and the number of zero values in the x and y accelerations. After extracting these features the values placed in a vector T. For comparison they used the mean and standard deviation of the vectors, and then compute the

distance vector, normalized. The norm vector compared with threshold. This system gave a result of 2.5%FRR and 8.6%FAR. He defines minimum rectangle surrounding the signature and divides it into 10 small rectangles from which features are looked out. The features are end points, branch points, loops, curves and crossing with boundaries of these ten rectangles. These features have to be detected from a contour where the corners on it are to be defined. This approach is ignored due to programming complexities, and then he tried another approach which is based on the study of distribution of black pixels of the signature image using Chi Square Test. Also the results he quoted were not promising so it can be ignored.

After those two trials [5] Dr. Kamal M.Jambi with a research team in King Abdul Aziz University revised the work of Reena Bajaj and Santanu in which they deals with signatures through three different features:

- Horizontal and vertical projection moments which is used to produce skewness measures.
- The second and third features are based on lower and upper envelopes which are processed to identify some structural features and their distribution with in the smallest rectangle surrounding this envelops,
- A neural network is used for recognition.

Also here the project was not completed but, they got some result, and they have some recommendations further work:

- Implementation and testing of the program for the project
- Obtaining the image of the signature then it should be shifted, rotated to get the same shape of the tested signature.

From the research in signature verification the most used feature are of three types: Global features (peaks and valley of the contour),

Statistical features (upper and lower contour using different methods), Geometrical and topological features (peaks and valley of the contour)., vertical and horizontal projection, divides the mage in rejoin then treat the rejoin in different methods, calculate the height of image .Most of the researchers combine different types of features to get good results.

In verification phase the most used method is Neural Net work, Dynamic Warping time and Euclidian Distance.

In this research vertical and horizontal projection was used to give the moments, upper and lower contour is taken, then a neural network is used.

Chapter Three
Image Processing

At this stage of the research an overview of image processing is to be given before we proceed into details of describing the system.

3.1 Introduction

Image Processing and Analysis can be defined as the "act of examining images for the purpose of identifying objects and judging their significance"**[**29**].**

As mentioned before the purpose of this research is offline signature verification. Signatures collected from some people then using scanner for data acquisition, so they are treated as images.

What is an image?

An image is a 2-dimentional light intensity function f(x,y),where x and y are spatial coordinates and the value of f at (x,y) is proportional to the brightness of the image at that point. A digital image is an image f(x,y) that has been discredited both in spatial coordinates and brightness. It represented by a 2-dimentional integer arrays. Each element of the array is called a pixel or a pel [19].

There are five basic types of images supported by the software used to develop our system. These image types are primarily for the purpose of display, they do not constrain the values of an image that can be processed using general image processing techniques:

1. Indexed images
2. Intensity images
3. Binary images
4. RGB images
5. 8-bit images

- Indexed image: It consists of a data matrix, X, and a color-map matrix, map. The data matrix can be of class uint8 (a unit here is a pixel), uint16, or double. The color-map matrix is an m-by-3 array of class double containing floating-point values in the range [0, 1].
- Intensity image: It is a data matrix, I, whose values represent intensities within some range. MATLAB stores an intensity image as a single matrix, with each element of the matrix corresponding to one image pixel. The matrix can be of class double, uint8, or uint16.While intensity images are rarely saved with a colormap. Elements in the intensity matrix represent various intensities, or gray levels, where the intensity 0 usually represents black and the intensity 1, 255, or 65535 usually rep-resents full intensity, or white.
- Binary image: Here each pixel assumes one of only two discrete values. Essentially, these two values correspond to on and off. A binary image is stored as a logical array of 0's (off pixels) and 1's (on pixels)
- RGB image: sometimes referred to as a true color image, it is stored in MATLAB as an m-by-n-by-3 data array that defines Red, Green, and Blue color components for each individual pixel. RGB images do not use a palette. The color of each pixel is determined by the combination of the red, green, and blue intensities stored in each color plane at the pixel's location.

3.2 Image processing operations

Image Classification is an important part of the fields of “Image Analysis and Pattern Recognition”. Digital Image Classification is the

process of sorting all the pixels in an image into a finite number of individual classes. *Pattern Recognition, Textural Analysis and Change Detection* is different forms of classification that are focused on 3 main objectives [29]:

1. Detection of different kinds of features in an image.
2. Discrimination of distinctive shapes and spatial patterns
3. Identification of temporal changes in image

An image is digitized to convert it to a form which can be stored in a computer memory or any secondary storage memory such as hard disk or CD-ROM. Digitization process can be done by scanner or other digitization device. Once the image has been digitized, it can be processed by any image processing software. Image processing operations can be divided into two stages: preprocessing and processing.

Preprocessing operations include: Threshold to reduce a gray-scale or conversion of a colour image to a binary image, reduction of noise by reducing extraneous data, segmentation to separate various components in the image, and, finally, thinning or boundary detection to enable easier subsequent detection of pertinent features and objects of interest. Generally preprocessing has three steps:

- Image Compression: most people use this type since it reduces the amount of memory needed to store the image.
- Image Enhancement and Restoration: is the process by which defined image is improved so that it looks better. It is used to correct errors of image that caused during scanning of the picture or bad lighting.
- Measurement Extraction: it is used to obtain needed information from the image.

3.2.1 Image Enhancement and Restoration

The enhancement techniques depend upon two factors mainly:

- The digital data (i.e. with spectral bands and resolution)
- The objectives of interpretation

As an image enhancement technique often drastically alters the original numeric data, it is normally used only for visual (manual) interpretation and not for further numeric analysis. Common enhancements include image reduction, image rectification, image contrast adjustments, Fourier trans-formations, principal component analysis and texture transformation [29].

3.2.3 Measurement Extraction

There are many types of operations that can be applied to digital images to transform an input image a [m,n] into an output image b[m,n] i.e. *obtain needed information from the image* (or another representation) such as**[14].:**

- *Point operation:* the output value at a specific coordinate is dependent only on the input value at that same coordinate.
- *Local operation:* the output value at a specific coordinate is dependent on the input values in the *neighborhood* of that same coordinate.
- *Global* operation: the output value at a specific coordinate is dependent on all the values in the input image.

3.3. Image analysis & Statistics

In image processing simple statistical calculations can be used for descriptions of images. The notion of a statistic is intimately connected to the concept of a probability distribution, generally the distribution of signal amplitudes. For a given region which could conceivably be an entire image we can define the probability *distribution* function of the bright nesses in that region and the probability *density* function of the brightness in that region.

In the analysis of the images it is essential that to distinguish between the portions of the image we interest in and "the rest of the image" which is called the background. The techniques that are used to find the portions of interest are usually referred to as *segmentation techniques* segmenting the foreground from background. Here is the two of the most common techniques *threshold* and *edge findin*g (contour) and we will present techniques for improving the quality of the segmentation result [14]. It is important to understand that:

- there is no universally applicable segmentation technique that will work for all images, and,
- no segmentation technique is perfect.

❖ **Threshold**

This technique is based upon a simple concept. A parameter θ called the *brightness threshold* is chosen and applied to the image a[m,n] as follows:

If a[m, n] >=θ a[m, n] = *image* = 1

Else a[m, n] = *background* = 0

This version of algorithm assumes that we are interested in light objects on a dark background. For dark objects on a light background we would use:

If a[m, n] < θ a[m, n] = *image* = 1

Else a[m, n] = *background* = 0

❖ **Edge finding**

Threshold produces a segmentation that yields all the pixels that belong to the object or objects of interest in an image. An alternative to this is to find those pixels that belong to the borders of the objects. Techniques that are directed to this goal are termed *edge finding techniques* which it can define by finding neighborhood.

3.3.1 Neighborhood operations:

When objects are described by their skeletons or contours, they can be represented more efficiently than simply by ON and OFF valued pixels in a raster image. One common way to do this is by chain coding, where the ON pixels are represented as sequences of connected neighborhoods along lines and curves. Instead of storing the absolute location of each ON pixel, the direction from its previously coded neighborhood is stored. A neighborhood is any of the adjacent pixels in neighbor around that centre pixel [28].

Neighborhood operations is most important operations in modern digital image processing, therefore it is important to understand how images can be sampled and how that relates to the various neighborhoods that can be used to process an image [14]:

- Rectangular sampling: In most cases, images are sampled by laying a rectangular grid over an image as illustrated in Fig (2.1).

- Hexagonal sampling: An alternative sampling scheme is shown in the figure and is termed hexagonal sampling.

Local operations produce an output pixel value b[m=*m o* ,n=*n o*] based upon the pixel values in the *neighborhood* of a[m=*m o* ,n=*n o*]. Some of the most common neighborhoods are the 4-connected neighborhood and the 8-connected neighborhood in the case of rectangular sampling and the 6-connected neighborhood in the case of hexagonal sampling illustrated in Figure (3.1)

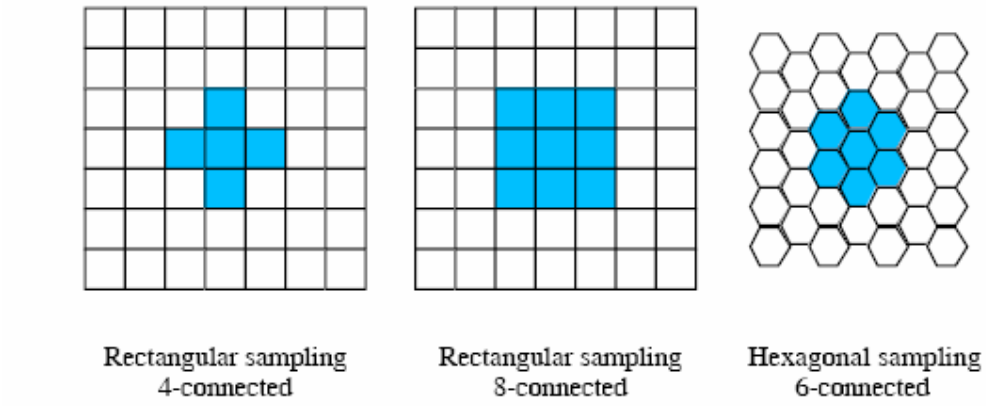


Fig (3.1) Neighborhood operations

3.3.2 Contour Representations

When dealing with an image or a region of it, several compact representations is available that can facilitate manipulation of and measurements on this image. In each case we assume that we begin with an image representation as shown in Fig (2.2). Several techniques exist to represent the image or a region by describing its contour .The most common used one is Chain code.

3.3.3. Chain code

To extract some feature from an image using it contour, it has to be traced to get pixels of the line surround this image. We follow the contour in a clockwise manner and keep track of the directions as we go from one contour pixel to the next. For the standard implementation of the chain code we consider a contour pixel to be an object pixel that has a background (non-object) pixel as one or more of its 4-connected neighbors.

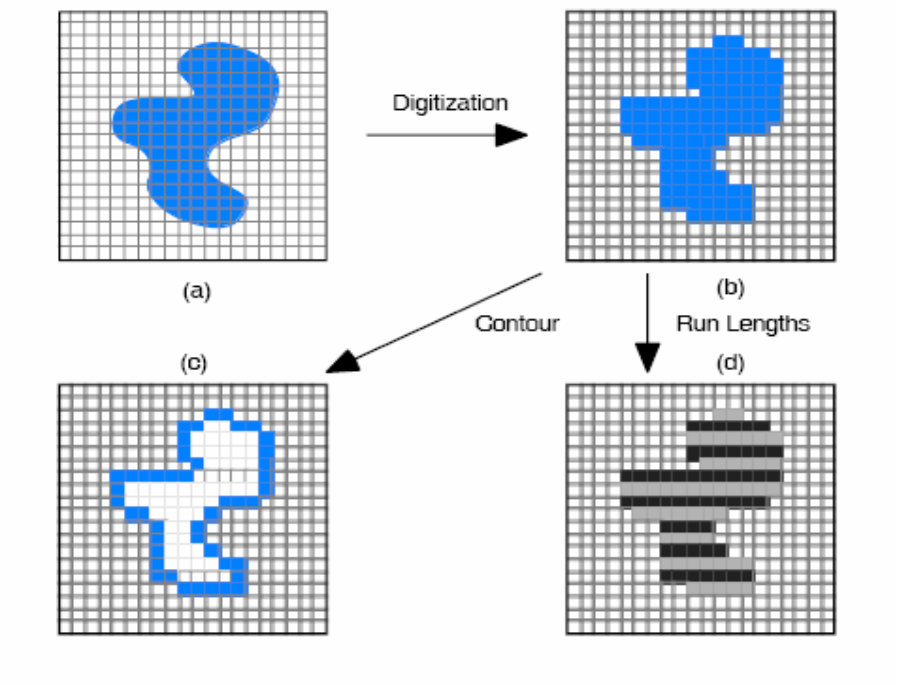


Fig (3.2) region (shaded) as it is transformed from (a) continues to (b) discrete from and then considered as a (c) contour or (d) run length

Chain code properties

- Even codes {0, 2, 4, 6} correspond to horizontal and vertical directions; odd codes {1, 3, 5, 7} correspond to the vertical directions.
- Each code can be considered as the angular direction that we must move to go from one contour pixel to the next.

- The absolute coordinates [m, n] of the first contour pixel (e.g. top, leftmost) together with the chain code of the contour represent a complete description of the discrete region contour.

3.4. Overview of handwriting recognition system

This research try to compare two signatures, the signature treated as an image, there are some process used with before comparison. Let us assume that a binary image of the signature be the input data. Before we start the segmentation process to do a *pre-processing* of the image which includes the following steps Fig (2.3):

- *filtering*: it is to reduce noise and make easier extracting the structural features. The most important reason to reduce noise is that extraneous features otherwise cause subsequent errors in recognition. The objective of a filter to reduce noise is that it remove as much of the noise as possible while retaining the entire signal.
- *thinning (get contour):* Thinning is an image processing operation in which binary valued image regions are reduced to lines that approximate the centre lines, or skeletons, of the regions. The purpose of thinning is to reduce the image components to their essential information so that further analysis and recognition are facilitated. i.e. removes outer pixels by iterative boundary erosion process until a skeleton of pixel chains only remains,
- *Searching vertices*: at this step we extract line junctions and the ends of the lines called vertices.

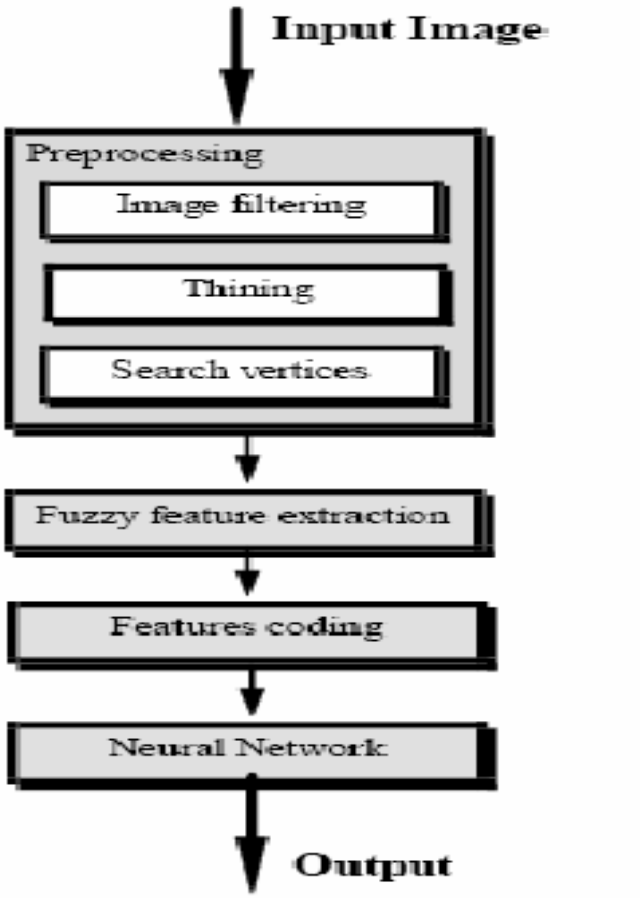


Fig (3.3) Handwriting recognition system

After preprocessing the image is ready for processing, this is done in the following steps:

- *Feature extraction.* The task of our feature extraction module is to process a binary image so that to obtain a vector of features.
- *Features coding block* includes a complex structure for input layer of a ANN classifier. The input vector contains details for our character extracted from previous step.
- The last step is to classify the character using *neural network*

3.5 Neural net work and image processing:

The input image is split up into blocks or vectors of 8×8, 4×4 or 16×16 pixels. When the input vector is referred to as *N*-dimensional which is equal to the number of pixels included in each block, all the coupling weights connected to each neuron at the hidden layer can be represented by $\{w_{ji}, j = 1, 2, \ldots K \text{ and } i = 1, 2, \ldots N\}$, which can also be described by a matrix of $K \times N$. From the hidden layer to the output layer, the connections can be represented by $\{w'_{ij} : {}^{1 \le i \le N, 1 \le j \le K}\}$ which is another weight matrix of $N \times K$. Image compression is achieved by training the network in such a way that the coupling weights, $\{w_{ji}\}$, scale the input vector of *N*-dimension into a narrow channel of *K*-dimension ($K < N$) at the hidden layer and produce the optimum output value which makes the quadratic error between input and output minimum. In accordance with the neural network structure, the operation can be described by the following equations:

$$h_j = \sum_{i=1}^{N} w_{ji} x_i \qquad 1 \le j \le K \qquad (1)$$

for encoding and

$$\overline{x_i} = \sum_{j=1}^{K} w'_{ij} h_j \qquad 1 \le i \le N \qquad (2)$$

for decoding.

where $x_i \in [0, 1]$ denotes the normalized pixel values for grey scale images with grey levels [0, 255]. The reason of using normalized pixel values is due to the fact that neural networks can operate more efficiently when both their inputs and outputs are limited to a range of [0, 1]. Good discussion on a number of normalization functions and their effect on neural network performances can be found in reference [15].

3.6. What is MATLAB?

MATLAB is selected to develop the program for the used method of verification and neural network training. MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include:

- Math and computation
- Algorithm development
- Data acquisition
- Modeling, simulation, and prototyping
- Data analysis, exploration, and visualization
- Scientific and engineering graphics
- Application development, including graphical user interface building

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations [30].

Chapter Four

Neural Network

4.1Biological Neuron

The brain is a collection of neurons. A neuron is the basic element of the brain. There are approximately 10 billion interconnected neurons. Each neuron has about 10 thousand synapses (connections).Neuron is a cell that uses biochemical reactions to receive, process and transmit information [6]. A neuron contains (see Fig 4.1):

- A cell body for signal processing
- Many dendrites to receive the signals
- An axon for outputting the result
- Soma for maintenance to keep neuron functional
- Synapse an interface between an axon and dendrite

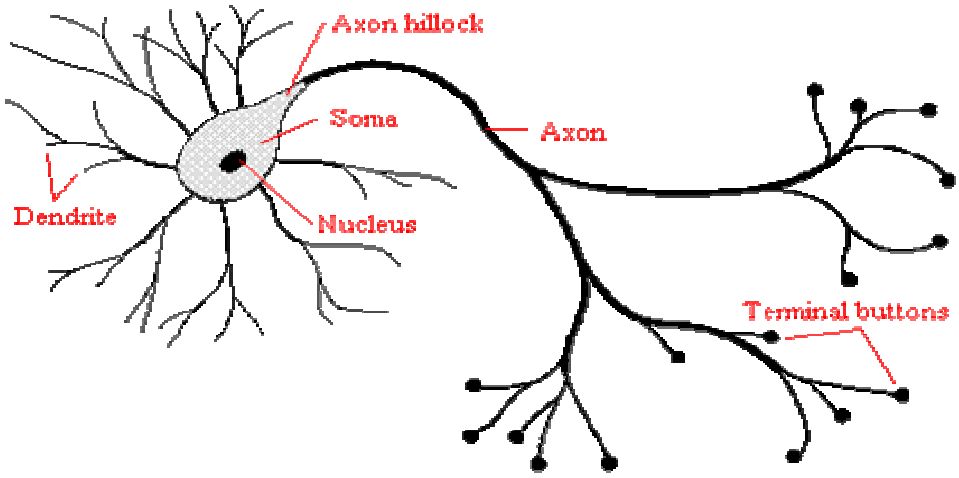


Fig 4.1 Biological neuron

The neurons of the brain are made of cell body from which springs dendrites that play the role of receptors. When an incoming axon joins another cell it results in what is known as synapse. The neuron transmits electrical signals through its axon to the dendrites of other neurons.

4.1.1 How biological neurons work?

- Signals (impulses) come into the dendrites through the synapses

- All signals from all dendrites are summed up in the cell body
- When the sum is larger than the threshold, the neuron fires, and send out impulse signal to other neurons through the axon

4.2 Artificial Neural Networks

Artificial Neural Networks (ANNs) are computational systems whose architecture and operation are inspired from our knowledge about biological neural cells (neurons) in the brain. In other words we can say it is study of information process in some way like neurons of the brain. Also it can be defined as: information processing system that has performance characteristics in common with biological neural networks.

ANNs are able to solve problems without any priori assumptions, as long as enough data is available, a neural network will extract any regularities and form a solution [7]. They may provide valid answers and solve problems beyond the scope of present computers. They are already in use in many fields. They have been generated based on [fundamentals]:

1. Information processing occurs at many simple elements called neurons.
2. Signals are passed between neurons over connection link.
3. Each connection link has an associated weight.
4. Each neuron applies an activation function to its net input to determine its output signal.

ANNs are able to solve difficult problems in away that resemble human intelligence. What is unique about neural networks is their ability to learn by example [7].Neural networks can be applied to many fields such as speech recognition, pattern recognition, handwritten character recognition…etc.

Neural Networks enjoy some features. Some of these features are [15]:

1. They can be trained to classify poorly structured inputs;
2. They are robust against noise in training data;

3. They are robust against loss of neurons;
4. They can be used in hybrid neural net/AI systems e.g, language learning (leading to compiler);

4.3 Architecture of Neural Networks:

Artificial Neural Networks (ANN) consist of

- **Node**: simple elements called cells or node, each cell receives information from another cell, process it then send the result to another cell
- **Layer**: a collection of cells called layer. Cells of the same Layer are either fully connected or not interconnected [17].
- **Link**: a connection between two cells from different layer is called a link.
- **Weight:** each link has an associated weight which represents, information used by the net to solve the problem. The main structure of ANN contains an input layer, one or more hidden layer and an output layer.

As ANNs are models of biological neural structures, the starting point for any kind of neural network analysis is a model neuron whose behavior follows closely our understanding of how neuron works. This model is shown in Fig 4.2.

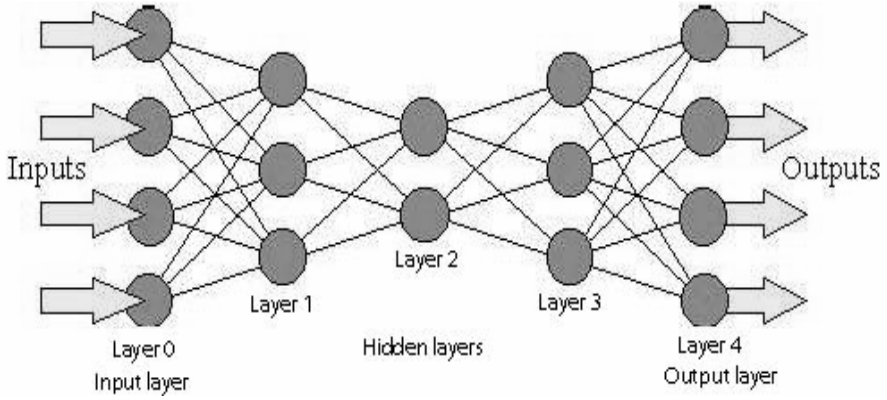


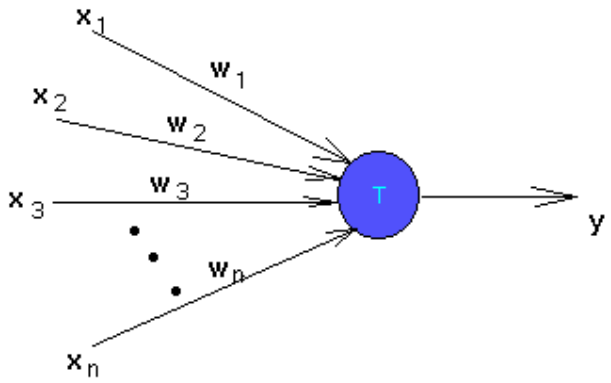
Fig (4.2) Multilayer neural network

4.4 Types of Artificial Neural Networks

Neural networks can be classified according to its number of layers into two types; the input layer is not to be counted since it is not a computation layer

Single layer Net: It has one layer, an input unit which receives signals and sends the results to the output unit. It has no hidden layer. This is a very simple model and consists of a single `trainable' neuron. Trainable means that its threshold and input weights are modifiable. Inputs are presented to the neuron and each input has a desired output (determined by us). If the neuron doesn't give the desired output, then it has made a mistake. To rectify this, its threshold and/or input weights must be changed. How this change is to be calculated is determined by the learning algorithm.

It looks like -



- x1, x2, ..., xn are inputs. These could be real numbers or boolean values depending on the problem.
- y is the output and is boolean.
- w1, w2, ..., wn are weights of the edges and are real valued.
- T is the threshold and is real valued.

The output y is 1 if the net input which is

$$w1\ x1 + w2\ x2 + ... + wn\ xn$$

is greater than the threshold T. Otherwise the output is zero.

Multi layer Net: It has one or more layers of nodes; these layers lay between the input and output units Fig (4.2).Usually it is used to solve the problems that can not be solved by the single layer net.

4.5 Pattern Recognition & Neural Networks

The inputs that we have been referring to, of the form (x1, x2, ..., xn) are also called as patterns. If a **ANN** gives the correct, desired output for some pattern, then we say that the per **ANN** recognizes that pattern. We also say that the **ANN** correctly classifies that pattern.

Since a pattern by our definition is just a sequence of numbers, it could represent anything -- a picture, a song, a poem... anything that you can have in a computer file. We could then have a **ANN** which could learn such inputs and classify them e.g. a neat picture or a scribbling, a good or a bad song, etc. All we have to do is to present the **ANN** with some examples -- give it some songs and tell it whether each one is good or bad. (It could then go all over the internet, searching for songs which you may like.) Sounds incredible? At least that is the way it is supposed to work. But it may not. The problem is that the set of patterns which we want the **ANN** to learn might be something like the XOR problem. Then no **ANN** can be made to recognize the taste. However, there may be some other kind of neural network which can do this.

4.6 Training

The method of setting the values of the weights is an important characteristic of different neural nets. There are two types of training [17].

1. Supervised
2. Unsupervised

4.6.1 Supervised training

When training is accomplished by presetting a sequence of training vector, each with an associated target output vector, then the weight adjusted according to learning algorithm, this is called supervised training.[fundamentals] .

4.6.2 Unsupervised training

Here the net is self-organizing, which means that the net performs some modification so that the similar input vectors assigned to the same output unit.

4.7 Some Activation Functions

There are many functions used with ANN, here we mention some common functions:

- Identity function it is defined by the equation

  F(x) = x for all x

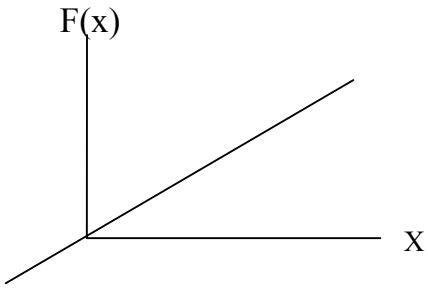


Fig (4.3) Identity Function

- Binary step Function

It is continues value variable with binary value (1 0r 0) it is represented by (Fig 4.4).

$$f(x)=\begin{cases} 1 & \text{if } x>=0 \\ 0 & \text{if } x < \theta \end{cases}$$

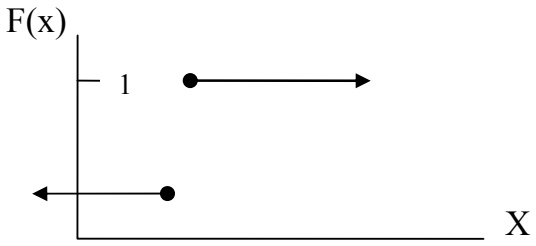


Fig (4.4) Binary Step Function

- Summation Function:

It gives the sum (y) of the products of each input (x_i) by the weight (w_i)

$$y = \sum_{i=1}^{n} x_i w_i$$

- Sigmoid Functions

There are two type of sigmoid function, binary sigmoid and bipolar sigmoid

binary sigmoid :it gives an output values either are binary or in the interval {0,1},it is given from the relations.

$$f(x)=\frac{1}{1+\exp(-\sigma x)}$$

$f'(x) = \sigma f(x)[1 - f(x)]$

bipolar sigmoid: it gives output values in the range of [-1,1] ,it is given from the relations

$$f(x) = \frac{2}{1+\exp(-\sigma x)} - 1$$

$$= \frac{1-\exp(-\sigma x)}{1+\exp(-\sigma x)}$$

$$f'(x) = \frac{\sigma}{2}[1+f(x)][1-f(x)]$$

4.8 Common Features of ANN

There are some common features of ANN which can be used to classify it accordingly, such as:

- Inputs
- No of layers(architectures)
- Storage
- Transfer (data between the layers)

ANN can be classified in terms of method of transferring data between the layers to:

- Feed forward nets: It has no connections between its units
- Feed backward nets :Its output can be used as inputs
- Auto associative nets: Its units do a multi-task, which means that they work together to receive inputs and give outputs.

There are two classes of auto associative nets

- Back Propagation
- Counter Propagation

The most used model is back propagation model

4.9 Back propagation neural nets:

In this type every neuron in a layer is connected to each of the neurons on the next layer, and there are no connections between two neurons in one layer. The input layer gives the original input of every pattern to every neuron in the first hidden layer (Fig 4.5). The input values are often restricted to an interval of [0,1] or [-1,1]. For most problems this doesn't matter much, but for some [-1,1] it is more suitable [8].

It can be used to solve problems in many areas like, image processing, speech production, character recognition…etc. The goal of the applications using this type is to train the net, then test it using an input similar to that used during the training, it responds yes, or no if the input is different.

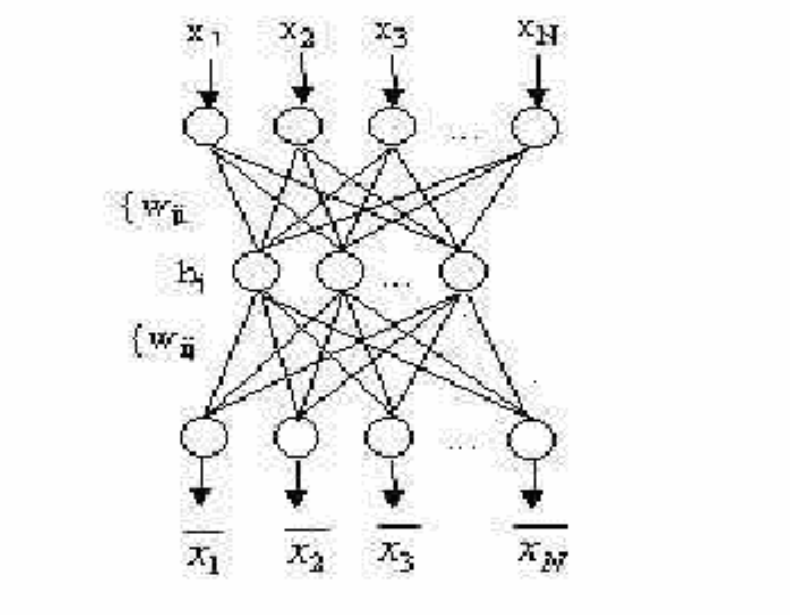


Fig 4.5 Back propagation net

4.9.1 Architecture

It is a multi layer net with more than one hidden layer. Use of two hidden layer produce an easy trained net [27]. The number of layers given by the following formulas:

1. $H \cong \frac{1}{2}(I + O) + \sqrt{p}$

   Where

   I: number of input units

   O: number of output units

   P: number of samples in the training file

2. $H \cong \log_2 I$

3. $H \cong \frac{1}{2}(I)$

4.9.2 Training by Adjusting weights

During the training period, a series of inputs are presented to the ANN- each of the form (x1,x2, ..., xn). For each such input, there is a desired output - either 0 or 1. The actual output is determined by the net input which is w1 x1 + w2 x2 + ... + wn xn. If the net input is less than threshold then the output is 0, otherwise the output is 1**.**

The Overall error in the output is calculated as the squared value of the difference between what we wanted and what we got from our net. Once we see that our net is in error, we can adjust each of the weights leading to the output node. We'll adjust each weight in such a way as to make the error value smaller. The Update Rule for a Weighted Edge of a net to update the weight from a node j leading to an output node, we adjust the weight according to the three steps [27]

Step1: Feed forward

Step2: compute back propagation error

Step3: Adjust weights

Step1:

- Here each input layer takes an input (x), sends it to a hidden layer (z) to compute the activation value.

  $Z_j = f(\sum x_i v_{ij})$

- Send this value to output layer(y)

  $Y_k = f(\sum w_{jk} z_j)$

Step2:

This step of comparison between what we need (y) and what we get (t) to determine the error, then calculate the value Θ_k

$\Theta_k = (t_k - y_k) f'(y_k)$

This value used to distribute the error by the number of input layers .

Step3:

To adjust the weights

$$\Delta w_{jk} = \alpha \partial_k z_j$$

$w_{jk}(new) = w_{jk}(old) + \Delta w_{jk}$

α Between [0.25, 0.75]

Between hidden layers, we can compute new weights using the following

- Calculate Θ_j (sum of delta input)

  $\Theta_j = (\sum_{k=1}^{m} \partial_k v j_k) f'(z_j))$

- Calculate weight correction term

  $$\Delta w_{jk} = \alpha \partial_j x_i$$

- Update weights

  $W_{ij}(new) = w_{ij}(old) + \Delta w_{jk}$

Also the following algorithm can help in net work training:

Initialize all weight to small random numbers.

Until satisfied do

1. Input training example to the network and compute the network outputs
2. For each output unit k

$$\delta_k \leftarrow o_k(1 - o_k)(t_t - o_k)$$

3. For each hidden unit *h*

$$\delta_h \leftarrow o_h(1 - o_h)\sum_{k \varepsilon output} w_{h,k}\delta_k$$

4. Update each network weight $w_{i,j}$

$$w_{i,j} \leftarrow w_{i,j} + \Delta w_{i,j}$$

Where

$$\Delta w_{i,j} = \eta \delta_j x_{i,j}$$

4.9.3 Implementation:

To build a **Back propagation**:

Collect the data: describe the data, put it in numerical form

1. Define the design of the net (define the number of layers)
2. Determine the values of the variable such as:
    - Type of the transfer functions;
    - Initial value of weights
    - Error value to stop training
3. Training and adjusting of weight
4. Testing and analysis of the result: test the net using weights

that we get during the training phase.

Chapter Five

System Development

At this stage of the research a description of the system developed and the results obtained from the program are examined.

5.1 Introduction

Arabic is a language spoken by Arabs and as a second language by several Asian countries where Islam is a dominant religion. Recognition of Arabic is different from other languages for many reasons:

Arabic letters based on 18 distinct shapes that vary according to the connections .There is no capital letters, the big difference between Latin and Arabic letters is that two letters in Arabic can differ by a dot while the general shape is the same which need a very restricted feature extraction classifier. Arabic character has different forms according to the position of the character so the system cannot rely on space between letters; this means the cursive nature of the Arabic writing makes recognition more difficult. In short,

"This field remains one of the most challenging problems in pattern recognition and all the existing systems are still limited to restricted applications"[38].

5.2 System Operation and Architecture

In this work, a system for offline signature verification for Arabic handwriting is designed as little has been done to verify Arabic handwritten signatures. The system uses data collected from users that have been scanned to get the image of signatures to use it in feature extraction phase, and then a neural network has been used in classification phase.

Signature verification process is an important task as part of other operations that need to identify a person; such as in a bank’s operation and other organizations that require presentation of someone’s official

documents. In banking field, thousands of cheques are being processed daily and many claims are being processed in insurance companies; so all are in great need to automate these processes. The proposed system shall be capable of classifying signatures as acceptable (genuine) or as rejected (e.g. a forgery or not within a acceptable tolerance).

The proposed system has 3 major phases, data acquisition phase, preprocessing phase, and feature extraction and verification phase. In the data acquisition phase images of signatures collected from different writers are scanned using high resolution scanners. In the preprocessing phase of the system, many methods can be used to prepare an image for features extraction phase. Two types of features are extracted, statistical features from which we get different degrees of momentum, and structural features from which upper and lower envelope of the signature are obtained. The extracted features are passed to a neural network for training in the verification phase. After the training and verification phase, the system can take a test signature and compare it with that the NN already acquainted with in the training phase.

5.3 Data Acquisition Phase.

Data acquisition methods have basic principles and techniques. A sample of data is a small representation of a large whole. In this case the signature samples are taken randomly since everybody signs his name differently. There should be a pattern in the way any individual signs, not in the way all individuals signs, i.e. the signatures used here are not defined in terms of number, value, age, or race. Therefore there is no reason to believe any bias has occurred in our sample.

As any sampling should have a concrete description [36], in our case signatures were taken from a homogenous community (university students/university staff) see fig [5.1]. We only need to analyze few cases

rather than analyzing mass data which would, if carried out in our case, be time consuming.

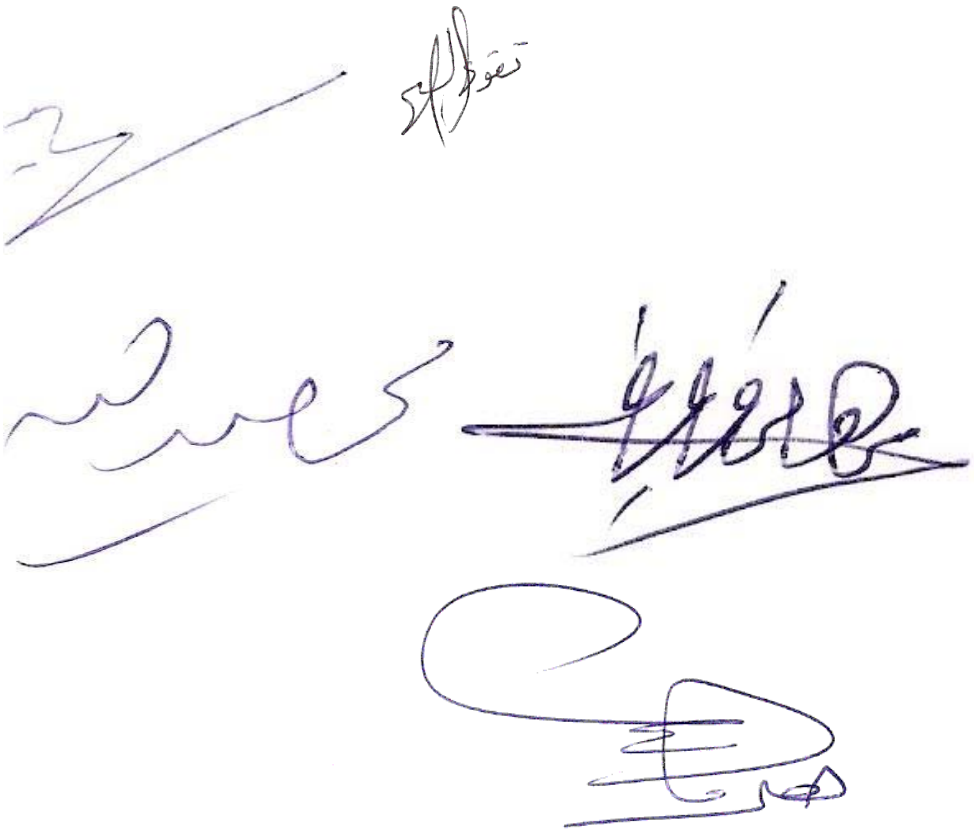


Fig (5.1) Specimens of Signatures

5.4 Pre-Processing Phase.

Depending on the data acquisition type, the signature has a number of different processing steps to make it useful in the stage of features extraction.

Preprocessing is used to remove unwanted and distracting elements such as noise. Any ordinary scanner with enough resolution can be used as an image acquisition device. The scanned signature image is preprocessed to be suitable for feature extraction phase. Scanning may introduce noise to the signature image. Another source of the noise may be paper background on which the signature assigned. The main objectives of pre-processing can be described as follows:

5.4.1 Noise reduction

A noise reduction filter is used with the binary image in order to black pixels on white background. Gaussian smoothing filter is selected to do this function. The Gaussian smoothing filter is used to remove details and noise. It has the formula

$$G(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}$$

where σ is the standard deviation of the distribution which determines the width of Gaussian. Gaussian function is symmetric, so smoothing performed by it will be the same in the two directions (vertical and horizontal), thus the edges will be in its direction [fig 5.2].

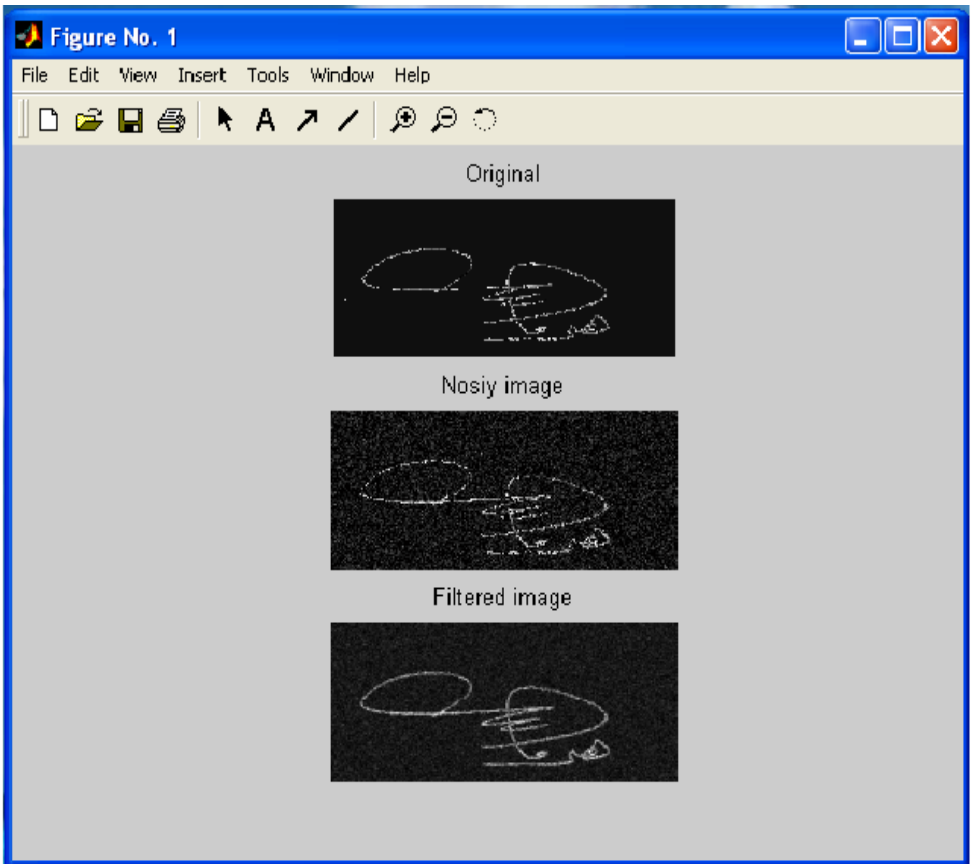


Fig (5.2) signature after filtering

5.4.2 Image cropping

The area of the image containing the signature is cropped, which means we get the bounding box, which is defined as the smallest rectangle that surrounds the image. The bounding box is used to get a matrix of the image with less number of rows and column than those of the matrix of the image without using bounding box, see(fig 5.3 and fig 5.4 Appendix 2).

5.4.3 Normalization

Usually people try to fill the space made available with their signature, so they have different size signatures according to the available space. For this reason normalization is done to normalize the size of the

image with respect to the width and height. The most used function for normalization is skew function. Normalization with respect to skew is a technique commonly used in handwriting recognition.

In handwriting recognition systems, this type is performed to recognize words independent of the writing style. In this system no normalization is done ,since in data acquisition all the signatures were collected with same style pens, so it is assumed that all signatures for each writer has the same size.

After normalization sometimes small dots remain, it needs using of morphological opening and closing operators. The opening operator with a given structuring will remove all the points which are too small to contain that structuring element. The closing operator, fill the holes and concavities smaller than that structure element [2]. Doing this is crucial step since a wrong selection for a method may give a wrong word; for example using opening operator with the letter (Faa) will give (Waaw), also using wrong closing operator with the letter (Baa) will give (Yaa) .

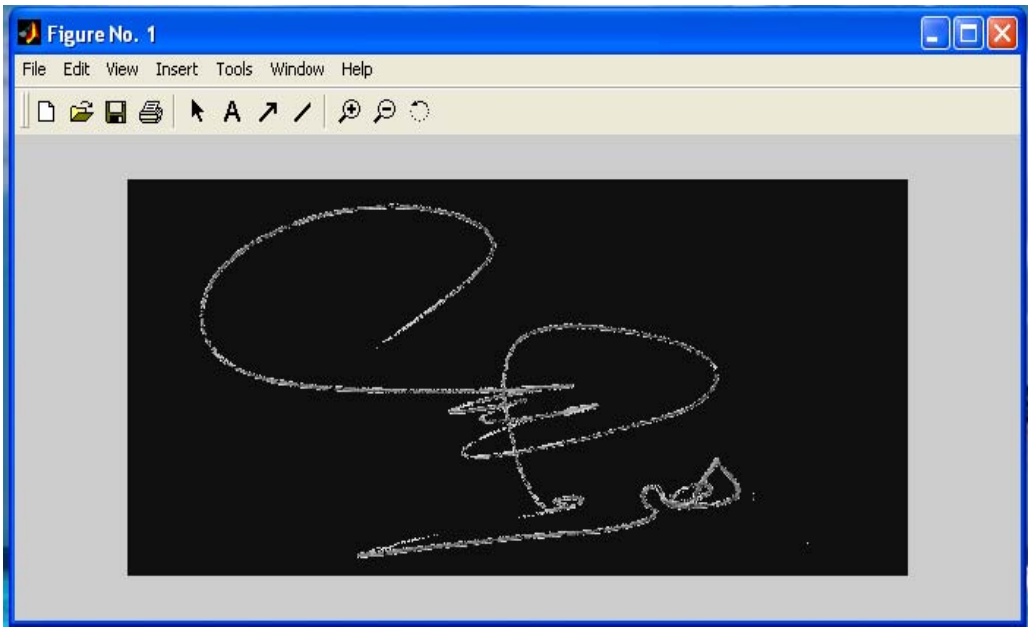


Fig (5.3) image before cropping

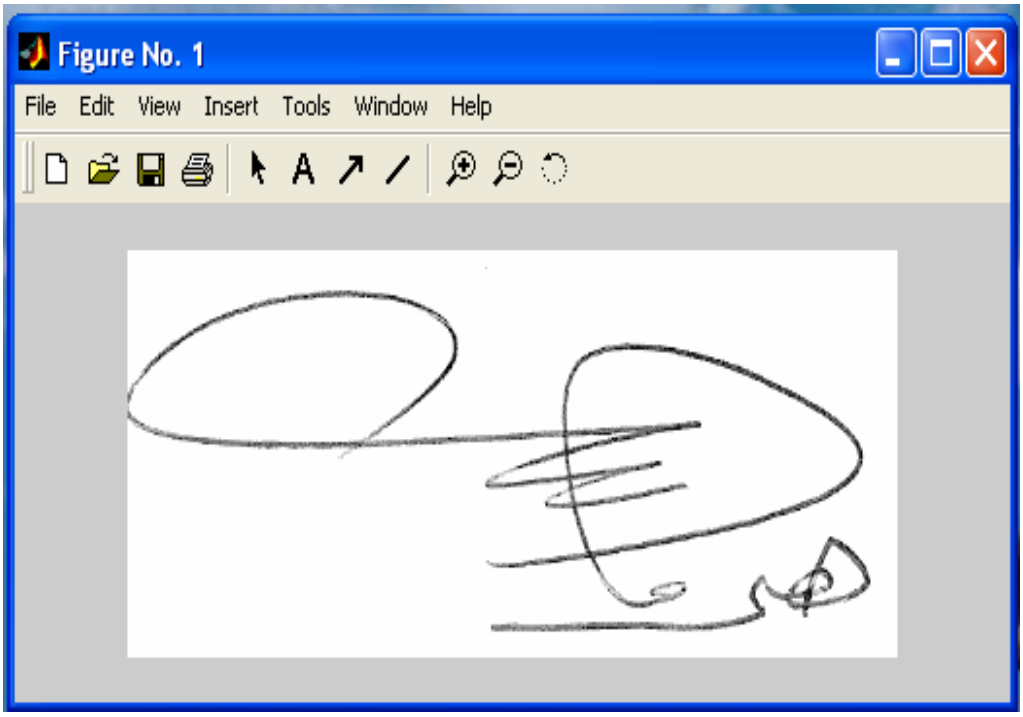


Fig (5.4) image after cropping

5.5 Features Extraction Phase.

The selection of features is critical process, because selecting wrong feature or not useful feature can give no or bad results. The most used criteria to choose features is:

"Features should contain information required to distinguish between classes, be insensitive to irrelevant variability in input, and also limit the amount of training data required" [22].

In this work three different features were extracted from the signature. These features are statistical feature based on some statistical calculations, structural features depending on the shape of the signature. After extraction signature images are of no interest, since feature vectors are what is used in verification phase .To get these features firstly we have:

1. Horizontal and vertical projection of the image (signature).
2. Lower envelop of the signature which can be defined as: the curve connecting lower most pixels of the signature trajectory.
3. Upper envelop of the signature which can be defined as: the curve connecting upper most pixels of the signature trajectory.

At the beginning let's define features as follows:

1- Horizontal projection image:

$$X(i) = \sum_{j} im(i, j)$$

2- Vertical projection image:

$$Y(i) = \sum_{i} im(i, j) \qquad [25]$$

where im(i ,j) is either 1 or 0 and I is the row index and j refers to the column, and the r^{th} order moment measure for projection is defined as :

$$\mu_r = \sum_{i} (xi - x^c) rG(xi)$$

where x^c is the centroid of the corresponding projection image and G(xi) can be either X() or Y().

The horizontal and vertical projection image can be defined as mean value of the vertical (horizontal) points on the x,y coordinates

The Moments is defined as follows: If X is a random variable, the r^{th} moment of X, usually denoted by μ_r', is defined as

$$\mu_r' = \varepsilon[X^r]$$

Note that $\mu_r' = \varepsilon[X^r] = \mu_x$, the mean of X

Central moments are defined as: if X is a random variable, the r^{th} central moment of X about A is defined as:

$$\mu'_r = \varepsilon[(X - \mu_x)^r]$$

Note that $\mu_1 = \varepsilon[(X - \mu_x)] = 0$ and $\mu_2 = \varepsilon[(X - \mu_x)^2]$ the variance of X.

This feature is used to produce skewness and kurtosis measures in the following equations, *V* indicates vertical moments computed from the vertical image and *H* indicates horizontal moments.

kurtosis measures

a) $Kv = \dfrac{\mu 4v}{(\mu 2v)^2}$

b) $K_H = \dfrac{\mu 4H}{(\mu 2H)^2}$

Skewness measures

a) $Kv = \dfrac{\mu 3v}{(\mu 2v)^{1.5}}$

b) $K_H = \dfrac{\mu 3H}{(\mu 2H)^{1.5}}$

Relative kurtosis and skewness measures

a) $R_v = \dfrac{\mu 3v}{(\mu 4v)0.75}$

b) $R_H = \dfrac{\mu 3H}{(\mu 4H)0.75}$

Relative vertical and horizontal projection measures

a) $VH_1 = \dfrac{\mu 2v}{(\mu 2H)}$

b) $VH_2 = \dfrac{\mu 4v}{(\mu 4H)}$

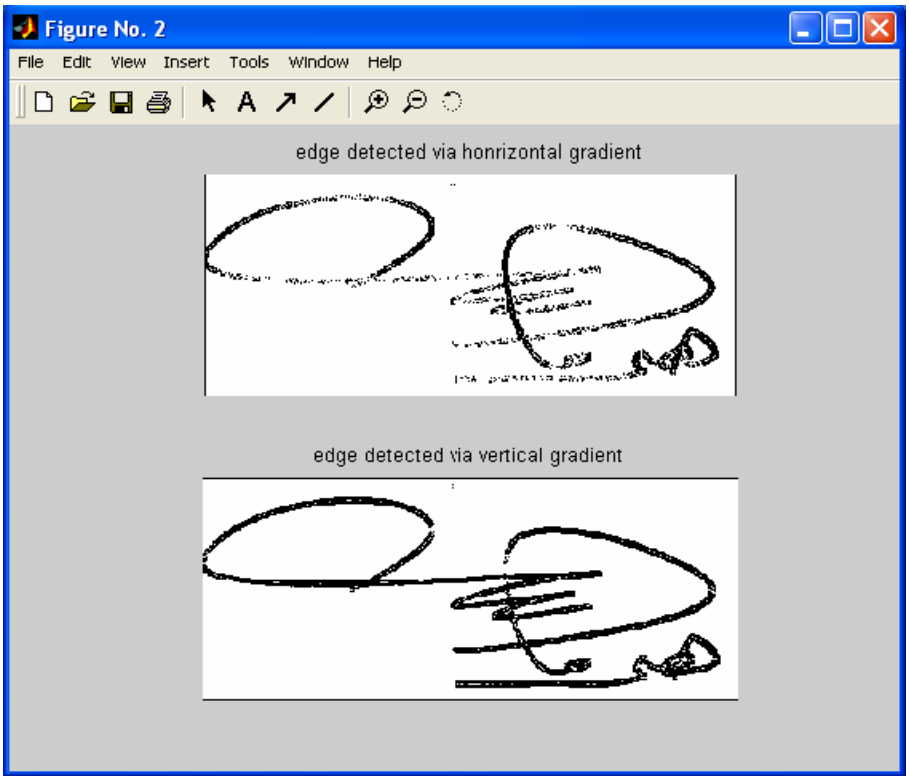


Fig (5.5) Contour using horizontal & vertical projection

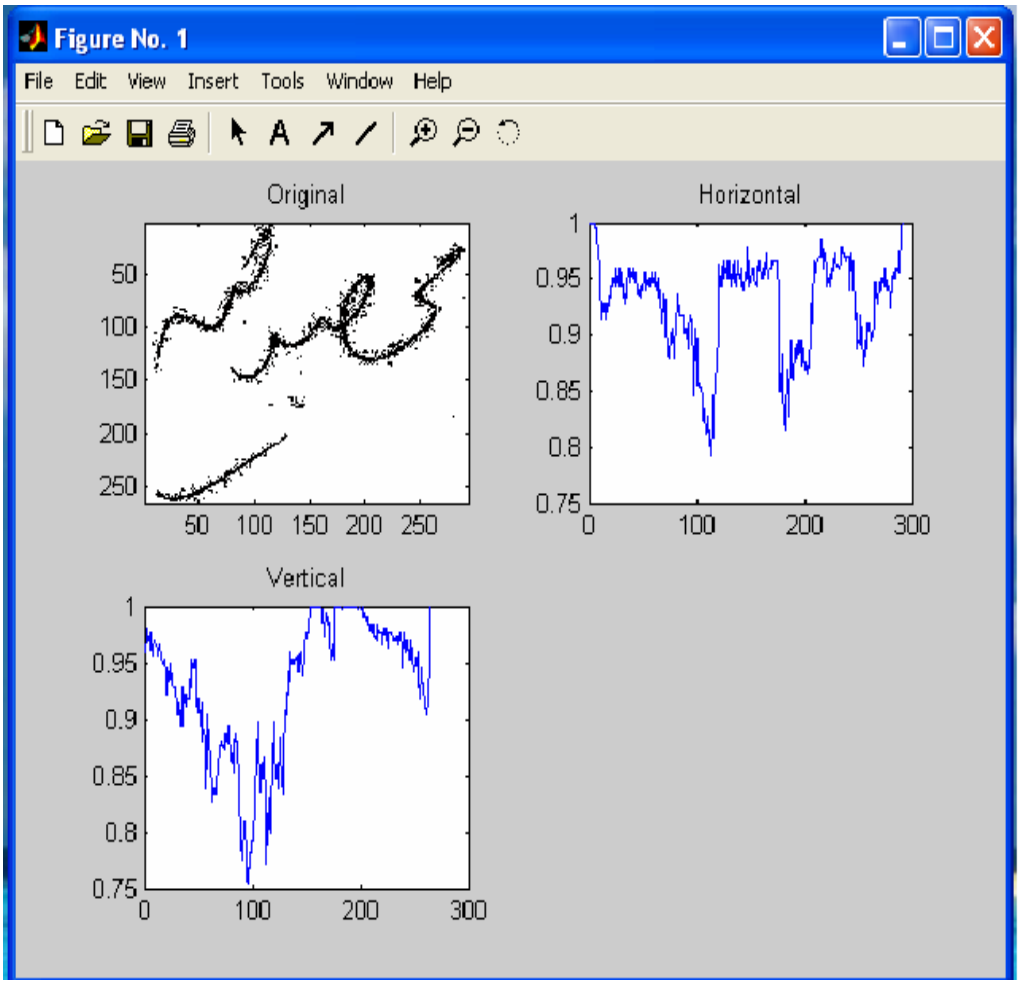


Fig (5.6) Horizontal & Vertical projection

For extracting the shape from the upper envelope each column within the bounding box surrounding the image is traced from top to bottom; the first location of a black pixel is taken as the point of the upper envelop. To extract lower envelope each column is traced from bottom to top, record the first black pixel as the stating lower point of the lower curve. Therefore, four numerical values assigned as follows:

We have to deal with the features as two types:

- Statistical features which are considered to define the different degrees of momentum; in this research we are interested in theta and radius (r,ɵ).
- The other type is Structural features that find the number of ɵs in each sector where six of them are considered.

Statistical features

To obtain "R Moments"

- get the coordinates (x, y) for black pixels of the signature.
- get the (r,ɵ) for all of the above (x,y)
- get R-avg=∑r/number of r, as well as the center of gravity
- divide all r over R-avg
- use the following equation to get the required moment

Kurtosis measures

$$Kr = \frac{\mu_{4r}}{(\mu_{2r})_2}$$

skewness measures

$$Sr = \frac{\mu_{3r}}{(\mu_{4r})_{0.75}}$$

Relative Kurtosis and skewness measures

$$Rr = \frac{\mu_{3r}}{(\mu_{4r})_{0.75}}$$

To obtain "Theta Moments ":

- get the(r, θ)for the entire picture
- convert from radians to degrees to identify the proper sector
- use the following equation to get the required moment

kurtosis measures

$$Kt = \frac{\mu_{4t}}{(\mu_{2t})_2}$$

skewness measures

$$St = \frac{\mu 3t}{(\mu 2t)2}$$

Relative Kurtosis and skewness measures

$$Rt = \frac{\mu 3t}{(\mu 4t)0.75}$$

The image of the signature after preprocessing can be used to get its contour in different forms; one of these forms is skeleton (see fig 5.5).The most used form of contour in the signature verification methods is the upper and lower contour, there are many ways to get the different points of the contour. The methods selected here to get the contour of the image is tracing.

Trace the signature upper and lower contour. To get the vector of the upper contour take each first black pixel in each column of the image matrix, this represents the upper most pixels of the signature. Also for the lower contour takes each first black pixel in each column of the image matrix, which represents the most lower pixels of the signature. Take these values as features of ANN.

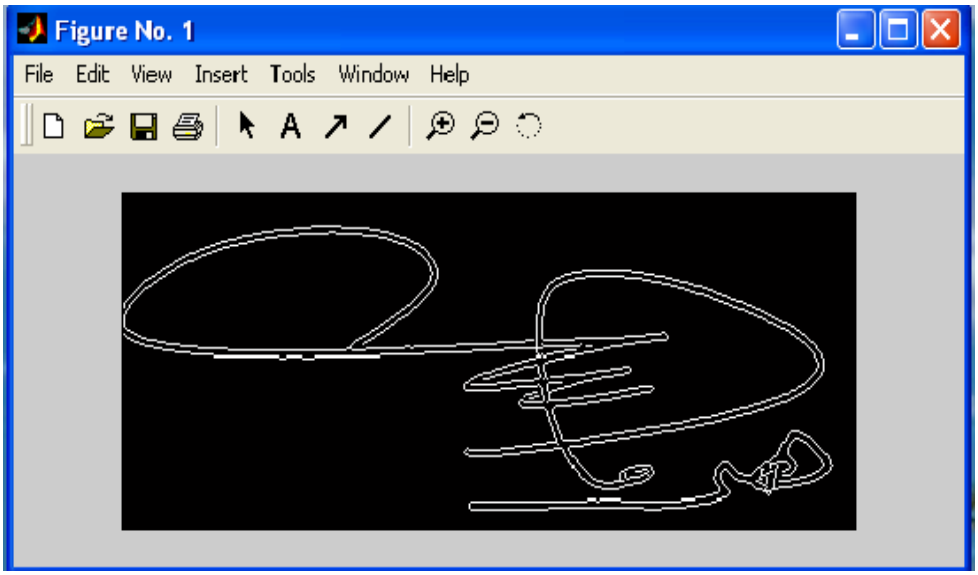


Fig (5.7) Image contour

5.6 The neural net structure

The learning stage is made up of a neural network known as multiplayer perceptron or MLR. MLR uses the back propagation algorithm to train the network.

Three feed-forward neural networks (NN1),(NN2),(NN3) were used in this system, one for each set of features (see Fig (5.8)). Each of these consists of sub-networks. The sub networks are trained such that it gives only two possible combination output neurons. If the output is (1,0), this means that the input is recognized. Output of (0,1) means that the input is not recognized. The neural networks of the system are Multi-layer perceptron.

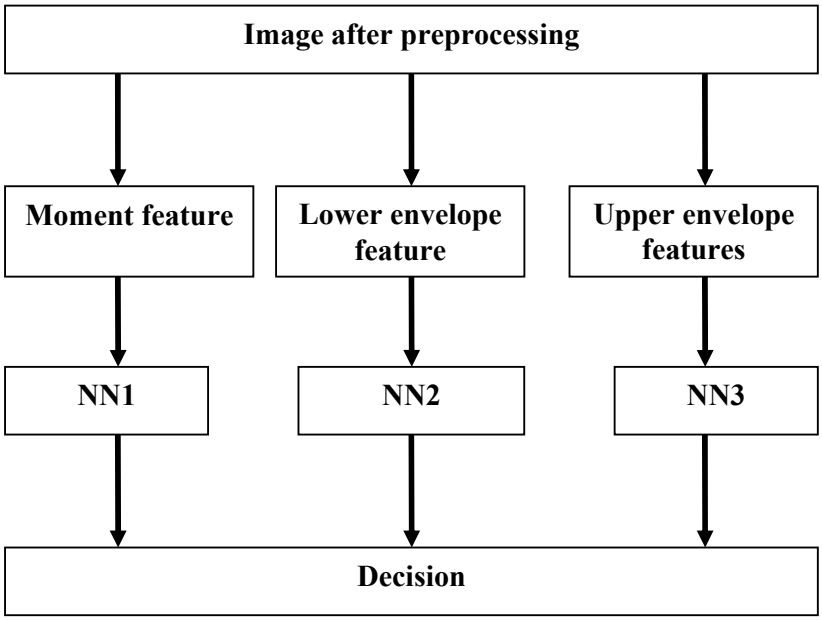


Fig (5.8) Verification System

In summary, the image of the signature is taken, in preprocessing phase it is prepared for feature extraction, and the extracted feature vectors are used for the training of the neural net.

5.7 Results

It is difficult to compare the results of this system with other systems because different systems use different sets of signatures.

For testing the performance of the system, a set of 80 signatures is used; different writers with different styles; (20 user with 4 samples of signatures for each one). First the system checked for the obvious case, in which a signature of a user is used as a replacement signature for another user. This means two different signatures here and no signature is accepted as a genuine signature to the original one.

In the second experiment, skilled forge signatures were needed, which is a subjective issue; no one can do the job of forging a signature of another; so we cannot judge whether or not it is skilled forge signature or is nearly so; for this reason the set is too small. The results of

checking the system using a (semi-skilled) signature gave 12.31% error rate; while totally replaced signatures gave 9.7% error. Results are shown in table (5.1).

Signature type	No.of signatures	FAR	FRR
Genuine	80	…….	9.73%
Forged	20	12.31%	…….

Table 5.1 Result of second experiment

It was found that some signatures are so complicated handwritten such as: it has many junction points, multiple components such as line below it see fig (5.9). Also in Arabic letters with one dot over or bellow it, the dot sometimes appears to the system as a noise, so in preprocessing phase it is ignored as a noise and is cleared.

Fig (5.9) Signature with different styles

An experiment done on a small sub set of signatures with those complications and gave high rate of error table (5.2).

Signature type	No.of signatures	FAR	FRR
Genuine multiple components	**8**	**25%**	**37.3%**

Table (5.2) Error rate for more complex signatures

Another experiment was done using a signature written in two language see fig (5.10) this signature is combination of Arabic and English, 7 signatures taken from the writer for the test, the system accept three of them and reject the other. It need to check the system with set contain a signatures from different users to comment on this results

Fig (5.10) signatures written in two languages

Also small set of continues handwritten signatures such as signatures written in English language see (fig 5.11) is used to check the system, It giver results of 15% FAR As general, good comments can not be given with a test done on a very small set. Big set is a time consuming because the training needs a lot.

Fig (5.11) continuous signatures

Chapter Six

Conclusion & Recommendation

This chapter is meant to summarize the final conclusions of this work as well as suggesting some recommendations that stem out of it.

6.1 Conclusions:

This thesis has aimed to deal with what appears to be an everlasting problem related to handwriting signature verification. This challenging problem arises from two facts:

1. That handwritten signature is still needed as identification to identify a person on all official matters and to establish his/her authenticity (in cheques, contracts, certifications…etc).
2. Handwritten signature is not an biological feature; rather it is a behavioral one. Unlike fingerprints, its recognition is not always easy or certain. Therefore it can be forged and hence the identity of the signer can be forged.

Attempts were made, and still ongoing, to develop a computer system to solve the problem or at least to ease the task for human experienced examiners. This thesis is no more than to attempt in that direction.

The thesis has confined itself to offline signature verification (one of two types of handwritten signatures; offline and online).The choice to address offline handwriting signature was because it is easy to forge due to variations found in one genuine signature resulting from behavioral factors, or otherwise.

Based on pattern recognition techniques methodologically three feature extracted from signatures were chosen:

1. Vertical and horizontal projection (which used to find the moments)
2. Upper envelope
3. Lower envelope

These extracted features were to train an artificial neural network.

To compare two signatures, features were extracted from the tested signature are matched with that the net knows.

To develop a system we started by data acquisition where 20 users were chosen randomly and asked to supply 4 genuine signatures each. This first set of signatures was supplied to the network to train it after normalization by getting it used to the familiarity of the signatures so as to serve as reference set.

Another genuine set of signatures was taken, one for each of the 20 users, for validation, that is to match it with the reference set and test the degree of similarity with two sets and see how distinct or close the one signature to any, or all, of the 4 signatures in the system, and so to classify the signature as genuine or otherwise.

To cross-examine the ability of the system to discover forgery; and as it was difficult to get real forgeries to test our set of genuine signatures, we turned to unprofessional forgers who have access to the genuine signatures. They had provided a third set of signatures which were supplied to the system as genuine through forged.

The result was very encouraging. Only 9% of the genuine signatures were rejected by the system, while 12.3% of the forged signatures were accepted as genuine. This results show that the system has performed well and can be trusted, especially when compared with previous other counterparts (with 20-25% errors) or that of human examiners. Yet it is no more, we hope, than one step in right direction.

6.2 Recommendations:

This thesis has contributed positively towards the fulfillment of its set of objectives. Yet a great deal remains to be done. Certain recommendations stem out of it. Among these are:

1. Using the system, as it is tested, to collect and scan the signatures is difficult; as official authorities (bank…etc) do not allow access to such data, and a bigger sample is needed for increasing the degree of confidence and the validity of the system.
2. More samples of genuine signatures per person are needed (i.e. 10 signatures instead of 4). This will allow the system to tolerate more variation in signature of each user. The system will be familiar with these variations from the genuine signatures collected and so will reduce the degree of rejection of genuine signatures. One should seek a balance between the level of rejections of genuine signatures and acceptance of forged signatures since increasing the number of samples will increase the system tolerance to variations in same signatures.
3. Further tests are needed to attempt using signature of different languages (using different writing styles such as in Arabic and English) to evaluate the system performance for the two languages and whether there is any significant difference in the results obtained in each case.
4. The system needs to incorporate a filtering techniques to avoid all 'noise' usually appear in offline signatures such as pen size, ink, signer state of mind…etc will contribute towards the betterment of the system.
5. The system needs a robust method for filtering, such as a good use of closing and opening operator to solve the problem arise during experiment in which the system delete the line or dot below the word.
6. A trail is needed to get the upper and lower contour using another method which uses a grid to take the peaks and valley

of the shape of the signature into consideration. In addition, a method to deal with rotated images of a signature is needed.

7. Our sets used for the test do not contain a set of real skilled forge data (no way to get this type of data).The result would be more valuable if true skilled forgeries where available.
8. Test the system using a classifier system other than neural net work, like Dynamic Plane Warping algorithm (DPW has been applied in field of optical character recognition to solve many problems) to find if using different methods make difference and build a system with very high accuracy.

References

1. A.K.Jain,F.D.Griess,and S.D.Connell, "On-line Signature Verification", **Patteren Recognition**,Vol.35,pp.2963-2972,Dec.2002
2. Alsher Anatolyevich Kholmatov "Biometric Identity Verification Using On-line and Off-Line Signature Verification", MSC dissertation, Graduate school of engineering and Natural science, Sabanci University,ex.Yogaslavia 2003.
3. B.Due, E .S Bigün, J. Bigün,G. Maî, S.Fischer "Fusion of Audio and video Information For multi Model Person Authentication." **Pattern recognition letters**, Vol. 18, No. 9, 1997.
4. D.Maltoni,D.Maio,A.K.Jain,S.prabhakar "Handbook of Fingerprint Recognition". Springer New York, 2003.
5. Dr.Kamal M.Jambi "Different Approaches for Arabic signature Recognition", King Abd-alaziz City for Science Technology project number AR-15-20, 1998
6. H.Dullink,B.van Daalen "Implementing a DSP Kernel for Online Dynamic Handwritten Signature Verification Using the TMS320 DSP Family", EFRIRE,Texas Instruments ,SPRA304,December 1995. EFRIRE,France ,December 1995.
7. http://annevolve.sourceforge
8. http://iit.demokritos.gr/neural/intro
9. http://U-aizu.ac.jp
10. http://www.A Short Introduction to Digital Image Processing.htm
11. http://www.biometrics.csc.msu/figure print
12. http://www.carleton.ca/neil/neural/neuron-a.html
13. http://www.niii.ru.nl

14. Ian T. Young, J.J. Gerbrands " Fundamentals of Image Processing " ,Netherlands Delft University of Technology.1998
15. J.G Taylar , **Neural Networks** king college 1995
16. John D.Woodwards, Jr. **Biometrics,** 2003
17. L. Fausett, **Fundamentals of Neural Networks**, Prentice-Hall, New Jersey, 1994
18. Lin Hong,Tifei Wan" Fingerprint Image Enhancement" Michigan State university, 2003
19. Maria Petrou&P.Bosdogianni **,Image Processing The Fundamentals,**John Wiley &Sons,LTD April 2000
20. Meenakshi K. Kalera " offline signature verification and identification using distance statistics" International Journal of **Pattern Recognition and Artificial Intelligence**,Vol. 18, No. 7,pp1339-1360, 2004.
21. Nabeel A.Murshed ,Flavio Bortolozzi and Robert Sabourin "Off-Line Signature Verification Using Fuzzy ARTMAP Neural Network".
22. Ondřej Rohlík "Handwritten Text Analysis", Ph.D. dissertation, Department of Computer Science and Engineering, University of west Bohemia in Pilswen, 2003.
23. http://www.ece.arizona.edu/~pgsangam/
24. R. chellappa, C. L. Wilson, S. Sirohey "Human and Machine Recognition of faces "A Survey Processing of the lEEE, Vol.83,No. 5, 1995.
25. R.Sabourina and G.Genest, "An Extended-Shadow-Code Based Approach for Off-line Signature Verification ",ICDAR, 1993.
26. Suliman M Mohamed "Fingerprint-Based Biometrics Recognition Allied to Fuzzy-Neural Feature Classification" Ph.D. dissertation. Sheffield Hallam University 2002.

27. هويدا علي " تمييز بعض الآلات بأصواتها باستخدام الشبكات العصبانية " بحث مقدم لنيل درجة الدكتوراة في علوم الكمبيوتر جامعة الخرطوم 2003

28. Rangachar Kasturi "Document image analysis: A primer", **Sadhana** Vol. 27, Part 1, February 2002, pp. 3–22.

29. http://www.gisdevelopment.net/tutorials/tuman005pf.htm

30. http://www.vision.caltech.edu

31. N.Papamarkos, and H.Baltzakis "Offline Signature recognition using multiple Neural Network classification structures" **Engineering Application of Artificial Intelligence**, Vol. 4,pp 95-103, 2001.

32. Diana Kalenova " Personal Authentication Using Signature Recognition**"**, Department of Information Technology, Laboratory of Information Processing, Lappeenranta University of Technology,2002

33. J. L. Wayman, “Digital Signal Processing in biometric identification: A review,” Proc. of ICIP-2002, vol. 1, pp. 22-25, 2002.

34. B.Fang,C.H Leung,Y.Y.Tang,K.W.Tse "Off-line Signature Verification by tracking of feature and Stroke Positions",**Patteren Recognition**,Vol.36,pp91-101,2003

35. J.K.Guo,D.Doermann "Forgery Detection by local Correspondance", **Patteren recognition and Artificial Intelligence**,Vol.15,pp.579-641,2001

36. Goode,w.and Hatt, **Methods in Social Research**,Mc Graw Books,New york 1952

37. Clabresi,M.et.al.” Time magazine”, p 26-35,vol 160,24,december 2002

Appendix 1

Gaussian filtered program

```
clear;
% Parameters of the Gaussian filter:
n1=10;sigma1=3;n2=10;sigma2=3;theta1=0;
% The amplitude of the noise:
noise=0.1;
I =imread('C:\Sig-data\h3.gif');
I2 = histeq(I)
 bw = im2bw(I2,0.4)
% filter1=d2gauss(n1,sigma1,n2,sigma2,theta);
filter1=d2gauss(10,3,10,3,0);
bw_rand=noise*randn(size(bw));
y=bw+bw_rand;
f1=conv2(bw,filter1,'same');
rf1=conv2(y,filter1,'same');
figure(1);
subplot(2,2,1);imagesc(bw);title('Original');
subplot(2,2,2);imagesc(y);title('noisy image');
subplot(2,2,3);imagesc(f1);title('smooth');
subplot(2,2,4);imagesc(rf1);title('noise cancel');
colormap(gray);
```

Appendix 2

bounding box

```
I=imread('C:\Sig-data\h3.gif')
bw = im2bw(I,0.4)
 % Find the boundary of the image
[y2temp x2temp] = size(bw);
x1=1;
y1=1;
x2=x2temp;
y2=y2temp;
% Finding left side blank spaces
cntB=1;
while (sum(bw(:,cntB))==y2temp)
   x1=x1+1;
   cntB=cntB+1;
end
% Finding right side blank spaces
cntB=1;
while (sum(bw(cntB,:))==x2temp)
   y1=y1+1;
   cntB=cntB+1;
end
% Finding upper side blank spaces
cntB=x2temp;
while (sum(bw(:,cntB))==y2temp)
   x2=x2-1;
   cntB=cntB-1;
end
```

```
    plot(cntB)
% Finding lower side blank spaces
cntB=y2temp;
while (sum(bw(cntB,:))==x2temp)
    y2=y2-1;
    cntB=cntB-1;
end
```

Appendix 3

Detect the edges

```
clear;
close all;
I=imread('C:\Sig-data\hend3.gif')
% Sobel filter horizontal
Hfilter = [-1 0 1;-2 0 2; -1 0 1];
% SObel filter vertical
Vfilter = [-1 -2 -1; 0 0 0; 1 2 1];
% image filtering
filteredImg1=filter2(Hfilter, I);
filteredImg2=filter2(Vfilter, I);
% Compare results
figure;
colormap(gray);
subplot(3,1,1);
imshow(I);
title('Original');
subplot(3,1,2);
imshow(filteredImg1,[]);
title('Honrizontal gradient');
subplot(3,1,3);
imshow(filteredImg2,[]);
title('Vertical Gradient');
% edge detection
threshold=80;
% use horizontal gradient to detect edge
index=find(abs(filteredImg1(:))>threshold);
```

```
edge1=255*ones(1, prod(size(I)));
edge1(index)=0;
edge1=reshape(edge1, size(I));
% use vertical gradient to detect edge
index2=find(abs(filteredImg2(:))>threshold);
edge2=255*ones(1, prod(size(I)));
edge2(index2)=0;
edge2=reshape(edge2, size(I));
figure;
subplot(2,1,1);
imshow(edge1/255);
title('edge detected via honrizontal gradient');
subplot(2,1,2);
imshow(edge2/255);
title('edge detected via vertical gradient');
```

Appendix 4

Horizontal and vertical projection

```
I=imread('C:\Sig3\ms11.gif')
I2 = histeq(I);   % Equalize I and output in new array I2.
bw = im2bw(I2,0.4)
h=mean(bw);
v=mean((bw)');
subplot(2,2,1);imagesc(bw);title('Original');
subplot(2,2,2);plot(h);title('Horizontal');
subplot(2,2,3);plot(v);title('Vertical');
% subplot(2,2,4);imagesc(rf1);title('noise cancel');
colormap(gray);
```

Appendix 5

Find moments

```
function [mom] = central_moments(inimage, maxorder)
I=imread('C:\Sig3\ms11.gif')
I2 = histeq(I);   % Equalize I and output in new array I2.
 bw = im2bw(I2,0.4)
[xpos, ypos, val] = find(im2double(I));
[xpos, ypos, val] = find(im2bw(I));
mom(1,1) = sum(val);
if (mom(1,1) == 0)
   mom = zeros(maxorder+1);
   return;
end
x0 = sum(xpos.*val)/mom(1,1);
y0 = sum(ypos.*val)/mom(1,1);
for xpow = 0:maxorder
   for ypow = 0:maxorder
     mom(xpow+1, ypow+1) = sum( ( (xpos - x0).^xpow ).*( (ypos -
y0).^ypow).*val );
   end
End
```

Appendix 6

Upper &lower contour

```
I=imread('C:\Sig3\ms11.gif')
I2 = histeq(I);   % Equalize I and output in new array I2.
 bw = im2bw(I2,0.4)
 L = bwlabel(bw);
  [i j] = size(bw);
  for a=1:j
      for b=1:i
      if (bw(a,b)==1)
          upper(a)=1;
    break
      end
  end
end
% subplot(2,2,2);plot(up(a));title('Horizontal');
plot(lower(a));
 for a=1:j
      for b=1:i
      if (bw(a,b)==1)
          lowe(a)=1;
    break
      end
  end
end
% subplot(2,2,2);plot(up(a));title('Horizontal');
plot(lower(a));
```

Appendix 7

Back Propagation

```
clear all;
close all;
tic;
Inp = [0 0; 0 1; 1 0; 1 1]; % Input
Targ = [0 ; 1 ; 1 ; 0];
%Tp = 0.20;      %Xo(1) = input('Enter First Input : ');
%Xo(2) = input('Enter Second Input : ');
%Tp = input('Enter Targeted Output : ');
Wi = [1.9 1.9; 17 17];
Wo = [-20 20]';
Bi = [0.5 -1];
Bo = -1;
eta = 0.3;
S = 1;
i = 1;
p = 1;
Xo = Inp(1,:);
L = Wi*Xo' + Bi';
 % L Vector consists of output from each hidden node
O = sigmf(L,[0.5 0]);
% Output Units
M = Wo'*O + Bo;
%Out_Hid = tansig(M);
Out_Hid = sigmf(M,[0.5 0]);
Tp = Targ(1);
```

```
MSE = 10;
while MSE > 0.25
%while abs(Tp - Out_Hid) > 0.20
L = Wi*Xo' + Bi'; %output before HL from the hidden layer
O = sigmf(L,[0.5 0]); % output of the hidden layer
M = Wo'*O + Bo; % Before HL from the output
Out_Hid = sigmf(M,[0.5 0]);%output
% Error for output units
   del_o = Out_Hid*(1-Out_Hid)*(Tp - Out_Hid); %%% For output Unit
del_j = O.*(1-O).*(del_o.*(Wo));  %%% 1-by-2 vector giving 2 errors
Wo = Wo + (eta*del_o).*O;

%%% CHECK IT AGAIN
Wi = Wi + eta.*(del_j*Xo);  %%% Input Weight Updation



disp('Iteation Number');disp(S);
Mean_Square_Error = MSE
Err(S) = abs(Tp - Out_Hid);
if(mod(S,4) == 0)
MSE = sqrt(sum(Err(S-3:S).^2));
% hold on
end

S = S+1;
%i = i+1;
i = mod(i,4)+1;
Xo = Inp(i,:);
Tp = Targ(i);
```

```
% if(S>50),
% for g = 1:4:S-5,
%
end
disp('*** RESULTS ***')
Final_Input_Weights = Wi
Final_Output_Weights = Wo
Total_Number_of_Iterations_Taken = S
Total_Time_Taken_in_Seconds = toc
% Plot Results
for i = 1:4:S-5,
        K(p) = sqrt(sum(Err(i:i+4).^2));
   p = p+1;
end
%MSE = sqrt(sum(Err(S-3:S).^2));
plot(K)
hold on
grid on;
xlabel('Iteration Number --->');
ylabel('Mean Square Error');
title('Convergence of BackPropagation Algorithm')
```

Appendix 8

Outline the Contour

```
I= imread('C:\Sig-data\un5.gif');
   edgeim = edge(I,'canny', [0.1 0.2], 1);
   figure, imshow(edgeim);
   % Link edge pixels together into lists of sequential edge points, one
   [edgelist, labelededgeim] = edgelink(edgeim, 10);
      % Display the labeled edge image with separate colours for each
   figure, imagesc(labelededgeim);
  colormap (vga), axis image, axis off
      tol = 2;
 % Line segments are fitted with maximum deviation from
   angtol = 0.05;
 % Segments differing in angle by less than 0.05 radians
   linkrad = 2;     % and end points within 2 pixels will be merged.
   [seglist, nedgelist] = lineseg(edgelist, tol, angtol, linkrad);
     drawseg(seglist, 3, 2); axis on
```

Appendix 9

Training the system

```
%%% Read the image
I=imread('C:\Sig3\ms11.gif')
I2 = histeq(I);   % Equalize I and output in new array I2.
 bw = im2bw(I2,0.4)
% bw2 = im2bw(I,0.4) %تحويل الصورة الى ثنائية
%% Image Preprocessing
img = edu_imgpreprocess(bw);
for cnt = 1:20
   bw2 = edu_imgcrop(img{cnt});
   charvec = edu_imgresize(bw2);
   out(:,cnt) = charvec;
 end

%% Create Vectors for the components (objects)
P = out(:,1:40);
T = [eye(10) eye(10) eye(10) eye(10)];
Ptest = out(:,41:50);

%% Creating and training of the Neural Network
net = edu_createnn(P,T);

%% Testing the Neural Network
[a,b]=max(sim(net,Ptest));
disp(b);
```