FEATURE EXTRACTION & COMPARISON OF SIGNATURES FOR OFFLINE VERIFICATION

Project work done at

System Logic Solutions Ltd., Bangalore

PROJECT REPORT P-786

Submitted in partial fulfillment of the requirements

for the award of the degree of

MASTER OF COMPUTER APPLICATIONS
Of Bharathiar University, Coimbatore.

Submitted by

Lekshmi.J.

9938M0615

GUIDED BY

Mr.Mahesh Jagirdar M.E.,
EXTERNAL GUIDE

Ms.V.Geetha M.C.A.,
INTERNAL GUIDE

Department of Computer Science & Engineering

KUMARAGURU COLLEGE OF TECHNOLOGY

Coimbatore – 641 006

May 2002

Department of Computer Science & Engineering

Kumaraguru College of Technology

(Affiliated to Bharathiar University)

Coimbatore – 641 006

CERTIFICATE

This is to certify that the project work entitled

FEATURE EXTRACTION & COMPARISON OF SIGNATURES FOR OFFLINE VERIFICATION

Done by

Lekshmi.J.

9938M0615

Submitted in partial fulfillment of the requirements

for the award of the degree of

Master of Computer Applications of **Bharathiar University.**

S. Thangasamy

Dr.S.Thangasamy

Professor & Head

Ms.V.Geetha

Internal Guide

Submitted to University Examination held on 10.05.2002

10/5/02

Internal Examiner

External Examiner

ystemLogic
lutions that work

SystemLogic Solutions Ltd.,
1281, 21st Main,
2nd Phase, J P Nagar,
BANGALORE - 560 078.

Phone : +91-80-659 3005, 659 6842
Fax : +91-80-659 6653
Web : www.systemlogicindia.com
E-mail : inquiry@systemlogicindia.com

Mar 25, 2002

TO WHOMSOEVER IT MAY CONCERN

This is to certify that ***Ms. Lekshmi J, MCA*** *(VI Semester) Student of* ***Kumaraguru College of Technology, Coimbatore,*** *has successfully completed a Project at our concern, as per the details given below.*

Project Title: ***Forgery Detection & Analysis of Offline Signatures.***
Project Period: ***December 2001 to March 2002.***
Technologies: ***C++ 3.0, OTL 4.0, MS-Access.***

This Project has been successfully implemented, and has met our requirements. She has completed the following modules – Feature Extraction & Comparison of Signatures for Offline verification. Her conduct was found excellent through out the project period. I also rate her to be technically very good and innovative in her approach to solving problems.

I wish her all success in her future endeavors.

With Best Regards,

For SystemLogic Solutions Ltd.

Vivekananda Hallur
Vice President

DECLARATION

I hereby declare that the project entitled **'Feature Extraction & Comparison of Signatures for Offline Verification'**, submitted to Bharathiar University as the project work of Master of Computer Applications Degree, is a record of original work done by me under the supervision and guidance of **Mr.Mahesh Jagirdar ME., Project Manager, System Logic India Pvt Ltd., Bangalore** and **Ms.V.Geetha M.C.A., Senior Lecturer, Kumaraguru College of Technology, Coimbatore** and this project work has not found the basis for the award of any Degree / Diploma / Associationship / Fellowship or similar title to any candidate of any university.

Station: Coimbatore

Date: 26-04-2002

Signature of the student,

Lekshmi.J.

Acknowledgements

I wish to express my deep sense of gratitude to r.K.K.Padmanabhan B.Sc.,(Engg), M.Tech.,Ph.D., Principal, Kumaraguru College of echnology, Coimbatore and Dr.S.Thangaswamy BE.,(Hons)., Ph.D., Professor, Head f Department, Computer Science & Engineering, Kumaraguru College of echnology, Coimbatore, for providing me with an opportunity to take up this project.

I would like to express my sincere regards and a big word of hanks to Ms.V.Geetha M.C.A., Senior Lecturer, Department of Computer Science & Engineering, Kumaraguru College of Technology, Coimbatore for her technical upport, encouragement, consistent guidance and inspiration all through my ssociation with her.

I am very much thankful to Mr.Mahesh Jagirdar ME., Project Manager, System Logic India Pvt Ltd., Bangalore, for his constant support, guidance, encouragement and help in learning many new things during my project period.

With profound sense of gratitude, I acknowledge the guidance and support extended by Ms.N.Chandrika M.C.A., Mr.R.Raghavendra BE., SystemLogic India Pvt Ltd., Bangalore, for their technical assistance, incessant encouragement and invaluable technical support have been of immense help in realizing this project. I am also thankful to all the members of our team, SystemLogic India Pvt Ltd., Bangalore. Their timely suggestions have greatly contributed in successful completion of this project.

I take a great pleasure in thanking all the faculty members of Department of Computer Science & Engineering, for their co-operation and encouragement during my study period. It was my parents and friends who really stood behind me giving all the support for my work.

Lekshmi.J.

Synopsis

With an intention to identify skilled forgeries met with offline signatures the gnature verification system was designed using Unified Modeling Language with Rational oftware Corporation's, Rational Rose 2000 and was programmed using Turbo C++ v3.0 or System Logic India Pvt Ltd., Bangalore.

Signature is a proof of authorship of any individual and therefore is a content of alue when represented over any legal document. Hence signature verification is now an ctive research area that aims to identify whether a signature is genuine or forged.

The thesis introduces an effective approach to extract the static features from the pre-processed signature samples with respect to the offline signature verification system. Both the specimen signatures, which are true samples of author and the questioned signature files in the windows bitmap format that are already clipped and zoomed are the inputs to the system.

Each signature file acquired is split into two horizontal segments. From these individual segments six features are calculated based on the sub stroke alignments. The segmentation of signature files and feature analysis is carried out for every signature sample. The aggregation of the values derived from feature extraction is calculated and stored. For each value, a tolerance value is found to accommodate slight variations. This increases the efficiency of the systems perfection in the forgery detection process.

This tolerance value of every extracted feature of the specimen when compared with the feature values obtained with the questioned signature are compared and the result of comparison made is checked whether or not it falls within the tolerance limits fixed. The percentage of variation of these tolerance values will facilitate the system to arrive at conclusions for the genuineness of a signature. The system is also designed to support its conclusions with the images of signature samples.

Table of Contents

1. Introduction

1.1 About the Organization

System Logic is a software solutions company, which was established in the year 1994 in US and later spread its wings to India in 1996.

The company offers comprehensive enterprise-wide and e-business solutions to a wide range of clients across the world.

System Logic, today, boasts of a client base of topnotch Indian companies and multinationals based in India and overseas, employee strength of 200 and a turnover of USD 7.0 million.

1.2 Problem Description

Signature is the way a person writes his name. They may be stylish or unconventional and have many personal characteristics that are challenging to reproduce by anyone other than the original author. For this reason, signatures are used and accepted as proof of authorship or consent on personal checks, credit purchases and legal documents.

Automatic signature verification is an extremely active research area. Although similar systems exist in the market, there are few that can promise sufficiently high accuracy rates at a reasonable level of efficiency.

Even today, signatures are verified only informally in many environments, but the rapid development of computer technology has stimulated great interest in research on automated signature verification and forgery detection.

Forgeries of signature can be Simple, Substitution and Freehand or Skilled. In this project, we focus on detection of skilled forgeries of offline signatures where the true and forged samples are almost alike.

1.3 Platform

The project was developed using Turbo C++ thus capturing the object orientation features of the language and serves as a very useful tool in investigations of alleged authentication forgeries. It can be used as a part of the security system in the banking sector, or in the matters involving any legal documents.

2. Problem Definition & Feasibility Analysis

Understanding the problem domain and the measures to solve it are the matters of significance on building a software intensive system. This phase under the software development life cycle is the layer of analyzing the requirements of the new system based on the limitations of the existing system.

2.1 Problem Definition

The primary requirement here is for an efficient standalone system, for verifying the signatures of various individuals in order to authenticate their genuineness .In addition to this, the system must impose access restrictions to prevent unauthorized users from tampering with the sensitive data. Periodic report generation is also needed.

2.2 Existing System-Limitations

For authenticating a signature there are two techniques available:

- **Manual verification:**

Comparison is based on clearly distinguishable features.

- **Automatic Verification:**

This can be classified broadly into two types:

1) Online verification System:

This is a computerized comparison technique that takes into account parameters such as the speed of signing, patching, retouching, pressure habits etc. These are measured dynamically while signing.

2) Offline Verification System:

These are static systems which judges the accuracy of signatures based on the static features of a signature and are useful in cases where like verification of legal documents where the signing has already been carried out.

The existing technique followed is a Manual verification system, wherein the comparison between two signatures is done manually based on clearly distinguishable standards. This technique mainly depends on the person's powers of observation.

It is assumed that there are certain characteristics in a person's signature, which are difficult to reproduce. There is a chance that these features might be overlooked when performing the verifications manually. Moreover, clearly distinguishable features are easy to forge, and a skilled forger can easily reproduce a signature with proper practice. So, the percentage of accuracy reduces in a manual verification system.

2.3 Client Requirements

The following are the requirements specified by the client.

- An efficient stand-alone offline signature verification system to verify the genuineness of the signatures.

- This being a security based system needs a proper authentication module for restricting the user entry.

- The system should analyze signatures, arrive at proper conclusions and justify the same using proper images & text. No batch processing is required.

- Ad hoc reports generation provision should be included.

- Provision for further enhancements to promote it into an online system & support more image formats.

2.4 The Proposed System

The new system is a standalone "Offline Signature Verification" system that attempts to authenticate the genuineness of a questioned signature by comparing it with various copies of the original signature.

In Offline systems, the signature acquisition is carried out long after the writing process actually occurs. These systems are often called as static systems, since the processing is done on static data. This method of verification is useful when authentication of records is needed in the absence of the individual concerned.

The system takes in the scanned images of signature specimens, analyses and compares the samples based on certain features with that of the questioned signature and arrives at a conclusion about its genuineness .The System produces a justified conclusion based on which it is decided whether the signature is forged or not.

2.5 System Inputs & Outputs

The system takes in clipped bitmap images and produces the result of comparison with the genuine signature samples as output.The details are given below.

2.5.1 Inputs

The system takes in processed bitmap images of signatures, which are already zoomed and clipped as input. These are then split horizontally into segments for further analysis. The number of signatures considered for analysis depends on the number of samples obtained after preprocessing.

2.5.2 Outputs

Based on the conclusions derived from analysis of the signature samples, the system lists whether a signature has been forged or not. For both the genuine as well as the forged samples the system displays bitmap images and relates the reasons for the conclusions along with the probability of accuracy of the conclusion.

2.6 User characteristics

Familiarity with Windows operating system is expected from the user of the system. He should also be aware of the file format in which the image has to be stored, as the System deals with .bmp files only.

3. Programming Environment

The support for achieving the effective system implementation depends upon the proper selection of the software and the tools. Hence the technical necessaries and the various software's needed are given a thought in this section.

3.1 Technical Requirements

The Feature Extraction and comparison process in the Forgery Detection System may require a printer for obtaining the hardcopy of the output. Since a dedicated printer is not required, there is no direct interface to it.

With regard to software requirements, the System needs Windows 9x or higher for its implementation because, the system works with Windows bitmap (BMP) image formats.

3.2 Description of the various tools used

System Design ----------UML(Unified Modeling Language)

Code----------C++ version 3.0

Platform----------Windows 98

Database----------MS Access

UML (Unified Modeling Language) from the Rational Corporation, which is a very powerful designing tool was used in order to derive the data flow, control flow, the dependencies between various modules and their associations with one another. The rational rose software was used in designing the classes and use case diagrams.

The software was coded using C++, thus utilizing to the language's Object Oriented Features, making use of a VC++ compiler. Tables are created using MS Access for storing the authors name, signature segments as well as the feature values obtained.

4. System Design

This phase of the software development cycle plays a prominent role to achieve a visual illustration of the system. Based on the system requirements gathered, all the processes are identified and are given a detailed view in this phase of system development.

4.1 Overall Architecture

The Feature Extraction and Comparison system works by horizontally splitting the bitmap images into segments and analyzing each segment further. The input BMP images are those that have already been zoomed and clipped such that they are of the same size and devoid of any background.

The segmentation process is repeated for all the specimens of the genuine signature and also for the single test signature. The segments of each signature are stored in linked lists. After this process is over, each and every segment of each signature is read through so as to note the various spacing of the stokes in the signature. These are considered as distinct features and the values are recorded .In the end, the average value is calculated for each feature of each segment of each genuine signature.

Similar values are calculated for the test signature segments and these are compared to the values of the original signature.

Since it has been found that the signatures of the same person tend to vary slightly every time he signs, this aspect is taken into consideration and a tolerance factor has been included for each value.

Thus, it is checked whether the test signature values fall within the permitted tolerance. Finally, based on the number of matches, the decision is taken as to whether the signature is forged or not.

Temporal information used in online verification is not available offline and the details like the relative height, spacing etc are to be used. The offline problem is approached by establishing a local correspondence between a genuine and a questioned signature.

The questioned signature is segmented into consecutive horizontal stroke segments that are matched to the stroke segments of the model (genuine signature) Since more than one sample of the genuine signature is recommended for effective judgment an average of the features extracted is considered for matching with that of the test signature

Setting of tolerance is considered an important step of the process. This is because; it is believed that there can be differences in the signatures of the same person. So, a tolerance value has to be set for each and every feature. The setting of the tolerance value is done after extensive testing with signature samples. The value for which maximum accuracy is obtained is set as the tolerance levels. These are then stored permanently in the system.

All the images dealt by the system are of the Windows Bitmap file format. This is the standard file format used by the Microsoft Windows. These files are relatively easy to use and usually they are not compressed.

Among signature 2 types of variability is observed:

Intra-class variability:

Intra-class variability is the variation among genuine signatures. It is due to this factor that 'n' numbers of signatures of the author are considered for feature extraction, and an average of the value is taken. Also tolerance levels are fixed to overcome this problem.

Inter-class variability:

Inter-class variability is the variation between signatures of two different persons. It is believed that a person's signature contains features that are unique to the individual. This is the reason why signatures are still the most common way of authenticating ones identity.

The system works under the assumption that vertical or horizontal displacements or a given signature will still preserve its characteristics.

4.2 Input and Output design

The inputs to the system are two linked lists which contain the genuine signatures and the test signature. These have already been clipped and zoomed to the same resolution.

The system, after arriving at a conclusion, lists the same using images and the justifications for the decisions taken.

Output Screen

Forgery Detection for the signature of (author name):

The questioned signature is found to be ***(forged/genuine)*** *with percent probability.*

(Bitmap image of the original signature)	(Bitmap image of the questioned Signature)

(The calculations are based on n samples of original signature)

The system is also capable of generating reports about the details of the users who are eligible to log on as well as the details of data stored in the Reference Table.

Reference Report

Author Name: Number Of Samples:

Date Of Birth: Date Of Last Access:

Segment Number	Feature1	Feature2	...	Feature 6

4.3 Database Design

The reference signatures of a person when input for the first time gets stored in the Reference Table and the Feature Table of the database after it has been processed and the features extracted. For future verifications of the same signature, the calculated values are directly fetched from the database.

Fields of the Reference Table:

1) Author Name : The name of the author
2) Date of birth : Date of birth of the Author
3) Signature : Bitmap image of any one Signature
4) Total number Of Inputs : No of signatures considered

Composite Primary Key

Fields of the Feature Table:

1) Author Name : The name of the author
2) Date of birth : Date of birth of the Author
3) Segment number: The i th segment of the Signature
4) Last Access Date: Last used date
5) Feature no_1 : The value for the first Feature
n) Feature no_ 6 : The value of the 6 th feature

Composite Foreign

Author name and his date of birth acts as the primary key to uniquely identify a record of the Reference table. These become the foreign key for the Feature table where the features of each signature of the author get stored

The Author name and his date of birth are captured at the beginning of the process and stored as a text file that is then used to search the tables for retrieving the data.

4.4 Process Design

The clipped, zoomed images of the signatures are first segmented horizontally. For each and every segment of the signature, certain predefined measurements are taken and the values are recorded. The average value is found and this gets compared to that of the measured values of the test signature.

Tolerance levels are set for each value and if the parameters of the questioned signature falls with in the permitted range, the result is taken as TRUE, else, FALSE. The result of the comparison is stored in the appropriate data structures and the values are taken for decision-making.

The results are shown displayed to the user with the appropriate justifications.

4.5 Supporting Diagrams

These are the diagrams, which are designed in order to facilitate an easy understanding of the system. Given below are some brief narrations about the methodologies followed for the visual depiction of the system.

4.5.1 Primary Process Control Flow Diagram

In order to present a picture on how the control flows through the system, the primary process control flow is given below.

The flow of control for the entire signature verification system is given a thought with this visual illustration. **Refer Figure (1), Page No (19).**

All the other diagrams used can be understood with ease with the following details provided.

4.5.2 DFD's & System Views

DFD (Data Flow Diagrams)

As the name indicates, data flow diagrams otherwise known as DFD's are the means of representing the flow of control among the system constituents and represent three levels as given below.

Context Level:

The context level is also known as the level 0 DFD which will give a general outline sketch of the system. **Refer Figure (2), Page No**

Level 1 DFD:

The Level 1 data flow diagrams will give a representation of the overall skeleton of the system and its constituents. **Refer Figure (3), Page No (21).**

Level 2 DFD:

These provide the control flow among the various constituents considering each one of them individually. **Refer Figures (4,5,6), Page No (23).**

System Views

Use Case View:

This view makes use of the use case diagrams and is written at the analysis level. The use case diagrams make use of **Actors & Use Case.**

Actors, which come outside the boundary of the problem domain. For example we can consider a user, database from another system. Use Cases, are used to capture the functions involved. **Refer Figures (7,8), Page No (26).**

Behavioral View:

Under this view the Sequence diagrams, Collaboration diagrams, Start Chart diagrams are used. These can be used at both at the analysis and at the design phases.

Sequence diagrams act as an input for the class diagrams or vice versa. They are written at the analysis level and show he interaction between the various modules These are two dimensional diagrams since they grow horizontally and vertically. The lifeline goes on growing with the increasing interaction between the objects. **Refer Figure (9), Page No (28).**

Structural View:

This serves as a view to chart out the elements of the system and makes use of the Class diagrams. These are drawn at the design level and are very rarely used at the analysis level also for identifying the ALC's (Analysis Level Classes), from where we identify the good classes which are taken for designing.

Class Diagrams represent the various classes associated and relate their association ship as uni-directional or bi-directional. **Refer Figure (10), Page No (30).**

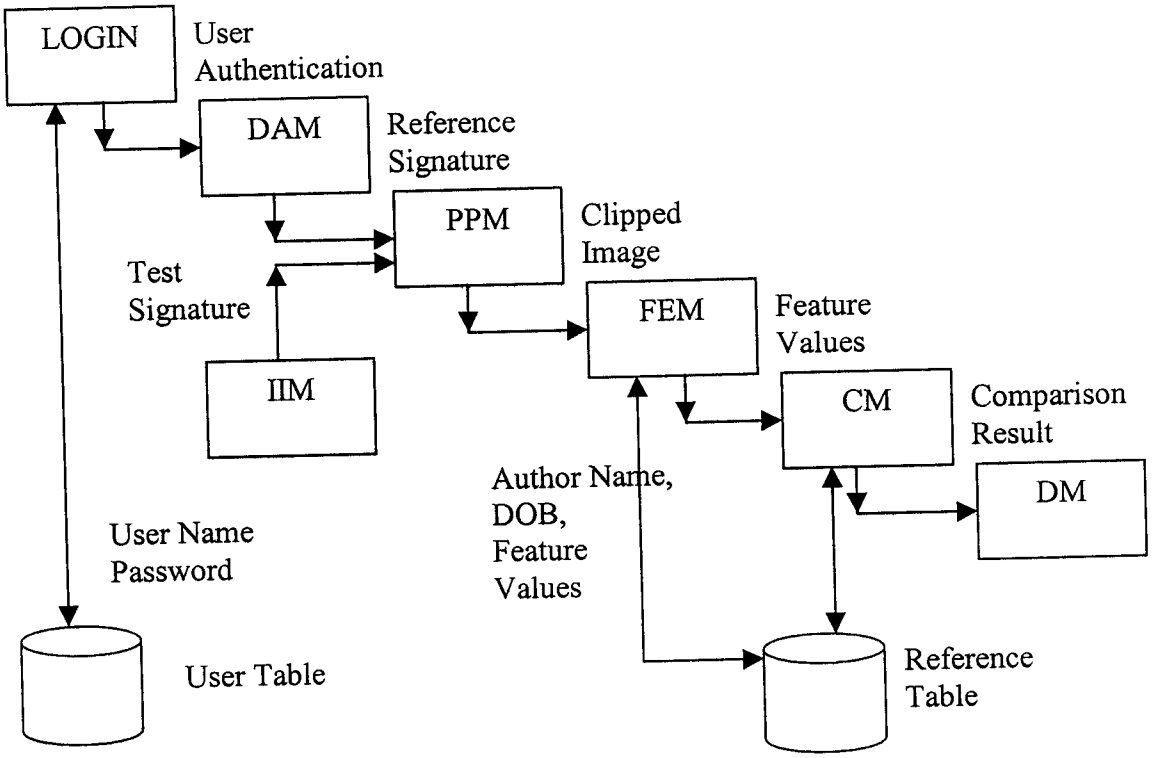
Implementation View:

When it comes to the representation of files we describe the files used and the relationship among them. Here we use the component diagrams.

Environmental View:

This is the view which narrates about the client environment where the developers deploy the engineered product and this can be used at any phase of the software development life cycle.

PRIMARY PROCESS CONTROL FLOW



DAM	Data Acquisition Module
PPM	Pre-processing Module
FEM	Feature Extraction Module
CM	Comparison Module
DM	Decision Module
IIM	Identification Input Module

Figure (1)

Data Flow Diagram

CONTEXT LEVEL DFD

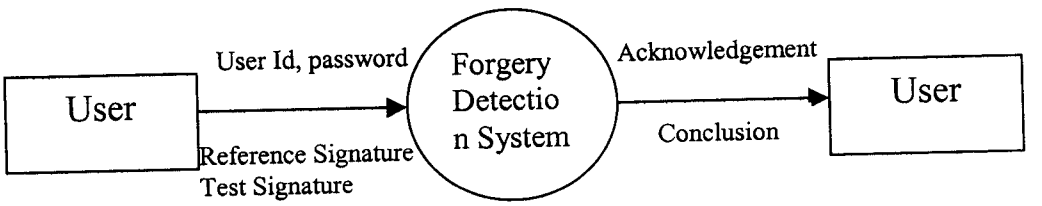


Figure (2)

LEVEL 1

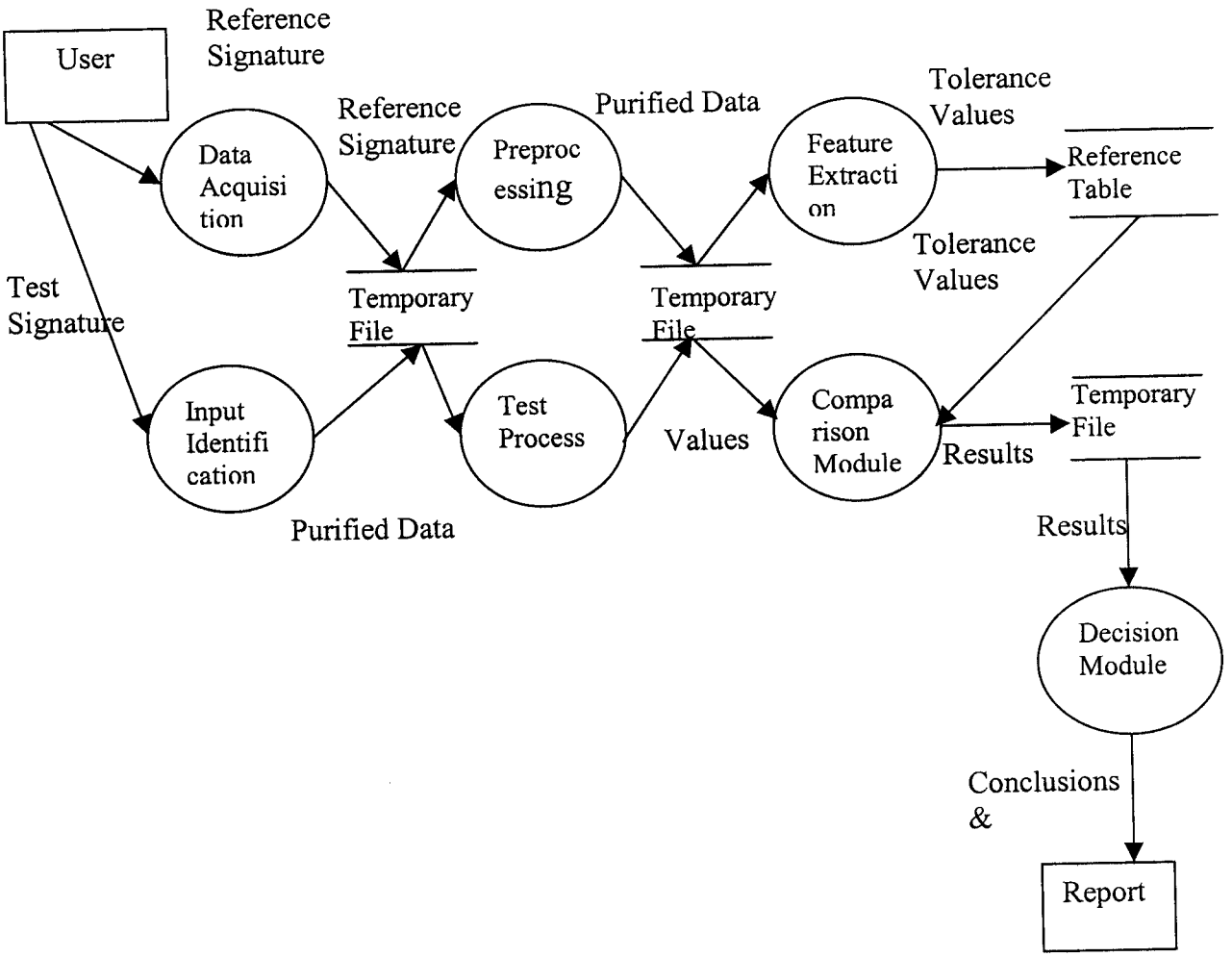


Figure (3)

LEVEL 2-DFD

PREPROCESSING MODULE

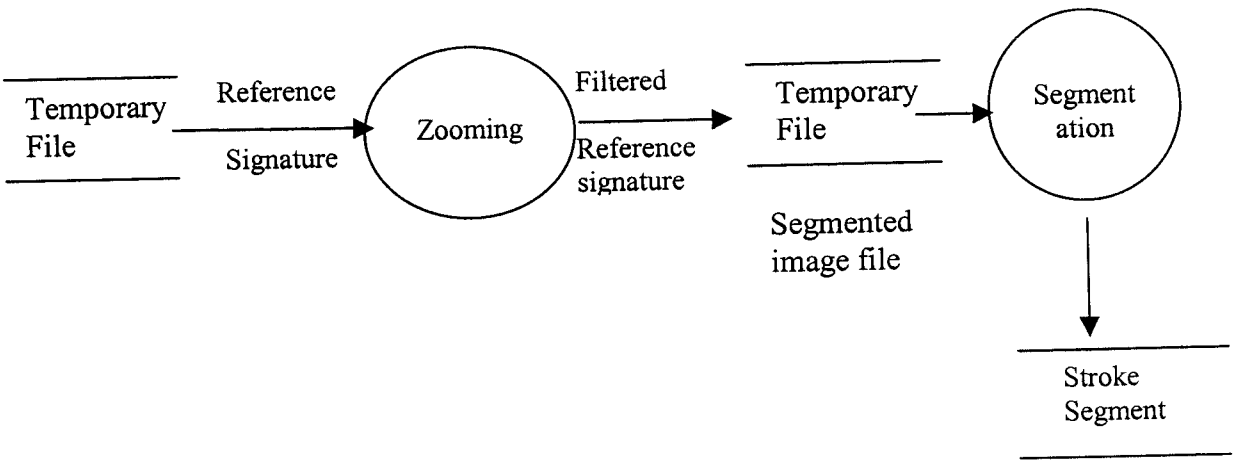


Figure (4)

LEVEL 2 DFD

FEATURE EXTRACTION MODULE

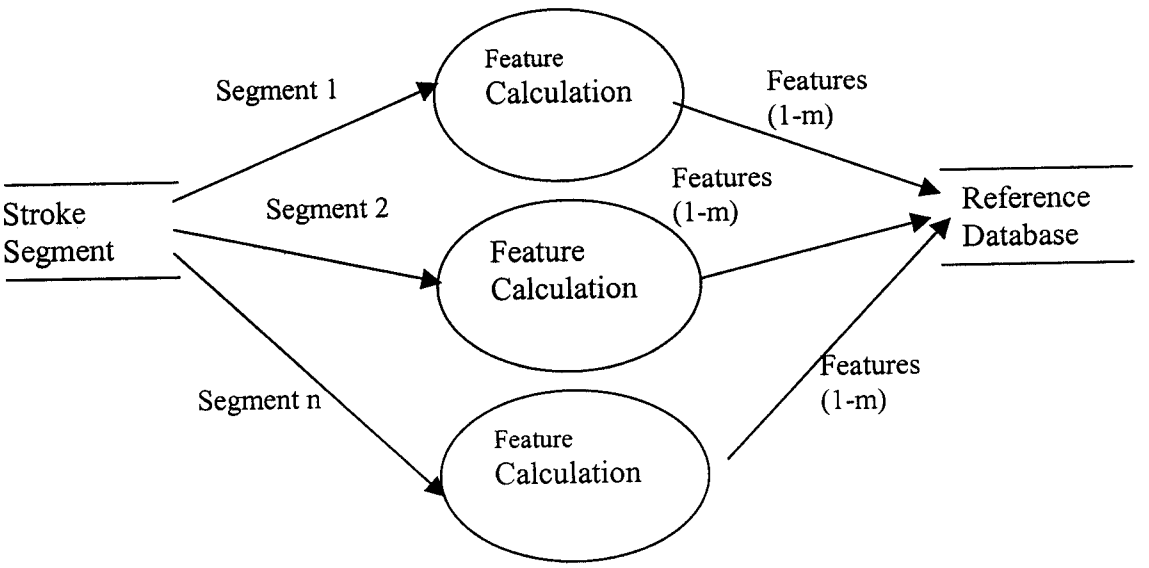


Figure (5)

LEVEL 2 DFD

COMPARISON MODULE

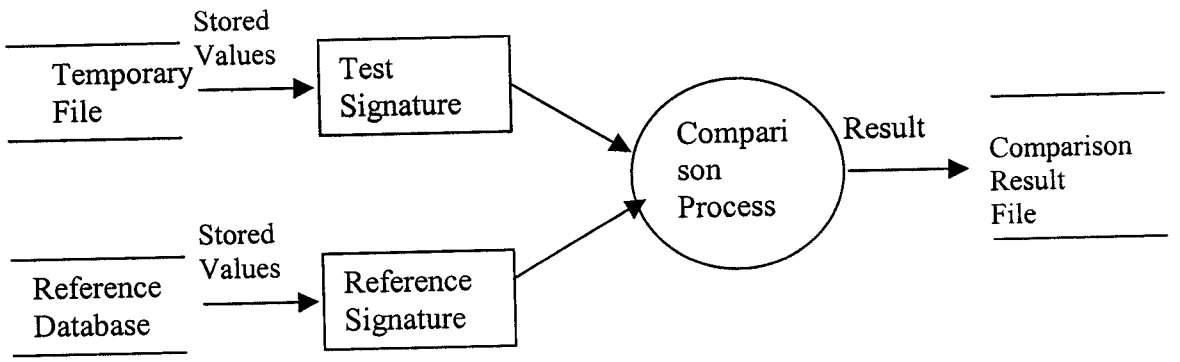


Figure (6)

LEVEL 2 DFD

DECISION MODULE

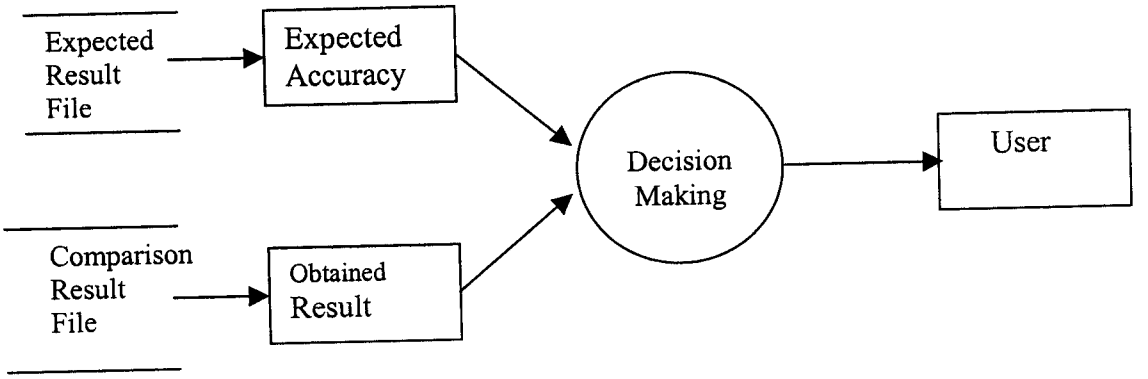


Figure (6)

USE CASE DIAGRAMS
(USING UML NOTATIONS)

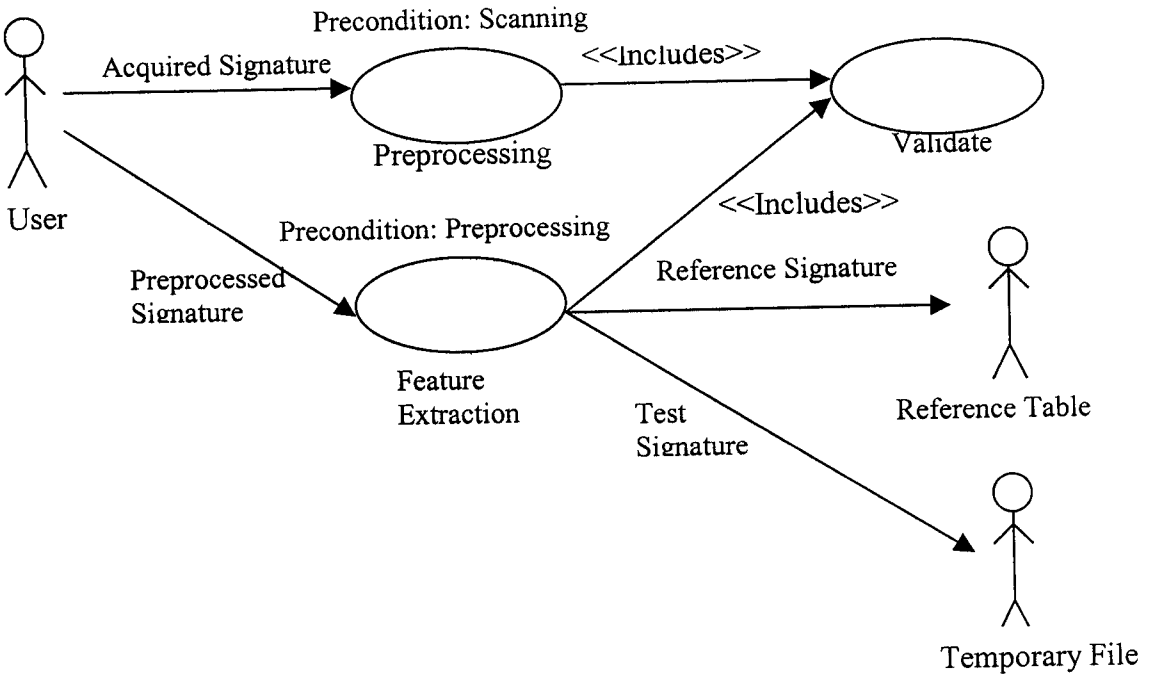


Figure (7)

USE CASE DIAGRAMS
(USING UML NOTATIONS)

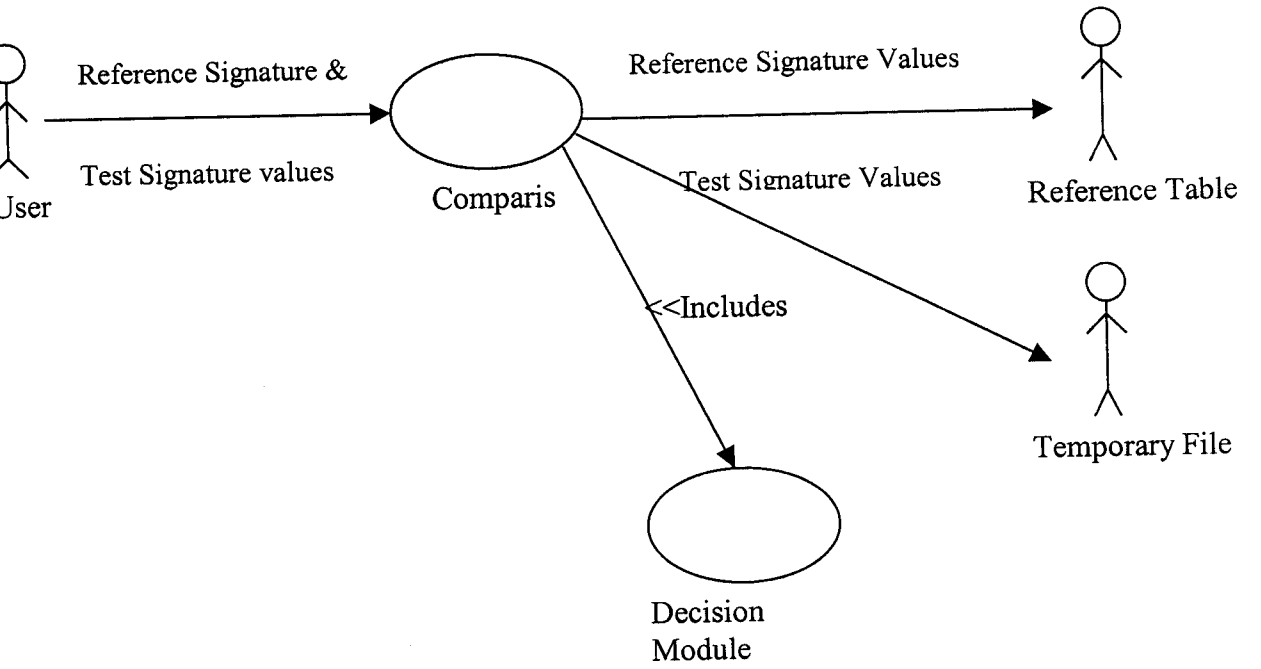


Figure (8)

SEQUENCE DIAGRAM

(For Object Oriented Analysis and Design)

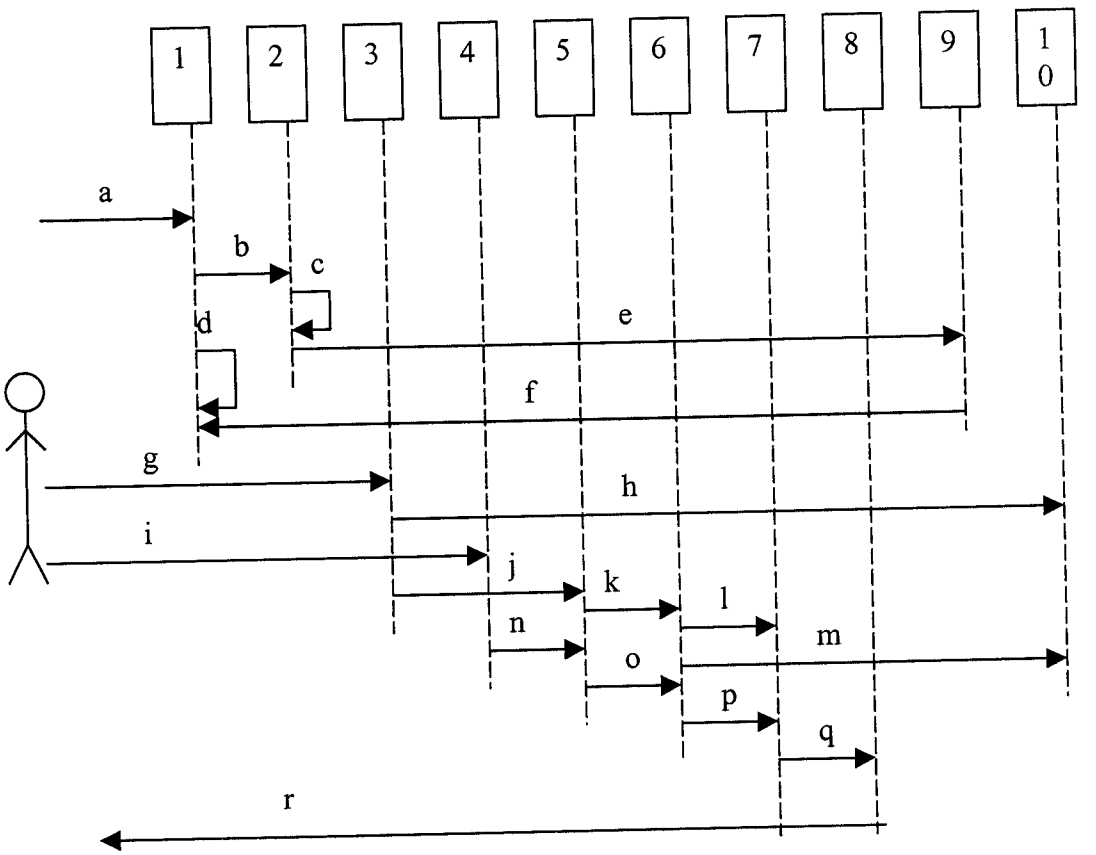


Figure (9)

1.Login Form Object

2.Connection Object

3.Reference Signature Object

4.Test Signature Object

5.Preprocessing Object

6.Feature Extraction Object

7.Comparison Object

8.Decision Object

9.AuthenticationObject

10.ReferenceRecord Object

a) User id, Password

b) Connect

c) Self-Message

d) self messaging

e) Authenticating the user

f) Acknowledgement

g) Reference signature (RS)

h) Copy of RS

i) Test Signature (TS)

j) Reference Signature

k) RS for Feature Extraction

l) Tolerance Values of RS

m) Store in Reference Database

n) Test Signature

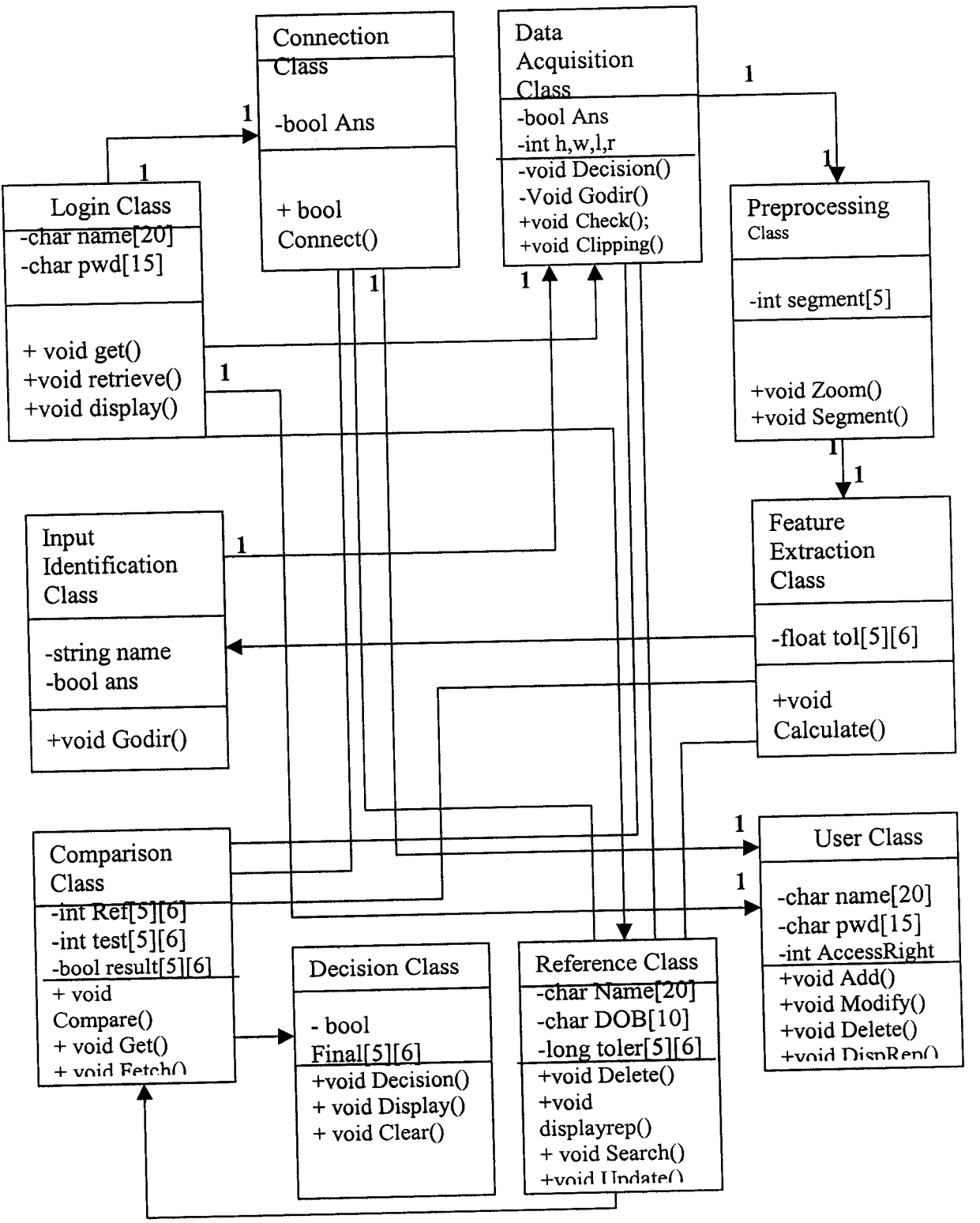
o) Test Signature for Feature Extraction

p) Calculated Values for TS

q) Compared Values

r) Result to the user

CLASS DIAGRAM (For Object Oriented Analysis and Design)



5. System Implementation & Testing

This is the phase of system development where the software is implemented and is tested for defects. Described below are the program modules of the entire system.

5.1 Program Modules

The process of analyzing the signatures in order to arrive at a conclusion about its genuineness is divided into various modules for clarity of functions and to reduce complexity, which are as follows:

- **Feature Extraction Module**
- **Comparison Module**
- **Decision Module**
- **Reference Module**

5.2 Description of Modules

After the Signatures are read into the System, be it a Reference Signature or a Test Signature, they are to be preprocessed before the can be send in for Feature Extraction & Comparison. This includes extracting every signature, zooming it to a predefined resolution and clipping the background such that the signature fits exactly into the bitmap image.

Feature extraction

The clipped image is horizontally split into two parts and each is stored as a separate node in a linked list (fig 5.2.a). This process is repeated for each signature acquired. Thus, each signature gets split into two horizontal segments (fig 5.2.b)

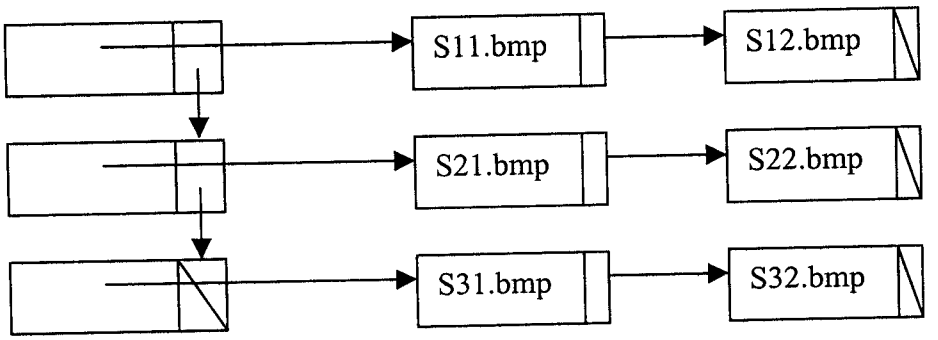
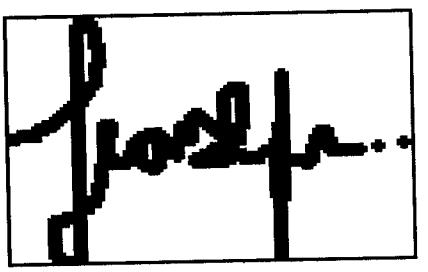
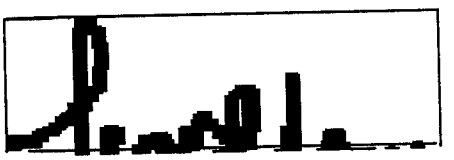
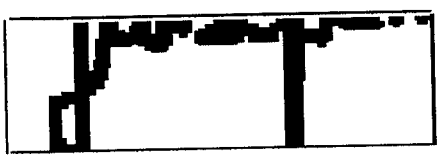


Fig 5.2.a : Segments of signature stored in linked lists



Original image





Segmented image

Each of the segmented images are then analyzed in order to extract certain parameters which are considered as features that are later used for comparison of signatures.

Here we are considering six features, which are measured as follows:

Feature 1 (F1):

The horizontal distance between the top leftmost corner of the image to the first pixel(t1) towards the right.

Feature 2 (F2):

The horizontal distance from the pixel(t1) to the end of the line or till another pixel(t2) is found towards the right.

Feature 3 (F3):

The vertical distance from the top leftmost corner of the image down to the first pixel(l1) found along the border is taken. Incase no pixel has been found the vertical distance till the end of the border is taken

Feature 4(F4):

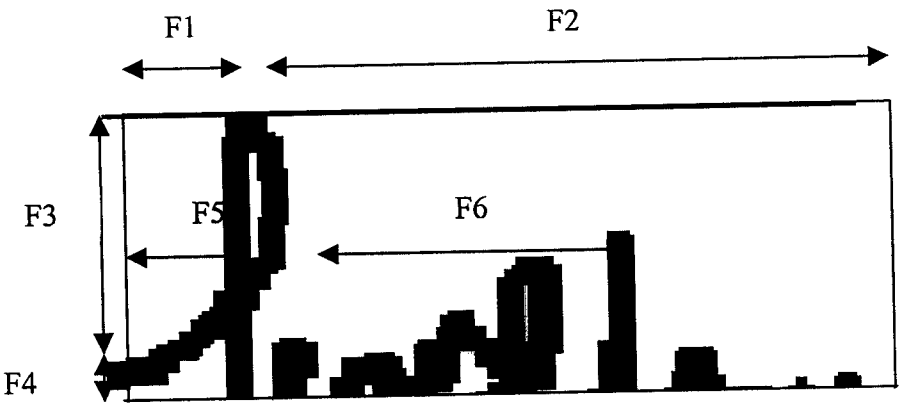
The vertical distance down from the pixel (l1) till the end of the segment in that direction, or, till another pixel (l2) is found.

Feature 5(F5):

The horizontal distance from the leftmost position at the middle of the file to the first pixel (m1) towards the right. Incase no pixel is found, the distance till the end of the line is considered.

Feature 6(F6):

The distance between the pixel (m1) to the next pixel (m2) towards its right, or till the end of the file. The above features can be diagrammatically shown as below:



These values are measured for each segment of each sample signature. After all the reference signature values are over, for the corresponding segment of each signature, the average value of the feature is found. These are stored in the feature table corresponding to the signature. Referring to the Author name and his/her date of birth does this, which have already been obtained from the user.

The failure to find the file containing the Author details will result in the system requesting the re-entry of the necessary details. Similar procedure is repeated with the test signature and the feature values are stored in temporary variables.

Comparison process

In order to determine the compatibility of the given signatures, the comparison process takes the stored values of the reference signatures for the database and performs a one to one comparison of the values of the features for the individual segments with that of test signature. The reference signature values are fetched from the Feature table.

A Tolerance level is set for each value taking into account the minor difference that can arise in the Signature of the same person. These values are fixed after repeated testing processes and are stored in the system.

The result of the Comparison Process is a linked list of Boolean values, which then acts as the input for the Decision Process.

Decision Process

Since the comparison is done on a one to one basis for the six features, the Boolean linked list has a total of 12 nodes, because each signature is split into two horizontal segments.

In order to decide the genuineness of the signature, the linked list is considered in two separate parts, i.e., the first six nodes form the first part and the remaining form the next set.

They are evaluated separately to ensure more accuracy of judgment. For each set of six nodes, the decision values are calculated as follows:

Number of nodes with value as "TRUE"	**Allotted weightage**
All six	100
Five out of six	80
Four out of six	60
Three out of six	40
Two out of six	20
One out of six	10

After calculating the values for both sets of nodes, the average value of the two is taken. This is used in the final decision making as follows:

80% or above:

The questioned signature is genuine

60% to 80%:

The questioned signature is genuine, with a probability of 70%.

40% to 60%:

The test signature is a forged one with a probability of 70%.

Below 40%:

The questioned signature is a forged one.

Output Screen

Forgery Detection for the signature of Mr.Arun Venketesh

The questioned signature is found to be "genuine " !

True Sample:

Questioned Sample:

(The calculations were based on 4 samples of original signature)

5.3 Testing

Testing of the software ensures the extent of software reliability. This is the only phase in the software development life cycle (SDLC) which is destructive in nature, where all the steps are taken to make a program fail. This is the transitive phase in the SDLC process where the software is implemented and is tested for defects. The defects identified are documented and corrected, so that the software is coded to form an executable base and is now ready to be deployed.

5.3.1 Test Plans

The modules of the System are to undergo Unit Testing as and when they are completed. After the results are found to be satisfactory, the modules will be integrated to form the entire system, which will then be subject to Integration Testing.

Finally an Acceptance Testing has been proposed to find the validity of the System. Regression Tests will be conducted whenever a change has been, made in any of the modules to ensure that it hasn't affected the rest of the system.

5.4 Test cases, test data, expected results and obtained results.

Test Case	Test Data	Expected Result	Obtained Result
1	The test signature is the genuine signature	Conclusion "Not Forged"	True
2	Multiple occurrences of the same signature as the Reference Signature and a forged Signature for testing	Conclusion "Signature has been Forged"	True
3	Test signature is forged.	Conclusion "Forged"	True
4	Improper image format for Signatures	The files should be skipped	True
5	Reference Signatures vary much	Accurate Prediction	False
6	"Author.txt" is deleted	The system displays a warning and asks the user to re enter the values	True
7	Improper scanning of the signatures	Produce accurate result	False

5.5 The criteria on which the completion of the test was judged

When the obtained results of the testing was the same as that of the expected result of the test case the system was considered stable. Wherever values of the expected results varied from those obtained necessary steps were to be taken to correct the differences.

In the above table, the test cases for which the expected result do not match with that of the obtained result, fall into the general constraint category of the system and are referred to in the section 7.2.1

Unit testing of the individual modules, followed by Integration Testing of the entire system was done. Regression Testing was also done to ensure that the changes made to a module have not affected the other parts of the software.

5.6 Corrective Actions

Incase of mismatch between the obtained and the expected result, if the case has not been ruled out as a constraint faced by the system, necessary corrective actions were taken in terms of re-designing and re-coding to ensure the accuracy of the test results.

6. Special Notes

6.1 Alternative procedures

Alternative procedures depict the alternative flows taken by the program in case there occurs an exception.

For storing the Feature values in to the Feature table the system tries to read the "Author.txt" file, which has the Author name and the date of birth. If this file is found to be deleted, the system throws a warning saying "Author.txt" not found and prompt the user for re-entering the contents. Then it does a verification of the same by checking with the Reference table. If the name has not been found, the system alerts the user about the mismatch and exits out of the process clearing the contents of the directories in the process.

6.2 Excerpts from other related documents

6.2.1 General Constraints and Dependencies (from Software Requirements Specification version 1.2)

Some of the cases that can occur with accepted specimen:

Authentic:

The author's signature is very much like the suspected signature.

Imitated:

Performed by j who is not the author and his signature j has a good similarity with that of the author.

Degenerate :

One of the true samples is rejected because the author's signatures vary considerably.

False :

If performed by some j who is not the author of the reference signature and the test and the reference signatures are not alike

This system is incapable of measuring velocity & the hand pressure when an individual signs. The system works under the assumption that no two persons can sign exactly alike and that however hard one may try he cannot succeed in making an exact replica of a genuine signature.

6.2.2 Design Constraints (Design Document Version 1.2)

The proposed System, being an Offline one, doesn't take into consideration the dynamic features of the signature like the hand pressure and velocity while signing.

7. Conclusion

The System was successfully designed and implemented. The accuracy level of the system is satisfactory but slightly less than the expected percentage. The expected accuracy was 90%, where as the System achieved 75% accuracy.

Since the system comes as an initial attempt by the Company at authenticating signatures and since further upgradations have been proposed, the results produced by the system are concluded to be satisfactory.

8. Further Enhancements

In order to increase the accuracy of the system, more features can be added to the existing code, like vertical division of the segments, checking for more features like intensity variations etc thus, increasing the reliability of the judgement. The system can also be made to handle more bitmap file formats like GIF or JPEG than just BMP.

Also the system omits the features like the signing speed, and the hand pressure while signing etc as they are not elements of importance for offline verification.

However, offline verification techniques form the backbone of on-line verification Systems. Hence by adding further features, the System can also be used as a full-fledged Online Signature Verification system.

9. Reference documents

Existing software documentation:

The Software Requirements Specification

The project has a revised Software Requirements Specification (version no 1.2), which describes the objective and scope of the System. It also describes the data and control flow through the System, the expected Input and Output of the Software and the Performance bounds.

The Design Document

This documents lists in detail the overall structure of the Software in general, as well as the structure of the modules. Supporting diagrams have been included which depicts the dependencies between the various modules of the software.

Technical reference

G.Rigoli, A.Kosmala, "A Systematic Comparison Between on-line and off-line Methods for Signature Verification with Hidden Markov Models", Information gathered from "Automatic Signature Verification", 2001.

N.Mohandrishnan, W.Lee, and M.Paulik, "A performance Evaluation of a New Signature Verification Algorithm", Information gathered from a published paper on "A Modern Approach to Signature Authentication", 2001.

T.Sebastian, P.Klein, B.Kimia ,"On Aligning Curves", McGraw Hill, 1992.

Will... Harington, "Computer Graphics", Prentice Hall of India, 1994.

Appendices

Keywords

- Authentication: Verification & conformation of the genuineness of signature.
- Bitmaps: An image format wherein the picture is stored as matrices of squares know as pixels
- Forgery: A deliberate attempt to counterfeit the identity of the person, by replicating his signature.
- Forgery detection: Identification of forgeries.
- Offline system: A system relying on the static features of the signature.
- Online system: Enhanced version of the offline system where some dynamic features such as velocity, hand pressure while signing etc., are considered.
- Reference Signature: Genuine signature given as input.
- Signature: The way a person writes his name for authentication purpose.
- Simple forgery: Forged signature with notable difference.

- Substitution forgery: Signing a totally different name instead of the original.

- Skilled forgery: Expert forgeries.

- Stroked Segment: One of the rows into which the signature gets split for analysis.

- Test Signature: The signature suspected to be forged.

- Tolerance: Acceptable degree of differences between two signatures of the same person.