How effective are Radial Basis Function Neural Networks for Offline Handwritten Signature Verification?

George Azzopardi
Student Number U/01/0316557

University of London
May 2006

Supervisor
Dr. Ing. Kenneth Camilleri

Submitted as part of the requirements for the award of the Degree in Computing and Information Systems of the University of London
A copy of this form (or a typed or computer-generated version) must be completed by each student and attached to each project report that is submitted to the University.

The report must be submitted in time to be received by the University before 15 May in the year of the examination.

Full name: Mr. George Azzopardi
(as it appears on your Registration Form)

Student number: U/01/0316557

Project title: How effective are Radial Basis Function Neural Networks for Offline Handwritten Signature Verification?

DECLARATION

I declare that:

- I understand what is meant by plagiarism
- I understand the implications of plagiarism
- This project report is all my own work and I have acknowledged any use of the published or unpublished works of other people.

Signature......................................................... Date ............................................
To my dear parents and to
my lovely fiancée Charmaine Borg
Acknowledgements

I would like to express my gratitude to all those who gave me the possibility to complete this thesis.

First of all, I would like to thank my supervisor, Dr. Kenneth Camilleri, for helping me sorting out my ideas and for his constant assistance throughout this project. His incredibly broad knowledge-base and his systematic approach helped me to understand the complexities of this study. I must say, that I have been honored to share his outstanding experience in the field of Pattern Recognition and also his excellent way in approaching complex problems.

I would like to thank Mr. Joseph Gaffiero for accepting me to interview him as part of this project. His dedication towards handwritten signature verification made me learn to love this area of research even more.

I would also like to express my gratitude to my project manager, Mr. Marco Scicluna, for approving the required dissertation leave and for some ideas on structuring the dissertation report.

I wish to express my appreciation to everyone who contributed to this project by providing signature samples for building the required signature database.

Special thanks go to my parents for their support and sympathetic help throughout the last years.

Lastly, but not less, I would like to express my appreciation to my girlfriend Charmaine Borg for her encouragement and strong support throughout the graduate studies. Her love and presence were the ingredients needed to let me focus on the studies especially in the last months.
Summary

The objective of this project was to investigate the effectiveness of totally radial basis function neural network (RBFNN) single-layer architecture for offline handwritten signature verification. An RBFNN, initialised by supervised clustering, was adopted for each author's signature samples.

RBFNNs are quite new in this domain, and are well-known for the robustness in eliminating outliers and for the relatively simple computations required to be trained. These were the main motivator factors that challenged the author of this project to investigate the effectiveness of RBFNNs in the field of offline handwritten signature verification.

A signature database was collected for the scope of this study as no international public database is available. Professional recommendations by J. Gaffiero who is a Maltese graphologist and personal recommendations by H. Baltzakis helped to acquire a signature database with as much intrapersonal variations as possible.

Three groups of signature features namely global, grid and texture features were used to evaluate the system in different scenarios. The grid and texture features were extracted from a superimposed grid of $812 \times 8$ segments, where a vector quantisation (VQ) technique was required to cluster the respective column feature vectors. In this case, two VQ approaches were investigated; an adaptively sized codebook VQ and a fixed size codebook VQ of 50 codewords.

The entire system was extensively tested with random signature forgeries and the high recognition rates obtained show that the proposed architecture is effective in this field. Surprisingly, the fixed size codebook VQ performed at least twice as good as the adaptively sized codebook VQ. In fact, the best results where obtained when global and grid features where combined producing a feature vector of 592 elements. In this case a Mean Error Rate (MER) of 2.04% with a False Rejection Rate (FRR) of 1.58% and a False Acceptance Rate (FAR) of 2.5% were achieved. The mentioned results were found to rank better than some other published studies.
# List of Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AHSVS</td>
<td>Automatic Handwritten Signature Verification System</td>
</tr>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>BMP</td>
<td>Bitmap</td>
</tr>
<tr>
<td>DRT</td>
<td>Discrete Radon Transform</td>
</tr>
<tr>
<td>EER</td>
<td>Equal Error Rate</td>
</tr>
<tr>
<td>ER</td>
<td>Entity Relationship</td>
</tr>
<tr>
<td>ESC</td>
<td>Extended Shadow Code</td>
</tr>
<tr>
<td>FAR</td>
<td>False Acceptance Rate</td>
</tr>
<tr>
<td>FRR</td>
<td>False Rejection Rate</td>
</tr>
<tr>
<td>HMM</td>
<td>Hidden Markov Model</td>
</tr>
<tr>
<td>HSV</td>
<td>Handwritten Signature Verification</td>
</tr>
<tr>
<td>IDE</td>
<td>Integrated Development Environment</td>
</tr>
<tr>
<td>JAI</td>
<td>Java Advanced Imaging</td>
</tr>
<tr>
<td>JAMA</td>
<td>Java Matrix</td>
</tr>
<tr>
<td>LOO</td>
<td>Leave-One-Out</td>
</tr>
<tr>
<td>MD</td>
<td>Minimum Distance</td>
</tr>
<tr>
<td>MER</td>
<td>Mean Error Rate</td>
</tr>
<tr>
<td>MLP</td>
<td>Multi-Layer Perceptron</td>
</tr>
<tr>
<td>NN</td>
<td>Neural Network</td>
</tr>
<tr>
<td>OCON</td>
<td>One-Class-One-Network</td>
</tr>
<tr>
<td>OHSV</td>
<td>Offline Handwritten Signature Verification</td>
</tr>
<tr>
<td>PDA</td>
<td>Personal Digital Assistant</td>
</tr>
<tr>
<td>PDF</td>
<td>Probability Density Function</td>
</tr>
<tr>
<td>PIN</td>
<td>Personal Identification Number</td>
</tr>
<tr>
<td>RBF</td>
<td>Radial Basis Function</td>
</tr>
<tr>
<td>RBFNN</td>
<td>Radial Basis Function Neural Network</td>
</tr>
<tr>
<td>RDBMS</td>
<td>Relational Database Management System</td>
</tr>
<tr>
<td>ROC</td>
<td>Receiver Operating Characteristic</td>
</tr>
<tr>
<td>TER</td>
<td>Total Error Rate</td>
</tr>
<tr>
<td>TATA</td>
<td>Train All Test All</td>
</tr>
<tr>
<td>TNTA</td>
<td>Train Non-Frame Test All</td>
</tr>
<tr>
<td>TNTN</td>
<td>Train Non-Frame Test Non-Frame</td>
</tr>
<tr>
<td>VQ</td>
<td>Vector Quantization</td>
</tr>
</tbody>
</table>
Table of Contents

Acknowledgements ........................................................................................................... ii
Summary........................................................................................................................... iii
List of Abbreviations ......................................................................................................... iv
Table of Contents ............................................................................................................ v
List of Figures ................................................................................................................ viii
List of Tables .................................................................................................................. x

Chapter 1 Introduction................................................................................................. 1
1.1 Overview of Handwritten Signatures ................................................................. 2
1.2 Nature of a Human Signature ............................................................................ 3
1.3 Signature Verification Techniques - Online and Offline ................................... 4
   1.3.1 Offline Signature Verification Methods ................................................. 4
   1.3.2 Online Signature Verification Methods ................................................. 5
1.4 Types of Signature Forgeries ............................................................................. 6
1.5 Handwritten Signature Verification (HSV) Applications ................................ 7
1.6 Overview of the Report...................................................................................... 8

Chapter 2 Literature Review.................................................................................... 9
2.1 Research Methods............................................................................................... 10
2.2 Introduction......................................................................................................... 11
2.3 Data Acquisition ............................................................................................... 11
2.4 Pre-processing ................................................................................................... 11
2.5 Feature Extraction and Selection ...................................................................... 12
   2.5.1 Global Features ................................................................................... 13
   2.5.2 Local Features ..................................................................................... 14
   2.5.3 Pseudo-dynamic Features.................................................................... 15
2.6 Published Results............................................................................................... 15
2.7 Comparison Process........................................................................................... 16
   2.7.1 Simple Distances .................................................................................. 16
   2.7.2 Hidden Markov Models (HMM) .......................................................... 17
   2.7.3 Neural Networks .................................................................................. 18
2.8 OHSV using RBF Neural Networks ................................................................ 19
2.9 Terms of Reference........................................................................................... 21
   2.9.1 Changes to original Project Objectives............................................... 21
   2.9.2 Project Motivation ............................................................................... 22
   2.9.3 Project Plan ......................................................................................... 22
2.10 Conclusion ......................................................................................................... 22

Chapter 3 Methodology .......................................................................................... 24
3.1 Data Acquisition ............................................................................................... 25
   3.1.1 Observations ........................................................................................ 26
   3.1.2 Boundary to this Work ........................................................................ 26
   3.1.3 Summary .............................................................................................. 27
3.2 Pre-processing.................................................................................................... 27
   3.2.1 Data area cropping .............................................................................. 27
   3.2.2 Width Normalization .......................................................................... 28
   3.2.3 Binarization ......................................................................................... 28
6.2 Limitations ................................................................. 84
6.3 Contributions ............................................................ 84
6.4 Future Work and Recommendations .............................. 85

Appendices .............................................................................. 86
A. Project Description Form .................................................. 87
B. Data Protection Commissioner Correspondence .................. 89
C. Data Acquisition Sample ................................................... 91
D. Transcript of interview with Maltese Graphologist ............... 95
E. Entity Relationship (ER) Diagram ....................................... 100
F. Summary of the Main Implemented Algorithms .................. 101
G. Program Implementation .................................................. 102
H. CD Contents ..................................................................... 103

Bibliography ............................................................................. 105

Evaluation ................................................................................ 110
List of Figures

Figure 1 - Biometric Market Report (International Biometric Group) estimates the revenues of various biometrics in 2006 in terms of market share [33]........................................2

Figure 2 - Difficulty of offline signature verification - Taken from our collected signature database.................................................................5

Figure 3 - Types of forgery (a) genuine signature; (b) random signature; (c) simulated simple forgery; (d) simulated skilled forgery - Taken from our collected signature database..........................................................6

Figure 4 - Forty skeletons of genuine signatures from three writers with centered and superimposed in the image plane - Taken from our collected signature database 14

Figure 5 - (a) Genuine signature; and forgeries with (b) Pressure areas; (c) Stroke curvature; (d) Stroke regularity – Taken from our collected signature database..............................15

Figure 6 - Displacement function of the authentic g and questionable f signatures..............17

Figure 7 - The borderlines used to delimit the area of acceptance and rejection in the validation process .................................................................17

Figure 8 - Structure of two-stage neural network [7]..........................................................20

Figure 9 - Proposed Project Plan Figure 10 - Actual Project Plan ......................................23

Figure 11 - Signature proportionality affected by provided frames ....................................26

Figure 12 - Data Area Cropping; (a) Original image, (b) white spaces removed...............27

Figure 13 - Normalised and Binarised Signature Image.....................................................28

Figure 14 - Thinning Process: Smoothing Templates used in boundary pixel check [58] ......29

Figure 15 - Example of Thinned Signature ........................................................................29

Figure 16 - Image Rotation for finding Global and Local angles; (a) Original Image; (b) Rotation of -45°; (c) Rotation of -10°; (d) Rotation of 15° and (e) Rotation of 30° .................................................................................33

Figure 17 - Number of Edges Calculation Example ..........................................................33

Figure 18 - Cross Points Calculation Example .................................................................34

Figure 19 - Zoom in a Cross Point ....................................................................................34

Figure 20 - Labeling Connected Components ..................................................................35

Figure 21 - Updating the Labels of Connected Components ...........................................35

Figure 22 - Extra Departures Calculation Example ..........................................................36

Figure 23 - Closed Loops Example ..................................................................................36

Figure 24 - Pixel Density Example ..................................................................................37

Figure 25 - Pixel Distribution Example ............................................................................37

Figure 26 - Predominant Axial Slant Example .................................................................38

Figure 27 - Texture Co-Occurrence Matrix for a Binary Image .........................................39

Figure 28 - Texture Features Example ............................................................................39

Figure 29 - RBFNN Single Layer Architecture .................................................................40

Figure 30 - Training with Grid Features ..........................................................................45

Figure 31 - Graphical User Interface – Main Window .........................................................48

Figure 32 - Graphical User Interface - Search Author .......................................................49

Figure 33 - Testing Summary Excel Workbook Sample ..................................................51

Figure 34 - Results: Global Features - TNTN..................................................................54

Figure 35 - Results: Global Features - TNTA..................................................................55

Figure 36 - Results: Global Features - TATA.................................................................56

Figure 37 - Results: Grid Features - TNTN - Adaptively Sized VQ Codebook ...............57

Figure 38 - Results: Grid Features - TNTN - Fixed Size (50) VQ Codebook ..................57

Figure 39 - Results: Grid Features - TNTA - Adaptively Sized VQ Codebook ...............58

Figure 40 - Results: Grid Features - TNTA - Fixed Size (50) VQ Codebook .................58

Figure 41 - Results: Grid Features - TATA - Adaptively Sized VQ Codebook ...............59

Figure 42 - Results: Grid Features - TATA - Fixed Size (50) VQ Codebook .................59

Figure 43 - Results: Texture Features - TNTN - Adaptively Sized VQ Codebook ..........60

Figure 44 - Results: Texture Features - TNTN - Fixed Size (50) VQ Codebook .............60
Figure 45 - Results: Texture Features - TNTA - Adaptively Sized VQ Codebook
Figure 46 - Results: Texture Features - TNTA - Fixed Size (50) VQ Codebook
Figure 47 - Results: Texture Features - TATA - Adaptively Sized VQ Codebook
Figure 48 - Results: Texture Features - TATA - Fixed Size (50) VQ Codebook
Figure 49 - Results: Global & Grid Features - TNTN - Adaptively Sized VQ Codebook
Figure 50 - Results: Global & Grid Features - TNTN - Fixed Size (50) VQ Codebook
Figure 51 - Results: Global & Grid Features - TATA - Adaptively Sized VQ Codebook
Figure 52 - Results: Global & Grid Features - TATA - Fixed Size (50) VQ Codebook
Figure 53 - Results: Global & Grid Features - TATA - Adaptively Sized VQ Codebook
Figure 54 - Results: Global & Grid Features - TATA - Fixed Size (50) VQ Codebook
Figure 55 - Results: Global & Texture Features - TNTN - Adaptively Sized VQ Codebook
Figure 56 - Results: Global & Texture Features - TNTN - Fixed Size (50) VQ Codebook
Figure 57 - Results: Global & Texture Features - TATA - Adaptively Sized VQ Codebook
Figure 58 - Results: Global & Texture Features - TATA - Fixed Size (50) VQ Codebook
Figure 59 - Results: Global & Texture Features - TATA - Adaptively Sized VQ Codebook
Figure 60 - Results: Global & Texture Features - TATA - Fixed Size (50) VQ Codebook
Figure 61 - Results: Grid & Texture Features - TNTN - Adaptively Sized VQ Codebook
Figure 62 - Results: Grid & Texture Features - TNTN - Fixed Size (50) VQ Codebook
Figure 63 - Results: Grid & Texture Features - TNTA - Adaptively Sized VQ Codebook
Figure 64 - Results: Grid & Texture Features - TNTA - Fixed Size (50) VQ Codebook
Figure 65 - Results: Grid & Texture Features - TATA - Adaptively Sized VQ Codebook
Figure 66 - Results: Grid & Texture Features - TATA - Fixed Size (50) VQ Codebook
Figure 67 - Results: Global, Grid & Texture Features - TNTN - Adaptively Sized VQ Codebook
Figure 68 - Results: Global, Grid & Texture Features - TNTN - Fixed Size (50) VQ Codebook
Figure 69 - Results: Global, Grid and Texture Features - TNTA - Adaptively Sized VQ Codebook
Figure 70 - Results: Global, Grid & Texture Features - TNTA - Fixed Size (50) VQ Codebook
Figure 71 - Results: Global, Grid & Texture Features - TATA - Adaptively Sized VQ Codebook
Figure 72 - Results: Global, Grid & Texture Features - TATA - Fixed Size (50) VQ Codebook
Figure 73 - Average ROC: All 7 features - TATA - Adaptively Sized VQ Codebook
Figure 74 - Average ROC: All 7 features - TATA - Fixed Size (50) VQ Codebook
Figure 75 - Entity Relationship (ER) Diagram
List of Tables

Table 1 - Experimental results, mean total error rate $E_t$ in % (Rejection Rate $R_t$) [66]........16
Table 2 - Results obtained by Baltzakis and Papamarkos [7]...............................................20
Table 3 - Predominant Axial Slant Templates..................................................................38
Table 4 - Software Tools Used ..........................................................................................47
Table 5 - Performance Summary Results Template ..........................................................52
Table 6 - Results: Global Features - TNTN.....................................................................54
Table 7 - Results: Global Features - TATA.......................................................................55
Table 8 - Results: Global Feature - TATA .........................................................................56
Table 9 - Results: Grid Features - TNTN ..........................................................................57
Table 10 - Results: Grid Features - TATA ........................................................................58
Table 11 - Results: Grid Features - TATA ........................................................................59
Table 12 - Results: Texture Features - TNTN .................................................................60
Table 13 - Results: Texture Features - TATA .................................................................61
Table 14 - Results: Texture Features - TATA .................................................................62
Table 15 - Results: Global & Grid Features - TNTN .......................................................63
Table 16 - Results: Global & Grid Features - TATA .......................................................64
Table 17 - Results: Global & Grid Features - TATA .......................................................65
Table 18 - Results: Global & Texture Features - TNTN ..................................................66
Table 19 - Results: Global & Texture Features - TATA ..................................................67
Table 20 - Results: Global & Texture Features - TATA ..................................................68
Table 21 - Results: Grid & Texture Features - TNTN ......................................................69
Table 22 - Results: Grid & Texture Features - TATA ......................................................70
Table 23 - Results: Grid & Texture Features - TATA ......................................................71
Table 24 - Results: Global, Grid & Texture Features - TNTN ........................................72
Table 25 - Results: Global, Grid & Texture Features - TATA ........................................73
Table 26 - Results: Global, Grid & Texture Features - TATA ........................................74
Table 27 - Hardware specifications ..................................................................................80
Table 28 - Summary of the main implemented algorithms ............................................101
Chapter 1  Introduction
1.1 Overview of Handwritten Signatures

Personal verification and identification is an actively growing area of research and development. There are different biometrics methods used to authenticate the identity of an individual, which can be categorised in physiological (face, iris, fingerprint, odour) and behavioural traits (signature, voice) among other characteristics. Biometrics is defined as the science and technology of interactively measuring and statistically analysing biological data, in particular, taken from human beings. Biometric authentication is becoming more suitable and convenient as a more trustable alternative solution to traditional password based security systems. This is due to the fact that a biometric characteristic is almost impossible to steal, copy, or even guess biometric properties [5].

The motivating force in this field is, above all, the increase in popularity of electronic commerce. For this reason, a lot of biometric applications are being introduced in the area of electronic commerce and electronic banking systems. Traditional authentication techniques (e.g. passwords, PIN numbers, smart cards, etc…) suffer from the management point of view. On the contrary, biometric authentication techniques are not easily transferable, are unique of every individual, and cannot be lost, stolen or broken, as they are natural to human beings [38].

The following criteria are used when applying a particular biometric [5]

1. Uniqueness - how unique is the biometric characteristic?
2. Ease of copying and stealing
3. Acceptability by the public; how acceptable is the biometric accepted by the public? For example handwritten signatures are widely accepted as they are used as a proof of authenticity in different fields.
4. Cost to implement the particular biometric data

The human hand, on its own, provides a number of physiological biometric features. The palm print, hand geometry, finger geometry and the vein pattern on the dorsum of the hand are the most frequently used. On the other hand, handwritten signatures are considered a behavioural biometric feature, and its acceptance is widespread socially and legally as a means of authentication [41]. The Biometrics Market and Industry Report 2006-2010 states that the use of hand and signatures based features are two of the eight leading biometric technologies and has together 10.5% of the world market in 2006 (See Figure 1):

![Figure 1 - Biometric Market Report (International Biometric Group) estimates the revenues of various biometrics in 2006 in terms of market share [33].](image-url)
A specific relevant component in handwriting biometrics is signature recognition. Signature recognition is focused on extracting writer-specific information, which makes the signature as unique as possible. Handwriting is considered a natural skill and can be used for different applications (See Section 1.5). Existing devices such as, PDA, Pocket PC, Tablet PC, or 3G mobile phones support handwriting capabilities [41].

Handwritten signatures goes back to the origin of the writing itself. Handwritten signature verification (HSV) systems is becoming more and more popular as are being considered superior to many other biometric authentication techniques mentioned above. These are reliable but much more expensive and intrusive and hence are generally accepted only in highly security sensitive situations where reliability is critical. Moreover, signature authenticity has already become a tradition in the western civilization [25]. This history of trust means that people are very willing to accept a signature based verification system [70]. Although HSV has the potential to gain popularity in the future, Miller [46] and Sherman [71] (cited by Gubta and McCabe [25]) both comment on the fact that this technique will be widely accepted only if it provides more reliability and robustness than the current products on the market.

1.2 Nature of a Human Signature

According to the American Heritage Dictionary a signature can be defined as “the name of a person written with his or her own hand; the act of signing one’s name” [2]. A second definition refers to the whole process of signing which implies the way the signature is made is part of the signature itself. This leads to the hypothesis that the characteristics of the process of signing (i.e. pen pressure, velocity, stroke, etc…) are unique to every individual [55] (cited by Kalenova [38]).

The first definition describes a signature as a static two-dimensional image, which does not contain any time-related information. On the other hand, the second definition is based on the dynamic features of the process of signing [55] (cited by Kalenova [38]). This subject is discussed further in detail in Section 1.3.

Handwritten signatures vary in many different forms and there is also a great variability even in signatures of people with different cultures [25]. For instance, people just write their name, others may use only their initials, or using signatures that are hardly related to their names and, as Brault and Plamondon in [13] (cited by Gubta and McCabe [25]) commented, some signatures may be quite complex, while others are simple and can be forged easily. A handwritten signature is considered as a natural art on its own.

Ruth Rostron [63] explains the graphology of a handwritten signature used to analyse and reveal the personality of an individual. She describes the variability in the signatures of different people and also of the same person, which may vary in a number of factors, including the individual’s mood at the moment. Graphology is a science in its own right and is used in different areas in modern societies.
In [25] Gubta points out that, some signature experts note that if two signatures of the same person were identical, they could be automatically considered as forgery by tracing. From a technical point of view, successive signatures of the same person will differ, both globally and locally and may also differ in scale and orientation. Notwithstanding these variations, it is said that even though the signatures might slightly differ, these will still have the same characteristics, such as the slant angle and the pressure, which will classify them as genuine signatures (Signature features are discussed in detail in Section 2.5).

It has also been suggested that human experts are very good in identifying forgeries but perhaps not so in verifying genuine signatures. For instance, in a detailed study, Herbst and Liu [29] cite references, state that signature experts managed to reject or classify as no-opinion as high as 25% genuine signatures, while accepting no forgeries. Untrained personnel accepted up to 50% forgeries.

1.3 Signature Verification Techniques - Online and Offline

Signature verification systems are categorised in the following two groups.

1. Offline methods (also referred to as static)
   - No information available at the time of signing the signature.
   - A scanned image of the signature is available

2. Online methods (also referred to as dynamic)
   - Time-related information in the form of p-dimensional function of time is available, where p represents the number of features of the signatures, such as the pressure pen, velocity and others [55] (cited by Kalenova [38]).

Although online methods have proved to be more accurate as they possess the dynamics of the signature as extra knowledge, offline methods are also essential in the areas where the customer is not present at the time of verification. That is, no knowledge is available describing the process of signing. For instance, verification of signature during payment processing of cheques could only be managed by an offline method as no online characteristics can be extracted.

1.3.1 Offline Signature Verification Methods

Offline signature verification was the first approach to be applied for solving the signature verification problem. It involves the discrimination of genuine and forged signatures on static images. Unlike online systems, offline systems have only the static image containing the signature as an input, without having any knowledge on the signing process. Some difficulties that may arise in offline systems are related to the scanning process (noise on the image) and to the signature acquisition process where different pen tips and widths can produce different shapes [5].

Figure 2 depicts an example where the four genuine signatures (on the left) are difficult to distinguish from the forged signature (on the right).
For this reason, offline methods were always involved in random and simple forgeries (unprofessional), whereas skilled forgeries are mainly tackled by online methods. Offline signature verification methods for detection of skilled forgeries, is still an open research question [5] (See Section 1.4).

### 1.3.2 Online Signature Verification Methods

On-line signature verification is based on the dynamic features of the process of signing. The fact that online verification contains more information, about the process of signing, the accuracy of the recognition is significantly higher than an offline method that does not have any kind of information of the signing process [39] (*cited* by Kalenova [38]).

However, the online method requires a special hardware to measure the dynamic characteristics of the signature process. A digitising tablet is used for this purpose, which mainly registers the trajectory and speed of the process together with pressure, pen tip position and other characteristics. The combination of these characteristics are said to be almost unique to every individual [38].

Furthermore, the dynamic signature verification process can be divided into two broad groups: functional and parametric. In the former case, the decision-making process is constructed on functions in which the input values constitute to the feature set measured by the equipment. However, in the latter approach, the parameters of the measured signal are considered as the feature sets [38].

The basic methodology for both methods is basically the same. The methodology usually involves data acquisition, pre-processing, feature extraction, decision-making, and performance evaluation [25]. Between offline and online methods, offline methods seem to be more practical than online methods, but more challenging as only static characteristics are available from the signature image [41].
1.4 Types of Signature Forgeries

The scope of a signature verification system is to detect whether a given signature is genuine or forged. For this reason, the methods for verification depend upon the types of forgery. Figure 3 shows the three main types of forgeries.

![Figure 3 - Types of forgery (a) genuine signature; (b) random signature; (c) simulated simple forgery; (d) simulated skilled forgery - Taken from our collected signature database](image)

In the above figure, Figure 3 (a) is a genuine signature whereas the other three signatures three different forgeries of the genuine signature. A random forgery (Figure 3 (b)) is usually represented as a signature that belongs to a different writer, in which the forger has no type of information about the signature style and the person’s name of the genuine signature. The second forgery, namely simple forgery (Figure 3 (c)), is represented by a signature with the same shape of the writer’s name. In this type of forgery, the forger knows only the name of the genuine person. The third and last type of forgery is usually referred to as skilled forgery (Figure 3 (d)) and is represented by a suitable imitation of the genuine signature model. A signature model is defined as the total signature samples produced by one author.

The three types of forgeries mentioned above are identified using different recognition methodologies. Usually, random and simple forgery types are identified using off-line methods based on static features, due to the fact that these algorithms have shown their potential in describing the characteristics related to the signature shape. Since the offline method lacks time-related information, and is not capable of modeling the handwriting motion, therefore it is harder to detect the skilled forgeries as a skilled forgery has almost the same shape of a genuine signature. For this reason, methods based more on pseudo dynamic characteristics are more suitable and robust to detect skilled forgeries, as they are able to capture handwriting motion details [35].
1.5 Handwritten Signature Verification (HSV) Applications

A reliable HSV technique can be applied in different areas of security applications. Handwritten signatures are already accepted in our societies and they continue to play an important role in financial, commercial and legal transactions [70].

1. Financial Transactions

The signature is considered to be the ‘seal of approval’ and is the preferred instrument for authentication due to its convenience. However, a handwritten signature is becoming an attraction for forgeries and threats and monetary losses continue to rise dramatically [70]. For instance cheque fraud is becoming a big issue and according to A. Kholmatov [5], MasterCard estimates a $450 million loss each year due to credit card fraud.

2. On-line Banking Transactions

HSV applications are also used in Online Banking systems for customer verification purposes. A digitising tablet is required to acquire the handwritten signature of the user. In this way, the user will not require to remember any password or PIN codes. The combined capturing of static and dynamic features makes a handwritten signature almost unique to every individual and hence very difficult to forge [70].

3. Cheque Processing

Signature verification system can be easily integrated into the cheque processing workflow, whereby the cheques are scanned and are automatically compared with the reference signatures of the individual stored in the database. Human intervention will only be required if the signature in question will result above a specific threshold [70].

4. Credit Cards

Unfortunately, due to their dynamic usage, credit cards are also a target for forgeries. Different approaches have been proposed to reduce credit card fraud. However they are still not perfect as it has been found that they lack competitive advantage or reliability issues as stated by Gubta and McCabe [25].

5. Computer User Authentication

HSV systems also have the potential to replace traditional use of passwords to access computer systems such as operating systems and information systems. A typical dynamic HSV system will of course require that a digitizing tablet be connected to each workstation to capture signature details [25].

6. Passports

Passport validation is another potential area where HSV systems can be used. A person, in this case, goes to some authorised office to provide a sample of signatures which are captured electronically on the magnetic strip of the passport (in the future passports are likely to have magnetic strips for faster processing). Once again the customer will be required to sign on a graphics tablet at the point of entry of another country, and compare it to the reference signatures stored on the magnetic strip [25].
1.6 Overview of the Report

In this chapter, a brief overview was given in order to put the reader into the right context before proceeding to the following chapters. Below is a brief outline for each chapter of this dissertation.

Chapter 2 - Literature Review
The literature review is focused on offline signature verification methods and presents an overview of several published algorithms used and results obtained. The research covers the progress in this area of research and discusses potential future work.

Chapter 3 - Methodology
This chapter is divided into four main sub-sections where data acquisition, pre-processing, feature extraction and classification methods are explained in detail.

Due to confidentiality reasons, no public database of offline handwritten signatures is yet available. For this reason a data acquisition exercise was performed in order to create a small database of signature samples. Subsequently, once the signature database is obtained, the signatures are pre-processed and prepared for feature extraction. Feature extraction is divided into three groups of features, including global features, grid features and texture features. The global feature measurements describe the structure of the signature image. Grid features provide more detail by segmenting the skeleton signature image into a number of cells, while texture features describe the changes in the black and white pixels of the signature image.

After extracting all the selected features, the classification methodology is explained in Section 3.4. The classification is based upon singe-layer RBF architecture using the mentioned feature mentioned above.

Chapter 4 - Implementation and Results
This chapter explains how the selected methodology is executed. A graphical user interface is implemented used to maintain the author signature models and also allows the user to view the signature images in three modes; original, binarised and thinned.

Furthermore, this chapter discusses 39 verification scenarios used to test the system. Several performance measurements are used to quantify the system performance and finally outlining the best results obtained. Testing of the system is performed on random forgeries already available in the acquired database.

Chapter 5 - Discussion
The discussion chapter comments on how successful the implemented architecture is. Moreover, it compares the results of different scenarios especially with regards to the Vector Quantization (VQ) technique. It also compares the obtained results with other published studies and argues about the robustness and weaknesses of the proposed architecture.

Chapter 6 - Conclusion
Finally, a conclusion is given in Chapter 6 that summarises the findings and identifying any related future work together with any recommendations. Any limitations and contributions of this study are also given in this chapter.
Chapter 2  Literature Review
2.1 Research Methods

An extensive literature search was performed in the area of signature verification methods. Initially, internet was the main source of research, from which a number of journals and conference proceedings could be found in IEEE and Elsevier Science publishers.

Furthermore, the Google Scholar [http://scholar.google.com/] together with the Scientific Literature Digital Library CiteSeer.IST [http://citeseer.ist.psu.edu/] were the main internet search engines used to search for citations.

Initially, general keywords, as listed below, were used in order to obtain a general feeling of the entire area.

- “signature verification”
- “handwritten signature verification”
- “handwritten signature recognition”
- “signature classification”

After several interesting findings, a general idea of this field was understood. At this stage, it was found that there are two types of handwritten signature verification, online and offline as explained in Section 1.3. After it was decided to focus on the offline method, the search keywords were targeted on the area of interest. The following keywords were used in order to search in more depth for offline signature verification methods:

- “offline signature verification”
- “offline handwritten signature verification”
- “offline signature verification state of the art”
- “offline signature verification comparison survey”

The results for the above mentioned keywords were numerous and exciting. It could be easily understood, that this area of research goes back to a number of decades and is still an open research problem. At this stage, the search criteria also included the year of publication in order to ignore outdated articles.

When a good grounding was acquired, the keywords were more targeted for the name of the algorithms used:

- “offline signature verification neural networks”
- “offline signature verification HMM”
- “offline signature verification Euclidean distance”
- “offline signature verification RBF”

The textbooks of Bishop [12] and Haykin [28] were used to obtain a deep understanding of the main algorithms found in the literature search. The literature search took quite a long time, however it was essential to understand the context of our interest by analysing what has been done so far, what has been achieved and where are they targeted.
2.2 Introduction

The purpose of this review is to highlight the different approaches adopted to-date to solve the problem of offline signature verification.

The study of signature verification goes back a couple of decades. Signatures were always accepted as a convenient means of authorization and identification. Regardless the long history of research in this field, “attempts to realise systems robust enough for routine practical use have not yet met with the degree of success originally expected” [20] cited by [41].

Since no knowledge is available about the process of signing, offline signature verification is quite challenging and thus complex. In fact, Quek and Zhou [59], Ismael and Samia [34] and Huang and Hong [30] (cited by [41]) explain the main difficulties in offline signature verification approach as listed below:

1. The complexity of signature patterns and the wide intra-personal variations
2. Ambiguity in pattern structure and the interaction between components
3. The minimal difference of a skilled forgery with the genuine signature
4. The quality of the signature depends on the different conditions at the time of signing
5. The number of reference signatures in the database requires a quick and efficient searching algorithm
6. Practically, no a priori knowledge is known of the forged signatures, as well as the genuine signature class is not fully established.

As discussed in Section 1.3, the basic methodology includes the data acquisition, preprocessing, feature extraction, signature comparison process, and performance evaluation. The review is split systematically in the above steps.

2.3 Data Acquisition

Since no international database of signatures is available due to data protection, the first step requires the collection of signature samples to use for the evaluation of the respective study. The bigger the number of signature specimen the greater he probability of achieving accurate results.

Due to intra-personal variations and other circumstances, Baltzakis and Papamarkos [7] suggest that the writers should try to use as much variation in their signature sizes and shapes as they should ever use in real world. Furthermore, whenever it permits, the signature acquisition should be performed on different days, ideally, without letting the signer see the signatures previously given in the previous days.

2.4 Pre-processing

Once the signature image is scanned, the next step is to pre-process the image to improve the quality of the image. Various methods were used to achieve this, including noise reduction, separating of signature from background, binarization through the identification of an optimal threshold, size normalization, data area
cropping, contrast and line improvement, edge detection by means of Sabor filter, skeletonization (also known as thinning) and segmentation. Well known techniques such as convolution masks, histogram analysis and equalization, gradient evaluation and morphological operators are used [26] [21] [54] [34] [60] [4] (cited by [41]).

2.5 Feature Extraction and Selection

Feature extraction, as explained in [1], is the process of extracting the characteristics of a pre-processed signature image. This process must be supported by feature selection that will highly influence the results in the evaluation of the comparison method. It is paramount to choose and extract features that are:

- Computationally feasible
- Lead to a good classification system with a low False Rejection Rate (FRR) and False Acceptance Rate (FAR).
- Reduce the problem data into a manageable amount of information without affecting the signature structure.

Basically, there are three general types of offline signature features, which can be categorised as global features, local features and pseudo-dynamic features.

Briefly, the study of Sabourin [68], cite references the following methods which were developed up to the mid 90’s.

1. 2D Transforms (Nemcek and Lin 1974)
2. Histograms of directional data [18], [17], [73]
3. Curvature analysis [14]
4. Horizontal and vertical projections of the writing trace of the signature [3]
5. Local measurements on the writing trace of the signature [50]
6. Position of feature points extracted from the skeleton image [69]
7. Structural approaches [57]

Harrison 1981 (cited by [17]) states that random and simple forgeries cover almost 95% of fraudulent cases. Consequently, Drouhard et al [17] suggests that a two-stage automatic handwritten signature verification system (AHSVS) would be more practical solution, where the first stage handles simple and random forgeries, while the second stage handles skilled (more difficult) forgeries. In 1992, Sabourin took a different approach in solving the problem by using Neural Networks (NN) with directional probability density function (PDF). However, this approach did not produce good results since a mean total error rate of almost 5% was obtained. This was due to the fact that global shape factors like the directional PDF do not consider the spatial position of local measurements in the representation of the signature.

Furthermore, similar level of performance was achieved with invariant moments [69]. In view of the lack of knowledge about the location of local measurements with global features (e.g. directional PDF), Sabourin took a different approach by using the extended shadow code (ESC). The reason is that ESC, as a shape factor, permits the local projection of the handwriting without losing the knowledge of the location of measurements in 2D space [66].
In fact, the most remarkable results were obtained by Sabourin and Genest [64], in 1994, where a mean total error rate of 1% has been obtained with the database of signatures used. In this study, Sabourin and Genest used the ESC as a global feature vector. They found that the class of shape factors seems to be a good compromise between the global features and the local features of the signature. The shadow codes of the signature binary image are calculated by superimposing a bar mask array where each bar is associated with the spatially constrained area of the signature. A shadow projection operation is the simultaneous projection of each black pixel into its closest horizontal, vertical and diagonal (HVD) bars.

In 1995, Sabourin carried out two other research studies with different colleagues. In his first study [49] carried out jointly with Murshed and Bortolozzi, they studied a new approach to offline signature verification where no a priori knowledge was used to verify the signatures. In their study, Murshed et al. [49] considered signature verification as a one-class problem (the class of genuine signatures for one writer). An OHSV system is trained by a Fuzzy ARTMAP neural network only using genuine signatures. Murshed et al. [49] also introduced an identity grid for each writer in a way that its shape reflects the average overall shape of the signature samples of the writer. The evaluation was done on a database of 5 writers having 40 signatures each giving a total of 200 signatures. Although promising results were achieved, Sabourin suggested further evaluation on a larger database.

In the second study [65], Sabourin and Genest continued their study of ESC by evaluating several multi-classifier combination strategies. Their results showed that the use of integrated classifiers allows the implementation of signature verification systems without a pre-established feature selection. This results in a single shape vector.

### 2.5.1 Global Features

The global features describe the signature image as a whole, generally related to the geometric of the signature. Example of global features include height, width and height/width ratio, vertical and horizontal centre of gravity, horizontal and vertical projection peaks, baseline shift, kurtosis measures, relative kurtosis and skewness, transforms, gradients, polygon descriptions together with upper and lower envelope based characteristics [54] [31] [60] [57] [6] [4] (cited by [41]).

The intra personal variations with respect to the global aspect appear to be very low. In fact, as shown in Figure 4, the twenty skeleton signatures obtained from the same person have more or less the same geometric orientation [68].

A key paper in this field, which is almost always cited by other key papers, is [57] where Qi and Hunt developed algorithms for extracting global geometric and local grid features. A multi-scale verification function was built from a combination of these features, in which it resulted over 90% in rejecting skilled forgeries and was perfect in rejecting simple forgeries based on a simple database.
2.5.2 Local Features

Local features are very similar to global features however the difference is that local features are applied to the cells of a grid virtually super imposed on an image, or to particular elements obtained after signature segmentation. Local features include the slant angle of an element, density factor (number of black pixels), length ratio of two consecutive parts, position relation between the global and local baseline, upper, central line features, corner line features and critical points [26] [31] [34] [7] [4] (cited by [41]).

It is important to note that global features are less sensitive to noise and small variations. On the other hand, local features used to describe signature portions, are more noise sensitive but are independent on each other. Furthermore, they are computationally expensive but much more accurate [34] (cited by [41]).

Sabourin et al [68] tackled the problem of offline signature verification with different feature measurements. A new algorithm, using local granulometric size distributions was adopted where the signature image (512 x 128 pixels) is centered on a grid of rectangular retinas and excited by local portions of the signature. Granulometry is the result of a set of morphological operators. Mathematical morphology is the analysis of geometric structures, such as Erosion and Dilation. A k-NN classifier was used to identify random forgeries by using a feature vector with a dimension equal to the number of retinas. The evaluation for this study [68] assumes that all signatures are of similar size, and the respective strokes of the signature fall into approximately the same retinas. The best result obtained from this study is a 0.02% of equal error rate.

Due to their detailed analysis, local features are said to vary significantly for the same writer. For instance, Figure 4 shows how the local positions of strokes vary from one signature to another for the same writer.

Figure 4 - Forty skeletons of genuine signatures from three writers with centered and superimposed in the image plane - Taken from our collected signature database
2.5.3 Pseudo-dynamic Features

Global and local features are very useful to detect random and simple forgeries. However, they do not produce the same results for skilled forgeries, which are much similar to the genuine signatures. Several attempts have been made to improve this situation by extracting pseudo dynamic features from a static signature image.

The study of Ammar et al [3] (cited by [38]) is one of the pioneers in this area of offline signature verification. They approached the problem by using statistics of high gray level pixels to identify pseudo-dynamical characteristics of signatures. The level of gray-scale is directly related with pen pressure, which represents an individual characteristic of each signature.

Pseudo-dynamic characteristics include the pressure area, stroke curvature and stroke regularity as are clearly explained in Figure 5.

Moreover, another approach is related to the evaluation of areas of different density [22] [52] (cited by [41]). Usually three levels of different pressure are used in handwriting, where they are generally connected with different speed at different parts of the signature [59] (cited by [41]).

K. Huang and Hong [31] (cited by [41]) suggest that the combination of global and local features together with pseudo-dynamic characteristics will improve the recognition rate for offline signature verification methods.

2.6 Published Results

In the survey carried out by Sabourin [66] he concluded that the local shape factors like the ESC and the one based on the pattern spectrum produce better results than global shape factors like PDF or the structural approach. On the other hand the cognitive approach (Fuzzy ARTMAP) produce results which are closed to those obtained using global shape factors and structural approaches.
In a number of studies (cited by [66]), two types of classifiers, Nearest Neighbour (NN) and Minimum Distance (MD) with threshold were used for the implementation of all methods except for the one based on Fuzzy ARTMAP (See Table 1).

<table>
<thead>
<tr>
<th>Approaches</th>
<th>NN</th>
<th>MD with Nref = 6</th>
<th>Nref = 18</th>
</tr>
</thead>
<tbody>
<tr>
<td>Structural Approach [69]</td>
<td></td>
<td>2.49%</td>
<td></td>
</tr>
<tr>
<td>Directional PDF [17]</td>
<td>2.69%</td>
<td>5.61%</td>
<td></td>
</tr>
<tr>
<td>ESC [64]</td>
<td>0.01%</td>
<td>0.88%</td>
<td></td>
</tr>
<tr>
<td>ESC with Integrated classifiers [65]</td>
<td>0.0% (0.18%)</td>
<td>0.05% (2.88%)</td>
<td></td>
</tr>
<tr>
<td>Cognitive Approach [48]</td>
<td></td>
<td>3.56%</td>
<td></td>
</tr>
<tr>
<td>Pattern Spectrum [68]</td>
<td>0.02%</td>
<td>0.85%</td>
<td></td>
</tr>
<tr>
<td>PS with Integrated classifiers [68]</td>
<td>0.0% (0.05%)</td>
<td>0.27% (2.38%)</td>
<td></td>
</tr>
<tr>
<td>Binary Shape Matrices [67]</td>
<td></td>
<td>0.81%</td>
<td></td>
</tr>
</tbody>
</table>

Table 1 - Experimental results, mean total error rate $E_t$ in % (Rejection Rate $R_t$) [66]

Sabourin, in [66], also proposes that the research of signature verification should be integrated into the cognitive approach proposed by Murshed in [48] (cited by [66]), and if the merge is successful then it could be possible to implement an entire AHSVS to handle all forgery classes by just using genuine signatures for training, i.e. without a priori knowledge of forgery signatures.

### 2.7 Comparison Process

The next step is the comparison process, where the signature being verified is compared with the reference signatures stored in the database. Over the years a number of approaches have been developed to evaluate their performance and the actual results in terms of FRR and FAR. The objective is to keep both FRR and FAR as low as possible.

#### 2.7.1 Simple Distances

Simple measures of similarities like Euclidean or Mahalanobis distance are most often used for classification. They are usually used when feature values represent geometric interpretation like coordinates of a point in the feature (dimension) space.

For instance, Euclidean distance between global features was used in [57] to measure the similarity in the height, width, slant angle, vertical center of gravity, maximum horizontal projection, area of black pixels, and baseline shift. On the other hand, in [21] (cited by [41]), the inverse Mahalanobis distance is used as measure of similarity between global and local feature vectors.

Mizukami [47] (cited by [38]) proposed a system that is based upon a displacement extraction method. The method used consists of the minimization of a function, defined as the sum of the squared Euclidean distance between two signatures and a penalty term of the smoothness of the displacement function. The signature images are transformed into coarse images by Gaussian filters to avoid the problem of local minima.
2.7.2 Hidden Markov Models (HMM)

A different approach to signature verification is the Hidden Markov Model (HMM) of intrapersonal and interpersonal variations of signature models. Notwithstanding the fact that it is usually used for on-line signature verification, it is also possible to be applied in segmented off-line signatures.

The approach used in [35] involves a grid segmentation technique to extract three important features from the signature image: pixel density feature (static), pixel distribution feature static (also called ESC [64]) and the predominant axial slant of the skeleton signature image (pseudo-dynamic). Subsequently, a set of codebooks was generated for each feature by a Vector Quantization technique based upon the k-means algorithm. During training, an HMM signature model is adapted for each writer. The cross-validation procedure was used to dynamically define the optimum number of states for each specific signature’s model whilst the Forward algorithm [37] was used in the verification process. Figure 7 depicts the acceptance and rejection areas used in the verification process. Promising results were obtained for random and simple forgeries, however further discriminate features are required to obtain better results in skilled forgeries.
Further studies using the HMM approach were performed in [36] and [16]. The scope of the approach was to outline the variability within interpersonal and intrapersonal skills and how they affect the overall performance of the verification process. On the other hand the latter study uses a Discrete Radon Transform (DRT) together with an HMM. Through this approach they have managed to develop a system that automatically authenticates offline signatures. From a database of over 900 signatures from 22 writers an equal error rate (EER) of 18% was achieved when only skilled forgeries were considered. On the other hand an EER of 4.5% with respect to random and simple forgeries was achieved. When the number of signatures was increased to 4800 signatures the EER for skilled forgeries decreased to 12.2%.

2.7.3 Neural Networks

Neural Networks are popular for their learning capability. They attracted the attention of many and were used as classifiers for signature verifications. The most popular neural networks are feed forward multi-layer perceptrons (MLPs) using backpropagation training techniques.

Mighell et al [45] (cited by [25]) is considered as the pioneer in this area. The fact that neural networks have produced plausible results in other signal processing tasks such as character recognition Mighell et al [45] believed that promising results could also be achieved in signature verification methods. A feed-forward network was used with a backpropagation-learning algorithm with both dwell and momentum. Having used a very small database, the obtained results were promising where using a threshold of 0.5 the network produced 1% FRR and 4% FAR.

This work helped others to improve the results by increasing the number of specimen signatures and applying different preprocessing techniques. McCormack and Brown [44] (cited by [1]) used Fourier and Haar wavelet transforms in the preprocessing stage. Both back propagation and cascade-correlation networks were used to evaluate and compare the generalization ability of wavelet encoded signature data. McCormack and Brown concluded that neural networks are very sensitive to each signer. Furthermore, they reported that Haar wavelet data was found to be more efficient than Fourier encoded data to solve the problem of offline signature verification.

In another study [15] (cited by [1]), Cardot et al also adopted a neural network approach in the learning and decision phase of a signature verification system. This study was targeted at simple and random forgeries. Cardot et al divided the problem into three levels. The first consists of two neural networks of the non-supervised Kohonen map type to classify signatures of the different writers into different classes. The second level consists of two multi-layer neural networks using an error gradient backpropagation learning technique. These two neural networks are specific to every author. Finally, the third level is another neural network, also using backpropagation learning algorithm, used to take the final decision of accepting or rejecting the questioned signature.
In [17] mentioned earlier, Drouhard et al mentions that a backpropagation network classifier performed better than the threshold classifier and compares well with the k-Nearest neighbour classifier.

In [59] Quek and Zhou (cited by [41]), a self-organizing Kohonen algorithm was used. Four groups of features were used being,

1. Horizontal and vertical binary and gray level features
2. Global baseline – the position of the maximal value of the horizontal or vertical projection
3. Pressure – 7 features extracted related to pressure
4. Slant features – 4 slants examined

The classifier is a five layer neural network. The input layer represents the feature measurements namely height and width amongst others. The second layer represents labels such as small, median and large. The third layer represents fuzzy rules. Finally the fourth layer represents labels such as low, median, high, etc of the output layer. The output layer, fifth layer, is a non-fuzzy data representing the output result.

2.8 OHSV using RBF Neural Networks

The survey of Marinai et al [43], cites the work of Gori and Scarselli [24], where it was found out that although MLPs are very good at performing discriminative classification between patterns of well-defined classes they are not adequate for applications requiring a reliable rejection. It was suggested that other architectures, such as autoassociator-based classifiers and Radial Basis Functions (RBF) networks are more suitable. In fact, Gori and Scarselli [24] (cited by [43]) concluded that MLPs are more adequate for rejecting dubious patterns while RBFs and autoassociators performs better for handling outliers.

These arguments motivated us to investigate the performance of RBFs in offline signature verification. After surveying the literature, only one study was found that applied RBFs to this area, namely the work by Baltzakis and Papamarkos [7]. Even Mr. Baltzakis is not aware of any study in this field which applied RBFs [10].

Baltzakis and Papamarkos [7] proposed a new signature verification technique based on two-stage neural network classifier. In the first stage they used a two-stage perceptron one-class-one-network (OCON) classification structure where it combines the decision results of the neural networks with the Euclidean distance obtained using three groups of features being global, grid and texture features. In the second stage they used an RBF neural network which is fed by the first stage classifier to take the final decision as depicted in Figure 8.
In the pre-processing stage they used noise reduction, data area cropping [23] (cited by [7]), width normalization, and skeletonization techniques. For feature selection and extraction, Baltzakis and Papamarkos were motivated by the work carried out by Qi and Hunt [57] where a number of global, grid and texture features were extracted from the pre-processed signature image.

As already mentioned above, the first stage is based on the use of three MLPs, one for each group of features. The second stage, which is the area of our interest, consists of an RBF neural network based on Gaussian functions which are characterised by the mean vectors (centers of two classes) $m_1$, $m_2$ and their respective covariance matrices.

Baltzakis and Papamarkos [7] suggest that the proposed structure which is based on 160 features, leads to small and easily trained classifiers. Furthermore, the system is more scalable, in a way that including new signatures in the system will not require training the entire system from scratch. This means that no *a priori* knowledge about the number of authors and number of signatures is required at design stage. Table 2 shows the results achieved in the verification stage. It is important to note that these results reflect the worst case scenario. This is due to the fact that the signers were asked to use as much as intrapersonal variations as possible in different phases.

<table>
<thead>
<tr>
<th>Cases that should be accepted</th>
<th>500</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cases that should be rejected</td>
<td>57,000</td>
</tr>
<tr>
<td>Accepted</td>
<td>1485</td>
</tr>
<tr>
<td>Rejected</td>
<td>56,015</td>
</tr>
<tr>
<td>Correct acceptances</td>
<td>485 (97%)</td>
</tr>
<tr>
<td>False rejections</td>
<td>15 (3%)</td>
</tr>
<tr>
<td>Correct rejections</td>
<td>51,211 (90.019%)</td>
</tr>
<tr>
<td>False acceptances</td>
<td>5689 (9.81%)</td>
</tr>
</tbody>
</table>

Table 2 - Results obtained by Baltzakis and Papamarkos [7]
2.9 Terms of Reference

The objective of this study is to study and compare existing techniques used in offline signature verification. Furthermore, an offline signature verification system was implemented based on a Radial Basis Function Neural Network (RBFNN) classifier. The implementation was carried out in 5 main stages being:

1. Data acquisition
2. Pre-processing
3. Feature selection and extraction
4. Classification
5. Evaluation of the results

The evaluation process analysed which features produced the best results in terms of False Acceptance Rate (FAR) and False Rejection Rate (FRR).

The deliverables of this project include:
- An extensive review of the existing methods used with respect to offline handwritten signature verification
- Signature data acquisition, together with the implementation pre-processing and feature extraction techniques
- A single layer RBFNN classifier used to take the final decision whether a questioned signature is genuine or not
- Performance results of the RBFNN in terms of FAR, FRR, TER and MER in view of the selected features
- A discussion on the obtained results and recommendations for future work regarding the use of RBFNN to offline handwritten signature verification
- A conclusion and personal evaluation of the entire study

2.9.1 Changes to original Project Objectives

The original proposal stated that a comparison analysis between RBFNN, Multi-Layer Perceptron (MLP), and a two stage-approach (MLP-RBF) [7] will be implemented. However, due to the effort required to perform data acquisition, pre-processing and especially feature extraction, it was decided to invest more effort to select and implement the required feature extraction techniques. Therefore, the proposed solution uses an RBFNN, as a classifier, and evaluates its performance with respect to the extracted features. However, the project title remains the same, namely “How effective are Radial Basis Function Neural Networks for Offline Handwritten Signature Verification?”

Literature review revealed several classes of suitable signature features have been used and were recommended in several publications. Since feature extraction is fundamental to the performance of an OHSV, it was decided to invest more effort on feature extraction and selection. Therefore, the work is now focused on the performance evaluation of an entire RBFNN architecture using different classes of signature features.
2.9.2 Project Motivation

Offline Signature Verification is still an open research problem. In the past years, several attempts have been made, where different techniques and methodologies were tried out to solve the problem.

RBFNNs are distinguished for their robustness in eliminating outliers and for the relatively simple computation due to single layer architecture. Furthermore, RBFNNs are quite new in this domain. In fact, only one study has been found where a two-stage approach was used; the first stage consisted of MLP classifiers and an RBFNN was used in the second stage to take the final decision.

2.9.3 Project Plan

Project tasks were carried out according to the project plan depicted in the Gantt chart below. Figure 9 and Figure 10 (See page 23) show two Gantt charts of the proposed and the actual project plans respectively. There were no major deviations from the proposed plan. The slack time planned for contingency was enough to cater for delaying some of the tasks.

2.10 Conclusion

It is evident that after a period of more than 30 years of active research, offline signature verification is still an open problem. Throughout these years different techniques emerged, where the one with the lowest total error rate remain the one produced by Sabourin and Genest [64] in 1994. In this study an extended shadow code was used as a global feature vector.

As suggested by a recent survey in [43], RBF networks are still to be investigated in this application. In this work, an RBF solution to offline signature verification is studied and reported in the next chapters.
Figure 9 - Proposed Project Plan

Figure 10 - Actual Project Plan
Chapter 3  Methodology
3.1 Data Acquisition

As discussed in Section 2.3, there exists no international database containing handwritten signatures. This is mainly due to confidentiality. For this reason, a database of reference signatures was required for this dissertation. This is necessary for training and testing purposes.

The local Data Protection Commissioner was contacted to identify the procedures required to collect this information without violating Data Protection laws (See Appendix B.1.) Article 16 of the Data Protection Act states, that a handwritten signature is not considered as a sensitive personal data and hence no authorisations are required. However, it has been suggested that a consent form should be compiled and distributed among those accepting to contribute their signatures (See Appendix B.2.). Further to this response, a small booklet containing a consent form was compiled and presented to each author contributing in this process (See Appendix C).

In the process of collecting signature samples the recommendations of Osborn [53] and Baltzakis [7] were taken in consideration. Osborn emphasised how the signatures of the same author vary under different conditions. He recommended that the more signatures obtained the better the signature model is. On the other hand Baltzakis in [7] recommends that the authors are requested to use intra personal variations as much as possible.

Further recommendations were provided by Mr. Gaffiero, a Maltese graphologist (See Appendix D). Mr. Gaffiero, who is also a technical referee to the Maltese Courts, suggested including different sized rectangles which can affect the proportionality of the signature. He emphasised the fact that a signature may also vary according to the visible orientation guidelines. On the other hand, Baltzakis [8] suggested using different pens having different colour and different tip widths since this would affect the acquired signature.

The data acquisition process was carried as follows.

1. Acquisition of a total of 40 signatures from each author; 25 on blank sheets and 15 in provided random sized rectangles
2. Acquiring the signatures on 5 different days (when possible); 5 on blank sheets and 3 in provided rectangles on each day.
3. Using 8 different pens which vary in colour (black, blue, red and green) and type (ball point, normal pen and fountain pen)
4. Authors asked to use as much as intra personal variations as possible

A total of 2492 signatures were collected from 65 different signers. These were subsequently scanned using a resolution of 300 dpi and stored as a BMP file type (no compression used). All signature sheets were manually cropped using photo editor software to separate them as individual images.

The 65 people contributing to this exercise comprised mainly of family members, friends and work colleagues. Their background comprises different context; education level, language, age and region. They represent a wide variety of signature styles, from completely incomprehensible line strokes to clear and tidy handwriting.
3.1.1 Observations

During the signature acquisition process it was observed that the suggested factors by Dr. Baltzakis and Mr. Gaffiero really affect the individuals. The different sized frames included in the template prepared for this exercise (See Appendix C) effected the proportionality of some signatures, whereas other signatures kept a constant shape irrespective of the provided frame. The diagram below shows an example of three genuine signatures of the same individual. The available space for the signature was a determining factor in the way the signature was written.

![Signature proportionality affected by provided frames](image)

Some people participating in this exercise were psychologically affected with some of the used pens. For example, some individuals complained that they were uncomfortable to sign with a ball point. Others complained that they disliked signing with a red pen.

Furthermore, it was also noted that some signers were affected by the baseline of the provided frames. For example, some individuals managed to constantly overlap the provided border line even if they were specifically asked not to do so.

3.1.2 Boundary to this Work

In real life scenarios, people are acquired to provide signatures on bank cheques, passports, fiscal receipts and others. This means that in reality, the signature is incorporated within some kind of background. Hence, prior to verification, the signature must be segmented from the background in order to remove the irrelevant information. Since the scope of our study is focused only on signature verification, the signature models were all acquired on white blank sheets of paper without any background.

Different techniques are already used and still investigated to segment a handwritten signature from its background. Particularly, the studies of Basir et al [11] and Lee et al [40] are focused on the segmentation of handwritten signatures from the background of a bank cheque.
3.1.3 Summary

The recommendations of Gaffiero and Baltzakis represent the real life scenario, where the signer is requested to sign under different circumstances. This means that the signers are asked to use as much intrapersonal skills as possible. The method used for signature acquisition represents the worst-case scenario due to the high variability in the signatures of the same author. Evidently, the high intrapersonal variations will be reflected in the results obtained in Chapter 4.

3.2 Pre-processing

The pre-processing stage is based upon the work carried out by Baltzakis and Papamarkos in [7]. It is divided into four steps mainly

- Data area cropping,
- Width normalization,
- Binarization,
- Skeletonization.

(See Appendix Error! Reference source not found. for implementation details)

Noise reduction algorithms were not used. This was due to the fact that the signature acquisition was performed on white sheet of papers and hence very little noise was envisaged on the images produced during scanning. Furthermore it was decided not to eliminate the small noise created during scanning in order to evaluate the proposed method with some noise in the images. The processed signature images are stored in the same database mentioned in Section 3.1.

3.2.1 Data area cropping

Initially, the original 24 bit colour image is segmented from the background to remove unnecessary data from the image. This is done by first binarizing (See Section 2.4) the image and subsequently using the segmentation method of vertical and horizontal projections [23] (cited by [8]). This process consists of scanning the binary image twice; vertically and horizontally. For each vertical and horizontal projection, the number of black pixels is counted and if these do not exceed a given threshold, the scanned pixels will be discarded from the image.

Eventually, the original image is segmented from the background, so that the white space surrounding the signature is discarded as shown in Figure 12.

![Figure 12 - Data Area Cropping; (a) Original image, (b) white spaces removed](image-url)
3.2.2 Width Normalization

The cropped image is scaled to a fixed width (503 pixels) while keeping an aspect ratio with the height. This will ensure that the image maintains the same proportionality. Bicubic interpolation is used in the scaling process to perform width normalization on the original 24-bit colour image. This process is necessary so that all signatures are analysed on the same width.

The normalised width was not randomly selected. The original widths of the coloured signatures were put on a histogram, and the normalised width was chosen to be the mode value; that is the most common width. This method enabled to scale only the least number of signature images.

3.2.3 Binarization

Binarization is the process of converting a coloured image into a black and white image. This process, initially, converts the coloured image into a grey image. The image is converted from a 24-bit image into an 8-bit image by combining the three bands into a single band.

A threshold is subsequently calculated based upon the histogram generated from the grey image. The calculated threshold is used to convert the grey image into a binary image (black and white pixels only). Figure 13 shows a normalised and binarised version of Figure 12.

![Figure 13 - Normalised and Binarised Signature Image](image)

3.2.4 Skeletonization

Skeletonization, also known as thinning, is an important pre-process step necessary in many image analysis operations. The main objective of thinning is to reduce data storage without losing the structural information of the image. It is also used to reduce transmission time as well as to facilitate the extraction of morphological features from digitised patterns. It reduces the amount of data to be stored by transforming the binary image into a skeleton.
The thinning algorithm proposed by Quek and Zhou [58] was selected for the purpose of this dissertation. The algorithm comprises the following steps.

1. Initialise a 2-dimensional array of 1s and 0s representing black and white pixels respectively of the binary image.
2. Initialise a 2 dimensional array (same size of the image array) where all data elements have a value of 1.
3. Scan the binary image using a 3x3 array mask.
4. If the current pixel is black then count the number of previous neighbours \( PN(P) \). Previous neighbours are the neighbours of the previous iteration.
5. If \( PN(P) \) is less than 8 then and satisfies condition \( a \) and (condition \( b \) or condition \( c \)), then flag it for deletion. The three conditions are:
   a. Count the number of current neighbours \( CN(P) \). Current neighbours are the neighbours of the current iteration.
   b. Calculate the neighbourhood transitivity, \( \text{Trans}(P) \), that is the number of white pixels followed by a black pixel
   c. Check whether it matches one of the smoothing templates in Figure 14
6. Delete the flagged pixels.
7. Repeat steps 3, 4, and 5 until no pixels are flagged for deletion.

![Figure 14 - Thinning Process: Smoothing Templates used in boundary pixel check [58]](image)

After a number of iterations this algorithm produces a skeleton of the binary image as shown in Figure 15.

![Figure 15 - Example of Thinned Signature](image)
3.3 Feature Extraction and Selection

The choice of powerful set of features is essential in optical recognition systems. The selected features must be suitable for the application of the applied classifier. The feature extraction methods used in this stage are based on the ones suggested by Baltzakis and Papamarkos in [7]. Feature extraction is divided into 3 sets of features being

1. Global features
2. Grid features
3. Texture features

(See Appendix Error! Reference source not found. for implementation details)

3.3.1 Global Features

Global features provide information about the entire structure of the signature. Qi and Hunt [57] propose a number of algorithms to extract global features from a signature. The following global features were extracted from the skeletonised signature image.

1. Signature Height

The height of the signature (in pixels), after width normalization, is considered as a global characteristic.

2. Height-to-Width Ratio

Height-to-Width ratio is the proportionality rate of the skeleton signature image [7]. This is calculated by dividing the height with the width of the signature.

3. Pure Width

This is the width of the skeleton signature image with horizontal blank spaces removed. The algorithm is based upon the pseudo code provided by Qi and Hunt in [57] as shown below.

Definition

- *Amplitude-threshold* - denotes the minimum number of pixels in the current vertical projection that are accepted. The pure width starts to be measured when the amplitude threshold is exceeded and stops when it is below.

- *Length-threshold* - denotes the number of white spaces accepted as part of the pure width, after the vertical projection goes below the amplitude threshold

These are both arbitrarily values and they depend on the type of data in question. A value of 5 for both thresholds was found to give the best results for the skeleton signature images.

The following is the pseudo-code of the applied algorithm to calculate the pure width of the skeleton signature image.
• Set the over-threshold and sub-threshold to zero
• Count the vertical projection by
\[ P_v[j] = \sum_i I_{ij} \text{ for } j = 0,1,...,N-1 \] (Eq. 1)
where \( I_{ij} \in \{0,1\} \) indicates the pixel level at the \( i^{th} \) row and \( j^{th} \) column and \( N \) denotes the signature width.
• Initialise the over-threshold counter when the vertical projection exceeds any given amplitude threshold
• Initialise the sub-threshold counter when the vertical projection goes below the given amplitude threshold. However, the over-threshold continues until the sub-threshold counter exceeds a given length threshold, called spatial delay.
• When the sub-threshold counter exceeds the given length threshold, calculate the difference between the over-threshold and sub-threshold and accumulate this difference to the width counter each time the sub-threshold is stopped.
• Go back to the first step until all projections are calculated.
• The pure width is the accumulated width counter

4. Pure Height
Similarly, this is the height of the signature after removing the vertical blank spaces. The same algorithm used to find the pure width was also used in this case. The horizontal projection \( P_h[i] \) was used to count the horizontal number of black pixels.
\[ P_h[i] = \sum_i I_{ij} \text{ for } i = 0,1,...,M-1 \text{ (} M \text{ denotes the signature height) } \] (Eq. 2)

5. Image Area
Image area is the number of black pixels in the skeleton signature image [7]. This is calculated by generating a histogram of the two colours available in the signature (black and white) and gets the frequency value of bin representing black pixels.

6. Vertical Centre of Gravity
Vertical centre of gravity is a measurement indicating the vertical location of the signature image [57]. It is calculated as
\[ C_v = \frac{\sum_i i \times P_h[i]}{\sum_i P_h[i]} \] (Eq. 3)

7. Horizontal Centre of Gravity
Horizontal centre of gravity is a measurement indicating the horizontal location of the signature image [57]. It is calculated as
\[ C_h = \sum \frac{i \times P_i[i]}{\sum P_i[i]} \]  
(Eq. 4)

8. **Baseline Shift**

It is the difference between the vertical centres of gravity of the left and right part of the skeleton signature image [7]. It is a measurement indicating the overall orientation of the signature. This was calculated by splitting the signature image vertically into two halves and calculate the vertical centre of gravity for each half, \( C_L \) and \( C_R \). Then the baseline shift is defined as \( BS = C_L - C_R \).

9. **Maximum Horizontal Projection**

The skeleton signature image is scanned vertically and each time calculating the horizontal projection [7]. The horizontal projection represents the number of black pixels in the current row. Then, the row containing the maximum number of black pixels is taken to represent the maximum horizontal projection.

10. **Maximum Vertical Projection**

Similarly to above, the maximum vertical projection represents the maximum number of black pixels in a column when scanning the skeleton signature image horizontally [7].

11. **Vertical Projection Peaks**

This represents the number of local maxima of the vertical projection histogram [7]. The vertical projection histogram is the frequency of black pixels for each column of the skeleton signature image.

12. **Horizontal Projection Peaks**

Similarly to above, this represents the number of local maxima of the horizontal projection histogram [7].

13. **Global Slant Angle**

The global slant angle represents the overall direction of line strokes in a skeleton signature [7]. The original 24-bit colour image is rotated from \(-45^\circ\) to \(45^\circ\) with a step of \(5^\circ\), using Bicubic Interpolation. After each rotation, the coloured signatures is pre-processed, that is scaled (width normalization), binarised and skeletonised. Subsequently, the number of vertical 3-pixels connections is counted from the rotated skeleton image. The global slant angle is the angle having the maximum number of vertical 3-pixels connections.
14. Local Slant Angle

The local slant angle represents the angle of long or dominant strokes in the skeleton image [7]. The original image is rotated similarly as mentioned above. For each angle of rotation the vertical projection histogram is calculated and the highest 70 projections are summed up. The local slant angle is the angle having the maximum sum of the top 70 projections.

Figure 16 shows an example of how the skeleton image is rotated at steps of 5° from -45° to 45°. The four skeleton images at the right are snapshots during rotation to calculate the global and local slant angles.

![Figure 16 - Image Rotation for finding Global and Local angles; (a) Original Image; (b) Rotation of -45°; (c) Rotation of -10°; (d) Rotation of 15° and (e) Rotation of 30°](image)

15. Number of Edge Points

According to Baltzakis and Papamarkos [7] an edge point is a black pixel having only one 8-neighbour. Figure 17 shows an example of a skeleton image having two edges. Both edges are encircled indicating that they have only one black pixel in the 8 neighbourhood.

![Figure 17 - Number of Edges Calculation Example](image)
16. Number of Cross Points

Baltzakis and Papamarkos [7] state that a cross point is black pixel which has at least three 8-neighbours. Figure 18 shows six different cross points in a skeleton image.

![Figure 18 - Cross Points Calculation Example](image)

However, if we consider cross point C2 (See Figure 18), for instance, we notice that the definition is not so generic. By zooming on this cross point, as shown in Figure 23, it can be noticed that by the above mentioned definition, C2 would be composed of 4 cross points, P1, P2, P3 and P4, since they all have at least three black 8-neighbours. In fact, H. Baltzakis was personally contacted [9] in order to discuss this issue. He immediately acknowledged the problem and recommended to tackle it using a connected components algorithm.

![Figure 19 - Zoom in a Cross Point](image)

Consequently, a connected components algorithm [56] was applied on the skeletonised image. The following is a pseudo-code of the applied algorithm:

- Create a two dimensional array reflecting the thinned binary image and fill the array with zeros.

- Scans the image pixel-by-pixel (from left to right and top to bottom) and label each black pixel, having at least three 8-neighbours, with a unique number (See Figure 20). The unique number is calculated storing a counter which increments by 1 for each pixel scanned.
Once the entire image is scanned, scan the two dimensional matrix containing either zeros or unique numbers representing black pixels with at least three 8-neighbours.

For each cell, having a value greater than zero, its value is replaced with the minimum value of its 8-neighbours. This will result in group of cells with the same label id as shown in Figure 21.

The number of cross points is the total number of non-zero different label ids in the two dimensional matrix. In this case there are only two cross points in the image; 24 and 28.

17. Number of Closed Loops

The number of closed loops is the number of closed circles in a skeletonised image. In [7], Baltzakis and Papamarkos define the number of closed loops as

\[ CL = 1 + \frac{EL - EP}{2} \]  \hspace{1cm} (Eq. 5)

where EP denotes the number of Edge Points (calculated above) and EL, the number of extra departures, is defined as

\[ EL = \sum_{All\ cross\ points} [(Number\ of\ 8-neighbours) - 2] \]  \hspace{1cm} (Eq. 6)

The number of extra departures is calculated as:

- Scans the resultant two-dimensional array used to find the number of cross points
• Consider a connected component occupying an $n \times m$ region. This region is extended by considering a 1-pixel border around it. Calculate the number of edge points using the same method described above.

![Component 1 and Component 2](image)

Figure 22 - Extra Departures Calculation Example

• Subtract two from the value obtained and accumulate it to a variable EL for each cross point (as per formula (5)).

\[
\begin{align*}
\text{EL}_1 &= (\text{No. of edges of Component 1}) - 2 \\
\text{EL}_1 &= 3 - 2 \\
\text{EL}_1 &= 1 \\
\text{EL}_2 &= (\text{No. of edges of Component 1}) - 2 \\
\text{EL}_2 &= 3 - 2 \\
\text{EL}_2 &= 1 \\
\text{EL} &= \text{EL}_1 + \text{EL}_2 \\
\text{EL} &= 2
\end{align*}
\]

Where EL$_1$ denotes the extra departures for Component 1 and EL$_2$ denotes the extra departures for Component 2.

• Hence the number of extra departures is the final value stored in variable EL. As shown above, in this example EL = 2.

• Since the number of edge points of Figure 20 is 4, then using formula (6), the number of cross points is given by

\[
\begin{align*}
CL &= 1 + \frac{2 - 4}{2} \\
CL &= 1 + \frac{-2}{2} \\
CL &= 1 - 1 \\
CL &= 0
\end{align*}
\]

Hence, as clearly shown in Figure 20, no closed loops were found. On the other hand, Figure 23 shows a skeletonised signature having 7 closed loops.

![Closed Loops Example](image)

Figure 23 - Closed Loops Example
3.3.2 Grid Features

As explained in the literature (See Chapter 2), grid segmentation is a technique used to zoom in the image for further detail analysis [36]. A virtual grid of $12 \times 8$ segments (See Figure 24) is superimposed on the skeletonised image so that for each segment the following features are calculated:

1. Pixels Density
2. Pixels Distribution
3. Predominant Axial Slant

1. Pixels Density
Since the authors were requested to provide their signature model with different pens having different colour and different tip thickness, the binary image does not reflect the real pressure of the signature. Hence, the skeletonised image was preferred over the binary image to count the number of black pixels in each cell of the grid.

![Figure 24 - Pixel Density Example](image)

2. Pixels Distribution
Similarly, the skeleton signature image was used to calculate the pixel distribution. Pixel distribution represents the pixel geometric distribution in a cell. The black pixels are projected in four side-line cell sensors from the central axis of the cell. Each sensor provides a numerical value corresponding to the total of the projected pixels.

![Figure 25 - Pixel Distribution Example](image)
3. **Predominant Axial Slant**

The predominant axial slant is a value representing the predominant inclination in each cell. For each cell the number of three pixels connections is calculated against the following templates. The template which features most within the cell is the predominant axial slant.

<table>
<thead>
<tr>
<th>0 0 0</th>
<th>0 1 0</th>
<th>0 0 0</th>
<th>0 0 1</th>
<th>1 0 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0</td>
<td>0 x 0</td>
<td>1 x 1</td>
<td>0 x 0</td>
<td>0 x 0</td>
</tr>
<tr>
<td>0 0 0</td>
<td>0 1 0</td>
<td>0 0 0</td>
<td>1 0 0</td>
<td>0 0 1</td>
</tr>
</tbody>
</table>

Table 3 - Predominant Axial Slant Templates

Figure 26 is an example showing the predominant axial slant values of the skeleton image (See Figure 26)

3.3.3 **Texture Features**

According to Tamura *et al* in [72], texture is regarded to what constitutes a macroscopic region. The structure is based upon the repetitive patterns where elements or primitives are arranged according to a placement rule.

Usually, texture analysis is performed on grey-level images. A grey-level co-occurrence matrix is one method of capturing spatial dependence of image grey-level value contributing to the perception of texture. A grey-level co-occurrence matrix is defined by first specifying a displacement vector $\vec{d} = (dx, dy)$, and counting all pairs of pixels separated by $\vec{d}$ having grey level values $i$ and $j$.

However, in this study, the signature image is binary so that only two colours are involved. Hence, the co-occurrence matrix is a $2 \times 2$ matrix which describes the transition of black and white pixels [27]. The co-occurrence matrix $P_{d}[i,j]$ for a binary image is defined as

$$P_{d}[i,j] = \begin{bmatrix} p_{00} & p_{01} \\ p_{10} & p_{11} \end{bmatrix} \quad \text{(Eq. 7)}$$
where \( p_{00} \) is the number of times a two white pixels, separated by \( \vec{d} \), occur. \( p_{01} \) is the number of times a white pixel is followed by a black separated by \( \vec{d} \). Similarly, \( p_{10} \) is the number of times that a black pixel is followed by a white pixel separated by \( \vec{d} \), and \( p_{11} \) is the number of times two black pixels separated by \( \vec{d} \), occur.

For instance, the co-occurrence matrices of the binary image, depicted in Figure 27, is given as

\[
\begin{bmatrix}
0 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 \\
1 & 0 & 0 & 0 & 1 \\
\end{bmatrix}
\]

![Figure 27 - Texture Co-Occurrence Matrix for a Binary Image](image)

For instance, the co-occurrence matrices of the binary image, depicted in Figure 27, is given as

\[
\begin{align*}
P_{\vec{d}[1,0]} &= \begin{bmatrix} 8 & 8 \\ 8 & 0 \end{bmatrix} \\
P_{\vec{d}[1,1]} &= \begin{bmatrix} 10 & 3 \\ 3 & 4 \end{bmatrix} \\
P_{\vec{d}[0,1]} &= \begin{bmatrix} 9 & 8 \\ 7 & 1 \end{bmatrix} \\
P_{\vec{d}[-1,1]} &= \begin{bmatrix} 9 & 4 \\ 2 & 4 \end{bmatrix}
\end{align*}
\]

The study [7] performed by Baltzakis and Papamarkos, also makes use of signature texture features. In fact, Baltzakis and Papamarkos divided the binary signature image into 6 segments (a grid of \( 3 \times 2 \) cells), where for each cell, the four co-occurrence matrices are calculated. However, the co-occurrence matrix used in [7] consisted of only \( p_{01} \) and \( p_{11} \) elements representing the changes from white to black and black to black pixels. In general, white to black (\( p_{01} \)) and black to white changes (\( p_{10} \)), should be equal in an infinite space. However, in a finite space, the exception lies in the border black pixels. On the other hand \( p_{00} \) is not considered as it represents the white background of the signature image.

In our study, texture features are extracted from the skeleton image using a superimposed grid of 96 \((12 \times 8)\) segments. This process sums up to 768 features \((96 \text{ segments} \times 4 \text{ matrices} \times 2 \text{ elements})\). Figure 28 shows the four co-occurrence matrices \( P_{\vec{d}[i,j]} = [p_{01}, p_{11}] \) of a segment applied on a skeleton signature image.

![Figure 28 - Texture Features Example](image)
3.4 Classification

3.4.1 RBF Neural Network

An RBF Neural Network is very well-known for the robustness in eliminating outliers. In fact, in study [42], Liu and Gader found that RBFs, in general, perform better than MLPs. An RBFNN is also popular for the relatively simple computation due to single layer architecture. RBFNNs are usually applied for Human Expression and Face Classifications.

From a large class of radial-basis functions covered by Micchelli’s theorem, it was decided to apply Gaussian functions in the hidden layer of the RBF architecture modelled in Figure 29.

\[
\phi(r) = \exp\left(-\frac{r^2}{2\sigma^2}\right) \text{ for some } \sigma > 0 \text{ and } r \in \mathbb{R} \quad (Eq. 8)
\]

An RBF neural network is dependent upon three arbitrary parameters including weights, centres, and spreads. The centres and respective spreads are directly related to the Gaussian function, where the centre is the representative data point of a cluster, and the spread represent the shape of the Gaussian function indicating the selectivity of the neuron.
Haykin [28] describes the following three different learning strategies for training an RBF neural network.

1. Fixed Centres Selected at Random
2. Self-Organised Selection of Centres
3. Supervised Selection of Centres

The first strategy is not considered for this study as it requires that the sample data be distributed in a representative manner. Due to high intra-personal variations a signature model cannot be represented by a single random signature.

The third strategy requires supervised training of the weights, centres and spreads. Since the supervised training for the three parameters is very slow, it was decided to opt for the second strategy. The second strategy, self-organised selection of centres, involves training the centres and spreads by different unsupervised algorithms and then training the weights separately.

The K-Means clustering algorithm was chosen as an unsupervised training algorithm to find the centres of the training data. It determines which points belong to which clusters, as well as the centres of the clusters. The following is a pseudo-code for the K-Means algorithm:

1. Initialise \( k \) centres by an equation which divides the sample space for each dimension into equal parts depending upon the value of \( k \).
2. For each data point determines which centre is closest. This determines each point’s cluster for the current iteration.
3. Compute the centroid (mean) of the points in each cluster. Set these the centres of the next iteration
4. Repeat steps 2 and 3 until centres do not differ appreciably from their previous value or until a given value.

Once the centres of the clusters have been calculated, the respective spreads are easily computed by:

\[
\sigma^2 = \frac{1}{n} \sum_{j=1}^{n} \left( \bar{x}_j - \bar{\mu} \right)^2
\]

(Eq. 9)

where \( \left\| \bar{x}_j - \bar{\mu} \right\| \) is the Euclidean distance calculated as

\[
\left\| \bar{x}_j - \bar{\mu} \right\| = \sqrt{\sum_{j=1}^{m} \left( x_{ij} - \mu_j \right)^2}
\]

Hence, (Eq. 9) can be simplified to

\[
\sigma^2 = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{m} \left( x_{ij} - \mu_j \right)^2
\]

(Eq. 10)

where \( m \) is the dimension of each data point \( n \) is the number of data points in the cluster and \( \mu \) is the centre of the cluster.
After some analysis, it was decided that a signature model be represented by a single
centre. This means that each signature model is equivalent to a single cluster and
hence represented by one centre. Therefore, the centre of a signature model is set to
be as the mean of all signature samples.

The radial basis function is given by

\[
\begin{bmatrix}
\varphi_{11} & \varphi_{12} & \ldots & \varphi_{1N} \\
\varphi_{21} & \varphi_{22} & \ldots & \varphi_{2N} \\
\vdots & \vdots & \ddots & \vdots \\
\varphi_{M1} & \varphi_{M2} & \ldots & \varphi_{MN}
\end{bmatrix}
\begin{bmatrix}
w_1 \\
w_2 \\
\vdots \\
w_M
\end{bmatrix}
= 
\begin{bmatrix}
d_1 \\
d_2 \\
\vdots \\
d_M
\end{bmatrix}
\]  
(Eq. 11)

where \( \varphi_{ij} \) is computed by (Eq. 8) in which the mean and spread are found by the
above mentioned method. The matrix is called the interpolation matrix, where the
number of columns represents the number of centres and the number of rows
represents the dimension size of each data point. The last column represents the bias
vector which is set to 1.

(Eq. 11) may be rewritten as

\[ \Phi \vec{w} = \vec{d} \]  
(Eq. 12)

where \( \Phi \) is the interpolation matrix where, \( \vec{w} \) is the weight vector and \( \vec{d} \) is the desired
response vector.

Once the interpolation matrix is constructed, the remaining parameter is the weight
vector. The weight values that minimise the error \( \Phi \vec{w} - \vec{d} \) can be obtained using a
pseudo-inverse technique. Hence the weight vector is computed by

\[ \vec{w} = (\Phi^T \Phi)^{-1} \Phi^T \vec{d} \]  
(Eq. 13)

### 3.4.2 Training

The following sub-sections describe the normalization and vector quantization
techniques for the respective features before being fed to the RBF neural network.

#### 3.4.2.1 Global Features

Since the global features (See Section 3.3.1) are used to calculate different
measurements of the signature, normalization is required to eliminate the units of the
global features. For instance, considering the pure width and baseline shift features,
their values cannot be compared or related since they have different units.

For this reason, all global features are normalised within the range \([0, 1]\).
Normalization is done by dividing each feature value with the highest value of the
same feature. Since the value of the global features namely, global and local angles,
can vary from -45° to 45°, normalization is done after shifting positively all angles by 45°, in order to eliminate negative values.

After normalization, the global features are fed to the RBF neural network depicted in Figure 29, and are trained to calculate the weight vector using the pseudo-inverse method as shown in (Eq. 13).

### 3.4.2.2 Grid Features

As discussed in Section 3.3.2, the grid features are calculated on a virtual grid of 12 × 8 segments superimposed on the skeleton signature image. Vector Quantization (VQ) technique based upon the K-Means algorithm is used to cluster each column vector. In this process, the column vectors of all signatures involved in the training are clustered using the unsupervised K-Means algorithm.

In this case, the first decision to be taken deals with the number of codebooks (clusters) to be adopted. As suggested by Justino et al [37], since the training database works with a small training vectors (40 specimens per author, the learning and verification ones), it was decided to use only one codebook for all the authors. On the other hand, different codebooks were used for each grid feature; pixel density, predominant axial slant and pixel distribution.

The second decision is to define the number of required codewords (centroids). As mentioned in [19] (cited by [37]), it is desirable for each symbol or codeword to be represented in the training set by at least two to five times the number of vector components used in clustering. A grid of 12 × 8 segments produces 12 column vectors with 8 components each for every signature sample. After some pilot studies it was decided that 50 codewords will be used to cluster the training data.

An alternative to this method is to find the number of codewords using the cluster validity measure proposed by Ray and Turi in [61]. The following is a pseudo-code of the suggested algorithm:

- Start by clustering all column vectors into two clusters
- Calculate the intra-cluster distance measure representing the clusters compactness as
  \[
  \text{intra} = \frac{1}{N} \sum_{i=1}^{K} \sum_{x \in C_i} \| x - z_i \|^2
  \]  (Eq. 14)
  where \( N \) is the number of column vectors, \( K \) is the number of clusters and \( z_i \) is the cluster centre of cluster \( C_i \). This value must be minimised.

- Calculate the inter-cluster distance measure representing the distance between clusters as
  \[
  \text{inter} = \min_{i=1,2,...,K-1} \min_{j=i+1,...,K} \left\{ \| z_i - z_j \|^2 \right\}
  \]  \( i = 1,2,...,K-1 \) and \( j = i+1,...,K \)  (Eq. 15)
  The minimum value must be maximised.
• Calculate the ratio of validity defined as:

$$\text{validity} = \frac{\text{intra}}{\text{inter}}$$ (Eq. 16)

• Split the cluster having maximum variance so that the K-Means algorithm is given good starting cluster centres. Calculate the average variance of the data points of each cluster defined as:

$$\sigma_i^2 = \frac{1}{M} \sum_{j=1}^{M} \sigma_{ij}^2$$ where M is the dimension size and $i = 1, 2, \ldots, K$ (Eq. 17)

• The cluster with the maximum average variance defined above is split into two clusters. Let $C_i$ be the cluster to split whose centre is $z_i$, then the two new cluster centres are calculated as:

$$z_i' = (z_{i1} - a_1, z_{i2} - a_2, \ldots, z_{im} - a_m)$$ (Eq. 18)

$$z_i'' = (z_{i1} + a_1, z_{i2} + a_2, \ldots, z_{im} + a_m)$$ (Eq. 19)

where $m$ is the dimension size and $a_1, a_2, \ldots, a_m$ are constants defined as

$$a_j = \frac{1}{2} \left( \min \left[ z_{ij} - \min_j, z_{ij} - \max_j \right] \right)$$ (Eq. 20)

• Repeat steps 2 to 6 for a given number of iterations $K_{\text{max}}$. The optimal number of clusters $k$ is the one having the minimum validity value (See (Eq. 16)).

With the above method, the optimal number of clusters is defined and the column vectors are clustered accordingly.

The two mentioned methods will both be used in Chapter 4 to analyse and compare the respective performance results.

During the training phase, each column vector is therefore replaced by the centroid of the respective cluster. The representative centroid vectors of each image (12 vectors) are then merged as one feature vector. For instance, the pixel density feature, extracted from a grid of $12 \times 8$ segments, produces a feature vector of 96 elements.

Vector quantization is processed for each grid feature discussed in Section 3.3.2. Since the distribution grid feature combined of four sensors is considered as four different grid features, a total of six vector quantization processes are executed. These six processes produce six feature vectors each containing 96 elements. Finally, the 6 vectors representing the 6 grid features are normalised within the range [0, 1] and subsequently merged together in order to produce one combined feature vector of 576 ($6 \times 96$) elements.
So, each skeleton image, which is superimposed by a $12 \times 8$ grid, produces a feature vector of 576 elements which is fed to the RBFNN as depicted in Figure 30.

![Figure 30 - Training with Grid Features](image)

### 3.4.2.3 Texture Features

Similarly to grid features, Vector Quantization is also adopted to cluster the columns of all skeleton signature images used for training. As explained in Section 3.3.3, texture features extract information about the changes from white to black pixels and from black to black pixels in 4 different direction vectors. So, recalling from Section 3.3.3, each skeleton signature image with a grid of $12 \times 8$ segments contains 96 texture values. This means that each cell contains 8 texture values and hence each column contains 64 texture values.

However, unlike to grid features, in this case there is only one feature vector rather than 6 feature vectors. So, using both alternative methodologies explained in Section 3.4.2.2, after vector quantization, each texture feature column vector is replaced with the centroid (codeword) of the respective cluster. After normalizing the result column vectors within the range $[0, 1]$, the column vectors of each signature are merged in order to produce one feature vector of 768 elements ($64 \times 12$). The resultant feature vectors are used as inputs to the RBF neural network in order to train the model.

### 3.5 Conclusion

This chapter explained in detail the adopted methodology used to implement an offline handwritten signature verification system. Various algorithms (See Appendix F) were implemented together with other processing techniques in order to process, extract information and classify signature images.

The following chapter shows the implementation of the mentioned methodology and publishes the results obtained by various training strategies.
Chapter 4  Implementation and Results
4.1 Introduction

This chapter is divided into two sections being

1. Implementation of the application and
2. Training and testing of the proposed RBF architecture for offline handwritten signature verification.

The signature database gathered for this study consists of 2498 genuine signature samples provided by 65 authors. This means that the system can only be tested for random signature forgeries since no skilled and simple forgeries were available.

4.2 Software Tools

Since a lot of imaging operations were found to be required for this study, an amount of time was invested to choose a robust product to perform the required imaging operations. After some research, MATLAB and Java Advanced Imaging (JAI) were found to outperform in this field. However, it was decided that JAI will be the main tool to perform imaging operations. The selection of JAI was motivated by the fact that the author of this project is familiar with Java programming language. JAI provides a set of object-oriented interfaces that support a simple, high-level programming model which allows the manipulation of images easily. A textbook of Rodrigues [62] was used as a reference guide to implement imaging operations using java technology.

Java Matrix (JAMA) and Java Excel (JExcelAPI) APIs were also used. These APIs include a set of object-oriented interfaces supporting matrices operations and read/write operations in excel sheets respectively. JAMA was required to perform matrix operations to calculate the pseudo-inverse (See (Eq. 13)) during training of the RBF neural network, while JExcelAPI was used during training and testing processes for writing the respective results to excel sheets for further statistical analysis.

Furthermore, a personal Oracle relational database management system (RDBMS) was used to maintain the signature models and other information of each author. JDeveloper was the preferred integrated development environment (IDE) for the implementation of Java.

4.2.1 Software Tools Summary

Table 4 shows a summary of software components used in this project.

<table>
<thead>
<tr>
<th>Software Component</th>
<th>Name</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating System</td>
<td>Windows</td>
<td>XP</td>
</tr>
<tr>
<td>Programming Language</td>
<td>Java Development Kit</td>
<td>1.5</td>
</tr>
<tr>
<td>Imaging API</td>
<td>JAI</td>
<td>1.1.2_01</td>
</tr>
<tr>
<td>Java Excel API</td>
<td>JExcelAPI</td>
<td>N/A</td>
</tr>
<tr>
<td>Java Matrix Package</td>
<td>JAMA</td>
<td>1.0.2</td>
</tr>
<tr>
<td>Java IDE</td>
<td>JDeveloper</td>
<td>10.1.3</td>
</tr>
<tr>
<td>Personal Database RDBMS</td>
<td>Oracle</td>
<td>10G</td>
</tr>
</tbody>
</table>

Table 4 - Software Tools Used
4.3 Implementation

This section describes the processes carried out in all stages of the implemented solution.

A graphical interface was implemented in order to input and maintain authors’ information and the respective signature model. The application is also used to view the signature samples of each author in three different modes; original, binary and skeleton image. Appendix H explains the contents of the CD which is also submitted as part of this project.

4.3.1 Main Window

Figure 31 shows the main screen of the application. Personal details (marked in red) were removed from the image to protect author’s details.

Functionality

1. The main menu is used to retrieve, insert, update and delete author details
2. Opens a new dialog window to search for a specific author shown in Figure 32
3. A sub menu used to add, delete or delete all signature samples of an author without effecting the author’s personal information
4. Another sub menu used to view the respective signature samples in different modes.
5. An original signature sample
6. The respective normalised and binarised signature sample
7. The respective thinned signature sample
4.3.2 Search Author Dialog

This screen (See Figure 32) is used to search for a particular author based on three attributes combined with logical operators. The result is displayed in the bottom part where the user is enabled to select only one row (author) and press OK. The selected author together with the respective signature model will be loaded accordingly in the main window (See Figure 31).

![Figure 32 - Graphical User Interface - Search Author](image)

4.3.3 Signature Acquisition and Pre-Processing Processes

The signature model of each author is entered into the system by using the main window (See Figure 31). All original 24-bit colour signature samples are stored on the hard drive (after being scanned) and are then loaded into the database on pressing INSERT button from the main menu.

The insert process includes storing author’s personal information together with the pre-processing of the sample model explained in Section 3.2.

4.3.4 Feature Extraction Process

Since the process of feature extraction takes quite a long time due to the numerous computations required, it was decided to execute this process as a batch stand alone program. Appendix Error! Reference source not found. includes the java source used to run feature extraction process (See Section 3.3) for all signature models stored in the database.

4.4 Training and Testing Protocol

As explained in Chapter 3, the system made use of three groups of signature features, namely global, grid and texture features. Moreover, the sample data acquired from a
number of authors (See Section 3.1) was split into two groups of signatures; those signed within a frame and those without a frame. The different groups of features and different types of signature samples available allow the system to be trained and tested in several ways.

Several training/testing strategies were performed on the acquired signature database. Each training/testing strategy was validated by splitting the available data into two parts, one part for training and the other part for testing. Since the number of signatures for each author is not low (40 signature samples per author), this validation was considered to be sufficient, and further cross-validation was not performed.

The training included a combination of the two types of signature samples and the three groups of signature features. Each signature model, representing the signature samples of one author, is trained and tested with the following three scenarios, for each combination of signature features, including:

1. Training and testing only samples without a frame (TNTN)
2. Training and Testing samples both with frame and without frame (TNTA)
3. Training signature without a frame and testing all signatures (TATA)

After several pilot studies, it was decided that a ratio of 3:5 will be used to train and test a signature model in the first two strategies mentioned above.

For instance, in the first strategy, a signature model containing 25 non-framed signature samples, the system is trained with 15 random samples, while it is tested with the other 9 together with all non-framed signature samples of the other authors. Furthermore, in the second strategy, a signature model composed of 25 non-framed samples and 15 framed samples, 15 random signatures are randomly selected from the non-framed signature samples while 9 samples are randomly selected from the 15 framed samples. On the other hand, the third strategy test the system for robustness, where from a signature model of 40 samples (25 non-framed, 15 framed), the system is only trained with 15 non-framed signature samples and tested with the rest 25 samples amongst other author’s signature samples.

As explained above, the training process includes a priori knowledge of signature forgeries. This means that each signature model is trained with the genuine and forgery signatures in order to classify the two classes accordingly.

The following combination of signature features is used to train and test the proposed system and subsequently evaluate and analyse the performance in terms of False Acceptance Rate (FAR), False Rejection Rate (FRR) and Total Error Rate (TER) and Mean Error Rate (MER).

1. Global features only 4. Global and Grid features
2. Grid features only 5. Global and Texture features
3. Texture features only 6. Grid and Texture features
7. Global, Grid and Texture features

As mentioned in Section 3.4.2 both grid and texture features use two alternatives of vector quantization (VQ); finding the number of codewords by using the validity
measure and setting a fixed number of 50 codewords. Vector quantization is used by the last 6 of the above list of signature features. Hence the system will be trained and tested in 39 \((7 \times 3) + (6 \times 3)\) different strategies.

4.4.1 Performance Measurement Results

The training and testing steps are implemented as one process where an RBFNN is first trained for a given author and subsequently tested with other signature samples of the same author and with signature samples of the other authors. This process creates an Excel workbook containing the RBFNN classification results for both the training and testing samples, besides additional process results as explained further down.

Additionally, this process creates another Excel workbook, consisting of the following two sheets:

1. Summary data table containing the best performance results of each author
2. Average Testing Receiver Operating Characteristic (ROC) curves.

Figure 33 shows two snapshots of a sample workbook generated for a particular testing scenario.

The above Excel sheets are required to calculate the performance measurements in terms of FAR, FRR and TER for each scenario defined as:

1. FRR, also referred to as type I error, is the ratio of falsely rejected signature samples from the total number of tested genuine signatures [32].
2. FAR, also referred to as type II error, is the ratio of falsely accepted signatures from the total number of tested random signature forgeries [32].
3. TER is the summation of FRR and FAR.
4. MER is the average rate of FRR and FAR.
The performance results for each verification scenario, explained in Section 4.4, are given in the following two formats:

1. A table containing statistical results
2. A line chart showing both FAR and FRR together with TER

When grid or texture features are involved, the performance results were given both for fixed size codebook vector quantisation (50 codewords) and for adaptively sized codebook vector quantisation. The above mentioned formats are explained in detail in the next sub-sections.

### 4.4.1.1 Performance Results Table

The following table defines is a template used to display the performance results of each testing scenario.

<table>
<thead>
<tr>
<th>Cases that should be accepted</th>
<th>The total number of test samples that must be classified as genuine</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cases that should be rejected</td>
<td>The total number of test samples that must be classified as forgeries</td>
</tr>
<tr>
<td>Accepted</td>
<td>The total number of test samples that are actually accepted consisting of the correct acceptances and the false acceptances</td>
</tr>
<tr>
<td>Rejected</td>
<td>The total number of test samples that are actually rejected consisting of the correct rejections and the false rejections</td>
</tr>
<tr>
<td>Correct Acceptances</td>
<td>The total number of test samples that are actually correctly accepted. A percentage of the correct acceptances is given in the brackets</td>
</tr>
<tr>
<td>False Rejections</td>
<td>The total number of test samples that are falsely rejected. This means they must have been classified as genuine but were classified as forgeries. The False Rejection Rate (FRR) is given in the brackets</td>
</tr>
<tr>
<td>Correct Rejections</td>
<td>The total number of test samples that are correctly rejected. A percentage of the correct rejections is given in the brackets</td>
</tr>
<tr>
<td>False Acceptances</td>
<td>The total number of test samples that are falsely accepted. This means that they must have been classified as forgery but were classified as genuine. The False Acceptance Rate (FAR) is given in the brackets</td>
</tr>
<tr>
<td>Total Error Rate (TER)</td>
<td>The summation of the FRR and the FAR</td>
</tr>
<tr>
<td>Mean Error Rate (MER)</td>
<td>The average rate of FRR and FAR</td>
</tr>
</tbody>
</table>

**Table 5 - Performance Summary Results Template**

### 4.4.1.2 Performance Results Chart

For each signature model, a Receiver Operating Characteristic (ROC) curve is created to find the operating point producing the lowest TER, FRR and FAR respectively. The ROC data is created by varying the threshold $\alpha$ in steps of 0.025 and time calculating the FAR and FRR respectively. The optimal $\alpha$ is the one producing the lowest FAR and FRR.

Finally, a summary data table is created containing 65 rows, one for each author, where each row is composed of the author id, average threshold (as explained above), together with the respective performance measurements; FAR, FRR and TER. The data table is sorted in ascending order of TER. Subsequently, a summary chart is created from this summary data table showing three performance measurements FAR,
FRR and TER. The scope of this chart is to show how the performance measurement data is distributed.
4.5 Training and Testing Results

4.5.1 Testing with Global Features

The following scenarios are used to measure the performance of the RBF neural network using global features (See Section 3.3.1). The following three test scenarios show the strength of global features (16 features) to classify and recognize signature samples.

4.5.1.1 Train Non-Frame Test Non-Frame (TNTN)

This scenario considers only signature samples without a frame. The training phase, as explained in Section 4.4, consists of 60% of each signature model and then tested with the remaining 40% of the same model together with all other signature models.

| Training Signature Samples: 941 |
| Testing Signature Samples: 631 |

| Cases that should be accepted | 631 |
| Cases that should be rejected | 40384 |
| Accepted                     | 2008 |
| Rejected                     | 39007 |
| Correct Acceptances          | 609 (96.51%) |
| False Rejections             | 22 (3.49%) |
| Correct Rejections           | 38985 (96.54%) |
| False Acceptances            | 1399 (3.46%) |
| Total Error Rate (TER)       | 6.95% |
| Mean Error Rate (MER)        | 3.475% |

Table 6 - Results: Global Features - TNTN

The following figure shows a line chart displaying the FAR, FRR and TER performance measures. In this scenario the FRR and FAR are very close and they differ by only 0.03%. It shows that 8 signature models maintained a TER of 0%, whereas 47 signature models maintained an FRR of 0%. The TER was incremented by around 15 signature models.
4.5.1.2 Train Non-Frame Test All (TNTA)

This test scenario is used to test the robustness of the RBF neural network. In this case, the RBF is trained with signature samples that were collected without a frame and is tested with signature samples that were collected both with a frame and without a frame.

*Training Signature Samples:* 941  
*Testing Signature Samples:* 1557

<table>
<thead>
<tr>
<th>Cases that should be accepted</th>
<th>1557</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cases that should be rejected</td>
<td>99648</td>
</tr>
<tr>
<td>Accepted</td>
<td>7206</td>
</tr>
<tr>
<td>Rejected</td>
<td>93999</td>
</tr>
<tr>
<td>Correct Acceptances</td>
<td>1469 (94.35%)</td>
</tr>
<tr>
<td>False Rejections</td>
<td>88 (5.65%)</td>
</tr>
<tr>
<td>Correct Rejections</td>
<td>93911 (94.24%)</td>
</tr>
<tr>
<td>False Acceptances</td>
<td>5737 (5.76%)</td>
</tr>
<tr>
<td>Total Error Rate (TER)</td>
<td>11.41%</td>
</tr>
<tr>
<td>Mean Error Rate (MER)</td>
<td>5.705%</td>
</tr>
</tbody>
</table>

Table 7 - Results: Global Features - TNTA

The following figure shows a line chart displaying the FAR, FRR and TER performance measures for each author. As expected, this scenario performed less with respect to the previous one. The reason is that the RBF networks were only trained with signature samples that were collected without a frame and then tested with all signature samples. In fact there were only two authors that were totally recognised.

Figure 35 - Results: Global Features - TNTA
4.5.1.3 Train All Test All (TATA)

In this testing scenario, the RBF networks are trained with signature samples both with frame and without frame.

*Training Signature Samples: 1495*

*Testing Signature Samples: 1003*

<table>
<thead>
<tr>
<th>Cases that should be accepted</th>
<th>1003</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cases that should be rejected</td>
<td>64192</td>
</tr>
<tr>
<td>Accepted</td>
<td>3980</td>
</tr>
<tr>
<td>Rejected</td>
<td>61215</td>
</tr>
<tr>
<td>Correct Acceptances</td>
<td>973  (97.01%)</td>
</tr>
<tr>
<td>False Rejections</td>
<td>30   (2.99%)</td>
</tr>
<tr>
<td>Correct Rejections</td>
<td>61185 (95.32%)</td>
</tr>
<tr>
<td>False Acceptances</td>
<td>3007 (4.68%)</td>
</tr>
<tr>
<td>Total Error Rate (TER)</td>
<td>7.67%</td>
</tr>
<tr>
<td>Mean Error Rate (MER)</td>
<td>3.835%</td>
</tr>
</tbody>
</table>

**Table 8 - Results: Global Feature - TATA**

This testing scenario maintained a better FRR than the previous scenario. The main reason is that the training process consisted of framed and non-framed signature samples. In fact, a total of 45 authors maintained an FRR of 0%, but there were only two authors that maintained an FAR of 0%. The worst signature model is the one of author 4 where an FAR of 26.6% is achieved. Looking at the signature images of author 4, it can be easily noticed the high intra personal variability of the author.
4.5.2 Testing with Grid Features

The objective of the following three scenarios is to analyse the performance of the RBF neural network when training and testing with grid features consisting 576 features (See Section 3.3.2).

4.5.2.1 Train Non-Frame Test Non-Frame (TNTN)

*Training Signature Samples: 941*

*Testing Signature Samples: 631*

<table>
<thead>
<tr>
<th>Performance Results</th>
<th>Adaptively Sized VQ Codebook</th>
<th>Fixed Size (50) VQ Codebook</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cases that should be accepted</td>
<td>631</td>
<td>631</td>
</tr>
<tr>
<td>Cases that should be rejected</td>
<td>40384</td>
<td>40384</td>
</tr>
<tr>
<td>Accepted</td>
<td>2120</td>
<td>1620</td>
</tr>
<tr>
<td>Rejected</td>
<td>38895</td>
<td>39395</td>
</tr>
<tr>
<td>Correct Acceptances</td>
<td>591 (93.66%)</td>
<td>618 (97.94%)</td>
</tr>
<tr>
<td>False Rejections</td>
<td>40 (6.34%)</td>
<td>13 (2.06%)</td>
</tr>
<tr>
<td>Correct Rejections</td>
<td>38855 (96.21%)</td>
<td>39382 (97.5%)</td>
</tr>
<tr>
<td>False Acceptances</td>
<td>1529 (3.79%)</td>
<td>1002 (2.48%)</td>
</tr>
<tr>
<td>Total Error Rate (TER)</td>
<td><strong>10.13%</strong></td>
<td><strong>4.54%</strong></td>
</tr>
<tr>
<td>Mean Error Rate (MER)</td>
<td><strong>5.065%</strong></td>
<td><strong>2.27%</strong></td>
</tr>
</tbody>
</table>

Table 9 - Results: Grid Features - TNTN

![Figure 37 - Results: Grid Features - TNTN - Adaptively Sized VQ Codebook](image)

![Figure 38 - Results: Grid Features - TNTN - Fixed Size (50) VQ Codebook](image)
4.5.2.2 Train Non-Frame Test All (TNTA)

Training Signature Samples: 941
Testing Signature Samples: 1557

<table>
<thead>
<tr>
<th>Performance Results</th>
<th>Adapatively Sized VQ Codebook</th>
<th>Fixed Size (50) VQ Codebook</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cases that should be accepted</td>
<td>1557</td>
<td>1557</td>
</tr>
<tr>
<td>Cases that should be rejected</td>
<td>99648</td>
<td>99648</td>
</tr>
<tr>
<td>Accepted</td>
<td>6280</td>
<td>99706</td>
</tr>
<tr>
<td>Rejected</td>
<td>94925</td>
<td>96303</td>
</tr>
<tr>
<td>Correct Acceptances</td>
<td>1426 (91.59%)</td>
<td>1499 (96.27%)</td>
</tr>
<tr>
<td>False Rejections</td>
<td>131 (8.41%)</td>
<td>58 (3.73%)</td>
</tr>
<tr>
<td>Correct Rejections</td>
<td>94794 (95.13%)</td>
<td>96245 (96.58%)</td>
</tr>
<tr>
<td>False Acceptances</td>
<td>4854 (4.87%)</td>
<td>3403 (3.42%)</td>
</tr>
<tr>
<td>Total Error Rate (TER)</td>
<td>13.28%</td>
<td>7.15%</td>
</tr>
<tr>
<td>Mean Error Rate (MER)</td>
<td>6.64%</td>
<td>3.575%</td>
</tr>
</tbody>
</table>

Table 10 - Results: Grid Features - TNTA

Figure 39 - Results: Grid Features - TNTA - Adaptively Sized VQ Codebook

Figure 40 - Results: Grid Features - TNTA - Fixed Size (50) VQ Codebook
4.5.2.3 Train All Test All (TATA)

Training Signature Samples: 1495
Testing Signature Samples: 1003

<table>
<thead>
<tr>
<th>Performance Results</th>
<th>Adapтивely Sized VQ Codebook</th>
<th>Fixed Size (50) VQ Codebook</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cases that should be accepted</td>
<td>1003</td>
<td>1003</td>
</tr>
<tr>
<td>Cases that should be rejected</td>
<td>64192</td>
<td>64192</td>
</tr>
<tr>
<td>Accepted</td>
<td>4205</td>
<td>2519</td>
</tr>
<tr>
<td>Rejected</td>
<td>60990</td>
<td>62676</td>
</tr>
<tr>
<td>Correct Acceptances</td>
<td>959 (95.61%)</td>
<td>981 (97.81%)</td>
</tr>
<tr>
<td>False Rejections</td>
<td>44 (4.39%)</td>
<td>22 (2.19%)</td>
</tr>
<tr>
<td>Correct Rejections</td>
<td>60946 (94.94%)</td>
<td>62654 (97.6%)</td>
</tr>
<tr>
<td>False Acceptances</td>
<td>3246 (5.06%)</td>
<td>1538 (2.4%)</td>
</tr>
<tr>
<td>Total Error Rate (TER)</td>
<td>9.45%</td>
<td>4.59%</td>
</tr>
<tr>
<td>Mean Error Rate (MER)</td>
<td>4.725%</td>
<td>2.295%</td>
</tr>
</tbody>
</table>

Table 11 - Results: Grid Features - TATA

![Performance Results Grid Features - TATA - Adapтивely Sized VQ Codebook](image1)

Figure 41 - Results: Grid Features - TATA - Adapтивely Sized VQ Codebook

![Performance Results Grid Features - TATA - Fixed Size (50) VQ Codebook](image2)

Figure 42 - Results: Grid Features - TATA - Fixed Size (50) VQ Codebook
4.5.3 Testing with Texture Features

The following three testing scenarios are used to analyse the performance of the RBF neural networks using 768 texture features (See Section 3.3.3).

4.5.3.1 Train Non-Frame Test Non-Frame (TNTA)

*Training Signature Samples: 941*
*Testing Signature Samples: 631*

<table>
<thead>
<tr>
<th>Performance Results</th>
<th>Adaptively Sized VQ Codebook</th>
<th>Fixed Size (50) VQ Codebook</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cases that should be accepted</td>
<td>631</td>
<td>631</td>
</tr>
<tr>
<td>Cases that should be rejected</td>
<td>40384</td>
<td>40384</td>
</tr>
<tr>
<td>Accepted</td>
<td>5206</td>
<td>1951</td>
</tr>
<tr>
<td>Rejected</td>
<td>35809</td>
<td>39064</td>
</tr>
<tr>
<td>Correct Acceptances</td>
<td>531 (84.15%)</td>
<td>600 (95.09%)</td>
</tr>
<tr>
<td>False Rejections</td>
<td>100 (15.85%)</td>
<td>31 (4.91%)</td>
</tr>
<tr>
<td>Correct Rejections</td>
<td>35710 (88.43%)</td>
<td>39033 (96.65%)</td>
</tr>
<tr>
<td>False Acceptances</td>
<td>4674 (11.57%)</td>
<td>1351 (3.35%)</td>
</tr>
<tr>
<td>Total Error Rate (TER)</td>
<td>27.42%</td>
<td>8.26%</td>
</tr>
<tr>
<td>Mean Error Rate (MER)</td>
<td>13.71%</td>
<td>4.13%</td>
</tr>
</tbody>
</table>

Table 12 - Results: Texture Features - TNTN

![Figure 43 - Results: Texture Features - TNTN - Adaptively Sized VQ Codebook](image)

![Figure 44 - Results: Texture Features - TNTN - Fixed Size (50) VQ Codebook](image)
4.5.3.2 Train Non-Frame Test All (TNTA)

Training Signature Samples: 941
Testing Signature Samples: 1557

<table>
<thead>
<tr>
<th>Performance Results</th>
<th>Adaptively Sized VQ Codebook</th>
<th>Fixed Size (50) VQ Codebook</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cases that should be accepted</td>
<td>1557</td>
<td>1557</td>
</tr>
<tr>
<td>Cases that should be rejected</td>
<td>99648</td>
<td>99648</td>
</tr>
<tr>
<td>Accepted</td>
<td>16425</td>
<td>6322</td>
</tr>
<tr>
<td>Rejected</td>
<td>84780</td>
<td>94883</td>
</tr>
<tr>
<td>Correct Acceptances</td>
<td>1302 (83.62%)</td>
<td>1449 (93.06%)</td>
</tr>
<tr>
<td>False Rejections</td>
<td>255 (16.38%)</td>
<td>108 (6.94%)</td>
</tr>
<tr>
<td>Correct Rejections</td>
<td>84524 (84.82%)</td>
<td>94776 (95.11%)</td>
</tr>
<tr>
<td>False Acceptances</td>
<td>15124 (15.18%)</td>
<td>4872 (4.89%)</td>
</tr>
<tr>
<td>Total Error Rate (TER)</td>
<td>31.56%</td>
<td>11.83%</td>
</tr>
<tr>
<td>Mean Error Rate (MER)</td>
<td>15.78%</td>
<td>5.915%</td>
</tr>
</tbody>
</table>

Table 13 - Results: Texture Features - TNTA

![Image of graph](image1)

Figure 45 - Results: Texture Features - TNTA - Adaptively Sized VQ Codebook

![Image of graph](image2)

Figure 46 - Results: Texture Features - TNTA - Fixed Size (50) VQ Codebook
4.5.3.3 Train All Test All (TATA)

Training Signature Samples: 1495
Testing Signature Samples: 1003

<table>
<thead>
<tr>
<th>Performance Results</th>
<th>Adaptively Sized VQ Codebook</th>
<th>Fixed Size (50) VQ Codebook</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cases that should be accepted</td>
<td>1003</td>
<td>1003</td>
</tr>
<tr>
<td>Cases that should be rejected</td>
<td>64192</td>
<td>64192</td>
</tr>
<tr>
<td>Accepted</td>
<td>7886</td>
<td>2970</td>
</tr>
<tr>
<td>Rejected</td>
<td>57309</td>
<td>62225</td>
</tr>
<tr>
<td>Correct Acceptances</td>
<td>850 (84.75%)</td>
<td>945 (94.22%)</td>
</tr>
<tr>
<td>False Rejections</td>
<td>153 (15.25%)</td>
<td>58 (5.78%)</td>
</tr>
<tr>
<td>Correct Rejections</td>
<td>57156 (89.04%)</td>
<td>62168 (96.85%)</td>
</tr>
<tr>
<td>False Acceptances</td>
<td>7036 (10.96%)</td>
<td>2024 (3.15%)</td>
</tr>
<tr>
<td>Total Error Rate (TER)</td>
<td>26.21%</td>
<td>8.93%</td>
</tr>
<tr>
<td>Mean Error Rate (MER)</td>
<td>13.105%</td>
<td>4.465%</td>
</tr>
</tbody>
</table>

Table 14 - Results: Texture Features - TATA

Figure 47 - Results: Texture Features - TATA - Adaptively Sized VQ Codebook

Figure 48 - Results: Texture Features - TATA - Fixed Size (50) VQ Codebook
4.5.4 Testing with Global and Grid Features

In the following scenarios the RBF neural network is trained and tested using a combination of global and grid features summing to 592 features.

4.5.4.1 Train Non-Frame Test Non-Frame (TNTN)

*Training Signature Samples: 941*
*Testing Signature Samples: 631*

<table>
<thead>
<tr>
<th>Performance Results</th>
<th>Adaptively Sized VQ Codebook</th>
<th>Fixed Size (50) VQ Codebook</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cases that should be accepted</td>
<td>631</td>
<td>631</td>
</tr>
<tr>
<td>Cases that should be rejected</td>
<td>40384</td>
<td>40384</td>
</tr>
<tr>
<td>Accepted</td>
<td>2126</td>
<td>1631</td>
</tr>
<tr>
<td>Rejected</td>
<td>38889</td>
<td>39384</td>
</tr>
<tr>
<td>Correct Acceptances</td>
<td>598 (94.77%)</td>
<td>621 (98.42%)</td>
</tr>
<tr>
<td>False Rejections</td>
<td>33 (5.23%)</td>
<td>10 (1.58%)</td>
</tr>
<tr>
<td>Correct Rejections</td>
<td>38857 (96.22%)</td>
<td>39375 (97.5%)</td>
</tr>
<tr>
<td>False Acceptances</td>
<td>1527 (3.78%)</td>
<td>1009 (2.5%)</td>
</tr>
<tr>
<td>Total Error Rate (TER)</td>
<td>9.01%</td>
<td>4.08%</td>
</tr>
<tr>
<td>Mean Error Rate (MER)</td>
<td>4.505%</td>
<td>2.04%</td>
</tr>
</tbody>
</table>

Table 15 - Results: Global & Grid Features - TNTN

![Figure 49 - Results: Global & Grid Features - TNTN - Adaptively Sized VQ Codebook](image)

![Figure 50 - Results: Global & Grid Features - TNTN - Fixed Size (50) VQ Codebook](image)
4.5.4.2 Train Non-Frame Test All (TNTA)

*Training Signature Samples: 941*
*Testing Signature Samples: 1557*

<table>
<thead>
<tr>
<th>Performance Results</th>
<th>Adapтивely Sized VQ Codebook</th>
<th>Fixed Size (50) VQ Codebook</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cases that should be accepted</td>
<td>1557</td>
<td>1557</td>
</tr>
<tr>
<td>Cases that should be rejected</td>
<td>99648</td>
<td>99648</td>
</tr>
<tr>
<td>Accepted</td>
<td>5924</td>
<td>4404</td>
</tr>
<tr>
<td>Rejected</td>
<td>95281</td>
<td>96801</td>
</tr>
<tr>
<td>Correct Acceptances</td>
<td>1442 (92.61%)</td>
<td>1504 (96.6%)</td>
</tr>
<tr>
<td>False Rejections</td>
<td>115 (7.39%)</td>
<td>53 (3.4%)</td>
</tr>
<tr>
<td>Correct Rejections</td>
<td>95166 (95.5%)</td>
<td>96749 (97.09%)</td>
</tr>
<tr>
<td>False Acceptances</td>
<td>4482 (4.5%)</td>
<td>2899 (2.91%)</td>
</tr>
<tr>
<td>Total Error Rate (TER)</td>
<td>11.89%</td>
<td>6.31%</td>
</tr>
<tr>
<td>Mean Error Rate (MER)</td>
<td>5.945%</td>
<td>3.155%</td>
</tr>
</tbody>
</table>

Table 16 - Results: Global & Grid Features - TNTA

Figure 51 - Results: Global & Grid Features - TNTA - Adapтивely Sized VQ Codebook

Figure 52 - Results: Global & Grid Features - TNTA - Fixed Size (50) VQ Codebook
4.5.4.3 Train All Test All (TATA)

Training Signature Samples: 1495
Testing Signature Samples: 1003

<table>
<thead>
<tr>
<th>Performance Results</th>
<th>Adapitively Sized VQ Codebook</th>
<th>Fixed Size (50) VQ Codebook</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cases that should be accepted</td>
<td>1003</td>
<td>1003</td>
</tr>
<tr>
<td>Cases that should be rejected</td>
<td>64192</td>
<td>64192</td>
</tr>
<tr>
<td>Accepted</td>
<td>3848</td>
<td>2815</td>
</tr>
<tr>
<td>Rejected</td>
<td>61347</td>
<td>62380</td>
</tr>
<tr>
<td>Correct Acceptances</td>
<td>967 (96.41%)</td>
<td>990 (98.7%)</td>
</tr>
<tr>
<td>False Rejections</td>
<td>36 (3.59%)</td>
<td>13 (1.3%)</td>
</tr>
<tr>
<td>Correct Rejections</td>
<td>61311 (95.51%)</td>
<td>62367 (97.16%)</td>
</tr>
<tr>
<td>False Acceptances</td>
<td>2881 (4.49%)</td>
<td>1825 (2.84%)</td>
</tr>
<tr>
<td>Total Error Rate (TER)</td>
<td>8.08%</td>
<td>4.14%</td>
</tr>
<tr>
<td>Mean Error Rate (MER)</td>
<td>4.04%</td>
<td>2.07%</td>
</tr>
</tbody>
</table>

Table 17 - Results: Global & Grid Features - TATA

Figure 53 - Results: Global & Grid Features - TATA - Adapitively Sized VQ Codebook

Figure 54 - Results: Global & Grid Features - TATA - Fixed Size (50) VQ Codebook
4.5.5 Testing with Global and Texture Features

The objective of the following three scenarios is to analyse the performance of the RBF neural network by combining global and texture features summing to 784 features.

4.5.5.1 Train Non-Frame Test Non-Frame (TNTN)

*Training Signature Samples: 941*
*Testing Signature Samples: 631*

<table>
<thead>
<tr>
<th>Performance Results</th>
<th>Adapively Sized VQ Codebook</th>
<th>Fixed Size (50) VQ Codebook</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cases that should be accepted</td>
<td>631</td>
<td>631</td>
</tr>
<tr>
<td>Cases that should be rejected</td>
<td>40384</td>
<td>40384</td>
</tr>
<tr>
<td>Accepted</td>
<td>4669</td>
<td>2012</td>
</tr>
<tr>
<td>Rejected</td>
<td>36346</td>
<td>39003</td>
</tr>
<tr>
<td>Correct Acceptances</td>
<td>579 (91.76%)</td>
<td>607 (96.2%)</td>
</tr>
<tr>
<td>False Rejections</td>
<td>52 (8.24%)</td>
<td>24 (3.8%)</td>
</tr>
<tr>
<td>Correct Rejections</td>
<td>36294 (89.87%)</td>
<td>38979 (96.52%)</td>
</tr>
<tr>
<td>False Acceptances</td>
<td>4090 (10.13%)</td>
<td>1405 (3.48%)</td>
</tr>
<tr>
<td>Total Error Rate (TER)</td>
<td>18.37%</td>
<td>7.28%</td>
</tr>
<tr>
<td>Mean Error Rate (MER)</td>
<td>9.185%</td>
<td>3.64%</td>
</tr>
</tbody>
</table>

Table 18 - Results: Global & Texture Features - TNTN

Figure 55 - Results: Global & Texture Features - TNTN - Adapively Sized VQ Codebook

Figure 56 - Results: Global & Texture Features - TNTN - Fixed Size (50) VQ Codebook
4.5.5.2 Train Non-Frame Test All (TNTA)

Training Signature Samples: 941
Testing Signature Samples: 1557

<table>
<thead>
<tr>
<th>Performance Results</th>
<th>Adaptively Sized VQ Codebook</th>
<th>Fixed Size (50) VQ Codebook</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cases that should be accepted</td>
<td>1557</td>
<td>1557</td>
</tr>
<tr>
<td>Cases that should be rejected</td>
<td>99648</td>
<td>99648</td>
</tr>
<tr>
<td>Accepted</td>
<td>11973</td>
<td>6217</td>
</tr>
<tr>
<td>Rejected</td>
<td>89232</td>
<td>94988</td>
</tr>
<tr>
<td>Correct Acceptances</td>
<td>1383 (88.82%)</td>
<td>1468 (94.28%)</td>
</tr>
<tr>
<td>False Rejections</td>
<td>174 (11.18%)</td>
<td>89 (5.72%)</td>
</tr>
<tr>
<td>Correct Rejections</td>
<td>89058 (89.37%)</td>
<td>94899 (95.23%)</td>
</tr>
<tr>
<td>False Acceptances</td>
<td>10590 (10.63%)</td>
<td>4749 (4.77%)</td>
</tr>
<tr>
<td>Total Error Rate (TER)</td>
<td>21.81%</td>
<td>10.49%</td>
</tr>
<tr>
<td>Mean Error Rate (MER)</td>
<td>10.905%</td>
<td>5.245%</td>
</tr>
</tbody>
</table>

Table 19 - Results: Global & Texture Features - TNTA

Figure 57 - Results: Global & Texture Features - TNTA - Adaptively Sized VQ Codebook

Figure 58 - Results: Global & Texture Features - TNTA - Fixed Size (50) VQ Codebook
4.5.5.3 Train All Test All (TATA)

Training Signature Samples: 1495
Testing Signature Samples: 1003

<table>
<thead>
<tr>
<th>Performance Results</th>
<th>Adapively Sized VQ Codebook</th>
<th>Fixed Size (50) VQ Codebook</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cases that should be accepted</td>
<td>1003</td>
<td>1003</td>
</tr>
<tr>
<td>Cases that should be rejected</td>
<td>64192</td>
<td>64192</td>
</tr>
<tr>
<td>Accepted</td>
<td>6174</td>
<td>2721</td>
</tr>
<tr>
<td>Rejected</td>
<td>59021</td>
<td>62474</td>
</tr>
<tr>
<td>Correct Acceptances</td>
<td>920 (91.72%)</td>
<td>954 (95.11%)</td>
</tr>
<tr>
<td>False Rejections</td>
<td>83 (8.28%)</td>
<td>49 (4.89%)</td>
</tr>
<tr>
<td>Correct Rejections</td>
<td>58938 (91.82%)</td>
<td>62425 (97.25%)</td>
</tr>
<tr>
<td>False Acceptances</td>
<td>5254 (8.18%)</td>
<td>1767 (2.75%)</td>
</tr>
<tr>
<td>Total Error Rate (TER)</td>
<td>16.46%</td>
<td>7.64%</td>
</tr>
<tr>
<td>Mean Error Rate (MER)</td>
<td>8.23%</td>
<td>3.82%</td>
</tr>
</tbody>
</table>

Table 20 - Results: Global & Texture Features - TATA

Figure 59 - Results: Global & Texture Features - TATA - Adapively Sized VQ Codebook

Figure 60 - Results: Global & Texture Features - TATA - Fixed Size (50) VQ Codebook
4.5.6 Testing with Grid and Texture Features

The next scenarios are used to measure the performance of the RBF neural network when both grid and texture features are combined (1344 features). Both features describe a signature sample in more detail by segmented it into 96 segments ($1344 \times 8$).

4.5.6.1 Train Non-Frame Test Non-Frame (TNTN)

*Training Signature Samples: *941

*Testing Signature Samples: *631

<table>
<thead>
<tr>
<th>Performance Results</th>
<th>Adaptively Sized VQ Codebook</th>
<th>Fixed Size (50) VQ Codebook</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cases that should be accepted</td>
<td>631</td>
<td>631</td>
</tr>
<tr>
<td>Cases that should be rejected</td>
<td>40384</td>
<td>40384</td>
</tr>
<tr>
<td>Accepted</td>
<td>2655</td>
<td>1443</td>
</tr>
<tr>
<td>Rejected</td>
<td>38360</td>
<td>39572</td>
</tr>
<tr>
<td>Correct Acceptances</td>
<td>588 (93.19%)</td>
<td>612 (96.99%)</td>
</tr>
<tr>
<td>False Rejections</td>
<td>43 (6.81%)</td>
<td>19 (3.01%)</td>
</tr>
<tr>
<td>Correct Rejections</td>
<td>38318 (94.88%)</td>
<td>39554 (97.94%)</td>
</tr>
<tr>
<td>False Acceptances</td>
<td>2066 (5.12%)</td>
<td>830 (2.06%)</td>
</tr>
<tr>
<td>Total Error Rate (TER)</td>
<td>11.93%</td>
<td>5.07%</td>
</tr>
<tr>
<td>Mean Error Rate (MER)</td>
<td>5.965%</td>
<td>2.535%</td>
</tr>
</tbody>
</table>

Table 21 - Results: Grid & Texture Features - TNTN

Figure 61 - Results: Grid & Texture Features - TNTN - Adaptively Sized VQ Codebook

Figure 62 - Results: Grid & Texture Features - TNTN - Fixed Size (50) VQ Codebook
4.5.6.2 Train Non-Frame Test All (TNTA)

Training Signature Samples: 941
Testing Signature Samples: 1557

<table>
<thead>
<tr>
<th>Performance Results</th>
<th>Adapтивely Sized VQ Codebook</th>
<th>Fixed Size (50) VQ Codebook</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cases that should be accepted</td>
<td>1557</td>
<td>1557</td>
</tr>
<tr>
<td>Cases that should be rejected</td>
<td>99648</td>
<td>99648</td>
</tr>
<tr>
<td>Accepted</td>
<td>8490</td>
<td>4671</td>
</tr>
<tr>
<td>Rejected</td>
<td>92715</td>
<td>96534</td>
</tr>
<tr>
<td>Correct Acceptances</td>
<td>1430 (91.84%)</td>
<td>1492 (95.83%)</td>
</tr>
<tr>
<td>False Rejections</td>
<td>127 (8.16%)</td>
<td>65 (4.17%)</td>
</tr>
<tr>
<td>Correct Rejections</td>
<td>92588 (92.92%)</td>
<td>96469 (96.81%)</td>
</tr>
<tr>
<td>False Acceptances</td>
<td>7060 (7.08%)</td>
<td>3179 (3.19%)</td>
</tr>
<tr>
<td>Total Error Rate (TER)</td>
<td>15.24%</td>
<td>7.36%</td>
</tr>
<tr>
<td>Mean Error Rate (MER)</td>
<td>7.62%</td>
<td>3.68%</td>
</tr>
</tbody>
</table>

Table 22 - Results: Grid & Texture Features - TNTA

Figure 63 - Results: Grid & Texture Features - TNTA - Adaptively Sized VQ Codebook

Figure 64 - Results: Grid & Texture Features - TNTA - Fixed Size (50) VQ Codebook
4.5.6.3 Train All Test All (TATA)

Training Signature Samples: 1495
Testing Signature Samples: 1003

<table>
<thead>
<tr>
<th>Performance Results</th>
<th>Adaptively Sized VQ Codebook</th>
<th>Fixed Size (50) VQ Codebook</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cases that should be accepted</td>
<td>1003</td>
<td>1003</td>
</tr>
<tr>
<td>Cases that should be rejected</td>
<td>64192</td>
<td>64192</td>
</tr>
<tr>
<td>Accepted</td>
<td>4023</td>
<td>2212</td>
</tr>
<tr>
<td>Rejected</td>
<td>61172</td>
<td>62983</td>
</tr>
<tr>
<td>Correct Acceptances</td>
<td>943 (94.02%)</td>
<td>971 (96.81%)</td>
</tr>
<tr>
<td>False Rejections</td>
<td>60 (5.98%)</td>
<td>32 (3.19%)</td>
</tr>
<tr>
<td>Correct Rejections</td>
<td>61112 (95.2%)</td>
<td>62951 (98.07%)</td>
</tr>
<tr>
<td>False Acceptances</td>
<td>3080 (4.8%)</td>
<td>1241 (1.93%)</td>
</tr>
<tr>
<td>Total Error Rate (TER)</td>
<td><strong>10.78%</strong></td>
<td><strong>5.12%</strong></td>
</tr>
<tr>
<td>Mean Error Rate (MER)</td>
<td><strong>5.39%</strong></td>
<td><strong>2.56%</strong></td>
</tr>
</tbody>
</table>

Table 23 - Results: Grid & Texture Features - TATA

Figure 65 - Results: Grid & Texture Features - TATA - Adaptively Sized VQ Codebook

Figure 66 - Results: Grid & Texture Features - TATA - Fixed Size (50) VQ Codebook
4.5.7 Testing with Global, Grid and Texture Features

Finally, the system is trained and tested with the combination of all three group of features; global, grid and texture features summing up to a total of 1360 features.

4.5.7.1 Train Non-Frame Test Non-Frame (TNTN)

*Training Signature Samples: 941*
*Testing Signature Samples: 631*

<table>
<thead>
<tr>
<th>Performance Results</th>
<th>Adaptively Sized VQ Codebook</th>
<th>Fixed Size (50) VQ Codebook</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cases that should be accepted</td>
<td>631</td>
<td>631</td>
</tr>
<tr>
<td>Cases that should be rejected</td>
<td>40384</td>
<td>40384</td>
</tr>
<tr>
<td>Accepted</td>
<td>2512</td>
<td>1380</td>
</tr>
<tr>
<td>Rejected</td>
<td>38503</td>
<td>39635</td>
</tr>
<tr>
<td>Correct Acceptances</td>
<td>592 (93.82%)</td>
<td>613 (97.15%)</td>
</tr>
<tr>
<td>False Rejections</td>
<td>39 (6.18%)</td>
<td>18 (2.85%)</td>
</tr>
<tr>
<td>Correct Rejections</td>
<td>38465 (95.25%)</td>
<td>39618 (98.1%)</td>
</tr>
<tr>
<td>False Acceptances</td>
<td>1919 (4.75%)</td>
<td>766 (1.9%)</td>
</tr>
<tr>
<td>Total Error Rate (TER)</td>
<td>10.93%</td>
<td>4.75%</td>
</tr>
<tr>
<td>Mean Error Rate (MER)</td>
<td>5.465%</td>
<td>2.375%</td>
</tr>
</tbody>
</table>

Table 24 - Results: Global, Grid & Texture Features - TNTN

Figure 67 - Results: Global, Grid & Texture Features - TNTN - Adaptively Sized VQ Codebook

Figure 68 - Results: Global, Grid & Texture Features - TNTN - Fixed Size (50) VQ Codebook
4.5.7.2 Train Non-Frame Test All (TNTA)

*Training Signature Samples: 941*
*Testing Signature Samples: 1557*

<table>
<thead>
<tr>
<th>Performance Results</th>
<th>Adaptively Sized VQ Codebook</th>
<th>Fixed Size (50) VQ Codebook</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cases that should be accepted</td>
<td>1557</td>
<td>1557</td>
</tr>
<tr>
<td>Cases that should be rejected</td>
<td>99648</td>
<td>99648</td>
</tr>
<tr>
<td>Accepted</td>
<td>7703</td>
<td>4490</td>
</tr>
<tr>
<td>Rejected</td>
<td>93502</td>
<td>96715</td>
</tr>
<tr>
<td>Correct Acceptances</td>
<td>1436 (2.23%)</td>
<td>1497 (96.15%)</td>
</tr>
<tr>
<td>False Rejections</td>
<td>121 (7.77%)</td>
<td>60 (3.85%)</td>
</tr>
<tr>
<td>Correct Rejections</td>
<td>93381 (93.71%)</td>
<td>96655 (97%)</td>
</tr>
<tr>
<td>False Acceptances</td>
<td>6267 (6.29%)</td>
<td>2993 (3%)</td>
</tr>
<tr>
<td>Total Error Rate (TER)</td>
<td>14.06%</td>
<td>6.85%</td>
</tr>
<tr>
<td>Mean Error Rate (MER)</td>
<td>7.03%</td>
<td>3.425%</td>
</tr>
</tbody>
</table>

Table 25 - Results: Global, Grid & Texture Features - TNTA

Figure 69 - Results: Global, Grid and Texture Features - TNTA - Adaptively Sized VQ Codebook

Figure 70 - Results: Global, Grid & Texture Features - TNTA - Fixed Size (50) VQ Codebook
4.5.7.3 Train All Test All (TATA)

Training Signature Samples: 1495
Testing Signature Samples: 1003

<table>
<thead>
<tr>
<th>Performance Results</th>
<th>Adapтивely Sized VQ Codebook</th>
<th>Fixed Size (50) VQ Codebook</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cases that should be accepted</td>
<td>1003</td>
<td>1003</td>
</tr>
<tr>
<td>Cases that should be rejected</td>
<td>64192</td>
<td>64192</td>
</tr>
<tr>
<td>Accepted</td>
<td>3794</td>
<td>2132</td>
</tr>
<tr>
<td>Rejected</td>
<td>61401</td>
<td>63063</td>
</tr>
<tr>
<td>Correct Acceptances</td>
<td>948 (94.52%)</td>
<td>973 (97.01%)</td>
</tr>
<tr>
<td>False Rejections</td>
<td>55 (5.48%)</td>
<td>30 (2.99%)</td>
</tr>
<tr>
<td>Correct Rejections</td>
<td>61345 (95.56%)</td>
<td>63034 (98.2%)</td>
</tr>
<tr>
<td>False Acceptances</td>
<td>2847 (4.44%)</td>
<td>1158 (1.8%)</td>
</tr>
<tr>
<td>Total Error Rate (TER)</td>
<td>9.92%</td>
<td>4.79%</td>
</tr>
<tr>
<td>Mean Error Rate (MER)</td>
<td>4.96%</td>
<td>2.395%</td>
</tr>
</tbody>
</table>

Table 26 - Results: Global, Grid & Texture Features - TATA

Figure 71 - Results: Global, Grid & Texture Features - TATA - Adapтивely Sized VQ Codebook

Figure 72 - Results: Global, Grid & Texture Features - TATA - Fixed Size (50) VQ Codebook
4.6 Average Receiver Operating Characteristic (ROC) Curves

In order to measure the error rates for the RBF neural networks, it was decided that ROC curves (FAR vs. FRR) will be plotted to visualise the overall performance of the above scenarios. Each ROC curve is used to find the best setting of the threshold $\alpha$ termed as the operating point. Each questioned signature which produces an output value greater than $\alpha$ is considered as genuine, otherwise it is considered as a forgery. The setting of $\alpha$ corresponds to the least total error rate possible. Note that the ROC curves are plotted for the test data set of signature samples and the operating point determined on them. In the curve, the operating point is the point closest to the origin.

The following two figures show ROC curves when the system was evaluated with an adaptive size VQ codebook and a fixed size VQ codebook respectively. In both cases, 7 ROC curves are depicted, representing the third scenario (TATA), for each of the features used for training and testing. The third scenario is the most representative as the system is trained with both framed and non-framed signature samples.

Figure 73 - Average ROC: All 7 features - TATA - Adaptively Sized VQ Codebook
The above figures depict the overall performance of each of the features using either an adaptive size VQ codebook or a fixed size VQ codebook of 50 codewords. The figures show that when the grid and texture column features were clustered using 50 codewords, the system performed much better than when the adaptive approach was used.

Figure 73 shows that the best performance was achieved when the system was trained with global features where the operating point is at (0.06, 0.04). On the other hand Figure 74 shows that the combination of global and grid features managed to produce the best performance results, where the operating is at (0.034, 0.033). In both cases, the system performed less when texture features were used for training.

### 4.7 Conclusion

The obtained results are quite promising where the best TER of 6.95% was achieved by the global features. The texture features were found to perform the least, while the grid features produced very good results close to the global features. The combination of the features helped to improve the performance measurements accordingly.

A complete and detailed discussion on the obtained results is given in the next chapter.
Chapter 5  Discussion
5.1 Introduction

The objective of this chapter is to analyse the obtained results, in the previous chapter, reflecting the best features and the best classification technique. Additionally, this chapter contains a comparison of our results with other published studies which used a similar methodology. Any limitations and robustness of the technique used are also discussed in this chapter.

5.2 Analysis of Results

Chapter 4 showed the results of 39 different experiments. The experiments consisted of a combination of global, grid and texture features together with two alternatives of vector quantization approaches where appropriate. For each of these combinations, the following three scenarios were evaluated:

1. The system is trained and tested with non-framed signature samples only
2. The system is trained with non-framed signature samples but tested with both framed and non-framed signature samples
3. The system is trained and tested with both framed and non-framed signature samples

5.2.1 Vector Quantization (VQ) Effect

In Section 3.4.2.2, it was argued how to define the number of codewords required for vector quantization. The following two alternatives were used to evaluate the system performance:

1. Adaptively sized codebook: finding the number of codewords by using the cluster validity measure proposed by Ray and Turi in [61]
2. Fixed size codebook: clustering the column feature vectors in 50 codewords

In both cases, only one codebook was used due to the small size of the signature database. Surprisingly, the fixed size codebook VQ performed at least twice as good as the adaptively sized codebook VQ. When texture features were used, the performance results of the fixed size codebook VQ were even three times better than the adaptively sized codebook VQ.

These results suggest that the column feature vectors do not form compact clusters that can be suitably partitioned, thus resulting in incorrect cluster validity measures. The number of codewords that were obtained by the cluster validity measure was too small resulting in a loss of discriminatory information. Consequently, the RBFNN was unable to perform a good classification. Choosing a fixed size of 50 codewords retained sufficient discrimination between the features to allow differentiation between signatures.
5.2.2 Results
The best overall performance of all experiments was achieved when the system was trained with global and grid features together where the system obtained a TER of 4.08% with an FRR of 1.58%, an FAR of 2.5% and an MER of 2.04%. These results were obtained when the system was trained and tested with non-framed signature samples only and a fixed size VQ codebook for the grid column features.

On the other hand, the worst performance results were obtained when texture features alone were used. In this case the system achieved a TER of 11.83% with an FRR of 6.94%, an FAR of 4.89% and an MER of 5.915%. This may be caused due to the fact that texture features were extracted from the skeleton image rather than from the binary image. Since the signature acquisition process (See Section 3.1) was carried out by different pens having different colours and tip thickness, the binary image would not reflect the real pressure of the author. For this reason, it was decided to use the skeleton image for both texture and grid features.

5.2.3 Data Acquisition Effect
The way the signature samples were collected also effected the obtained results. The proposed method by Mr. Gaffiero (See Appendix D) to collect signatures within different dimension size frames effected the orientation of the signature by many authors. The fact that the authors were provided visible orientation guidelines, the proportionality of many signatures depended on the available space. In reality, there are a lot of circumstances where a signature is required in restricted spaces such as credit cards and passports. Other signature samples were collected on blank sheets of paper where the authors where allowed to provide 5 signatures on each A4 blank paper.

The second scenario of each feature combination was performed to analyse the performance of the system when no knowledge of the framed signatures was available at training stage. In fact, in each experiment the less successful results were obtained when the system was trained with signature samples that were collected without a frame and subsequently tested with both framed and non-framed signature samples.

The proportionality effect of the signature samples collected within frames effected the performance of the RBF neural networks. However, the best results in this case, were also obtained when the system was evaluated with global and grid features. A TER of 6.31% with an FRR of 3.4%, an FAR of 2.91% and an MER of 3.155% were obtained for this scenario.

5.2.4 FRR vs. FAR
An optimal biometric system would have an FRR and FAR as low as possible. However, in reality, it is almost impossible to achieve this situation because tuning a biometric system to produce lower FRR causes to increase the FAR and vice-versa. So, a balance must be found between FAR and FRR.
Sometimes it is desirable that a biometric system produces a less successful TER but produces a lower FRR or a lower FAR. A lower FRR means that the system rejects genuine signatures as low as possible but accepting more signature forgeries. On the other hand a lower FAR means that the system is more secure by accepting less signature forgeries with a penalty of rejecting more genuine signatures.

In our experiments the lowest FRR was achieved when the system was evaluated with global and grid features, in which the system was trained and tested with both framed and non-framed signatures. In this case, an FRR of 1.3% was achieved where the system rejected just 13 genuine signature samples out of 1003. On the other hand, the lowest FAR of 1.8% was achieved when the system was evaluated with all features; that is global, grid and texture features.

In general, the system achieved very close results when it was evaluated with the first and third scenarios. This means that if the system is also trained and tested with framed signatures it produced very close results to when the system is trained and tested with non-framed signatures only.

5.3 Cost of Training and Verification

5.3.1 Hardware Specifications

The following table shows the hardware specifications used for this study:

<table>
<thead>
<tr>
<th>Hardware</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processor</td>
<td>Intel Centrino 1.6GHz</td>
</tr>
<tr>
<td>Physical Memory</td>
<td>DDR2 - 1024MB</td>
</tr>
</tbody>
</table>

Table 27 - Hardware specifications

5.3.2 Training

The cost of the training depends on the number of features and the number of signature samples used for training. Since no general purpose gradient descent algorithm was used, the RBF neural network had relatively cheap computations. The proposed RBF neural network is trained by finding the weight vector using a pseudo-inverse of the interpolation matrix. The computation of the weight vector is very fast and requires simple mathematics.

Hence the cost of the training lies in normalizing the input vectors and particularly in vector quantization using the K-Means algorithm. When the system was evaluated with only global features (16 features), the training of a signature model using a priori knowledge of the other 64 authors was completed in less than 15 minutes. However, when the training required vector quantization such as using grid and texture features, a signature model took about 30 minutes to complete. The worst case scenario was when all three features were combined as one feature vector containing 1360 elements (16 global features, 576 grid features and 768 texture features). In this case a signature model took circa 45 minutes to complete.
5.3.3 Verification

The process of verifying a questioned signature consists of the following steps:

1. Pre-process the signature image by generating binary and skeleton images.
2. Extract the features required by the system (global, grid and texture).
3. If grid or texture features are required, use vector quantization to replace each feature vector by the representative codeword.
4. Normalise the feature representative vector.
5. Computes the interpolation matrix using Gaussian functions.
6. Computes a multiplication of the resultant interpolation matrix with the weight vector established during the training phase.
7. If the output is greater than or equal to the threshold established during the testing phase, then the signature is genuine otherwise it is a forgery.

The above steps are carried out in less than 45 seconds irrespective of the features used. The longest step, in this case, is the calculation of two global features namely local and global slant angles where they take approximately 15 - 20 seconds to complete.

5.4 Limitations

One limitation of the proposed system is that the training and testing did not involve any cross validation technique. One possible way of achieving cross-validation would have been the leave-one-out (LOO) method, which ensures that each signature model is trained and tested with every signature sample available, enabling a better estimation of the performance results.

Mainly, the reason behind this limitation is due to time constraints. It was preferred to investigate the performance of several features using adaptive size VQ codebook and fixed size VQ codebook of 50 codewords rather than adopting the mentioned cross validation technique.

5.5 RBF Robustness

The RBF neural network is popular for the robustness in eliminating outliers [42]. In our context, outliers are the signature samples that do not match with the genuine signature model.

From the obtained results, it could be observed that the best results were obtained by signature models containing low intrapersonal variations. The lower the intrapersonal variations the better the performance results. For instance, Author #19 maintained the lower TER from all evaluation scenarios. On the other hand, Author #5 whose signatures had a large intrapersonal variation achieved the maximum TER.

The robustness in eliminating outliers effect the RBF neural network to be more sensitive in rejecting genuine signature samples. This results an FRR greater than FAR. In fact, when a fixed size VQ codebook of 50 codewords was used, the average FRR of all experiments is 3.74% whereas the average FAR is 3.18%. Furthermore,
when an adaptively sized VQ codebook was used, the average FRR of all experiments is 7.76% while the average FAR is 6.64%. These results may be seen to confirm the robustness of the RBF neural network in eliminating outliers.

5.6 Comparison of Results

Since no international common signature database exist, every study in this field uses different signature databases. For this reason, comparison of results is very difficult to perform.

Hence comparison must be carried out with the mentioned restriction in mind. The study performed by Baltzakis and Papamarkos in [7], produced a TER of 12.81% with an FRR of 3% and an FAR of 9.81% (See Table 2). In this study, Baltzakis and Papamarkos used a combination of global, grid and texture features together with a Euclidean distance, and the final decision is performed by an RBFNN. When adopting the same three groups of features, our system achieved better results where a TER of 4.79% was obtained with an FRR of 2.99% and an FAR of 1.8% (See Section 4.5.7.3).

Another interesting study performed by Justino et al in [35], achieved an MER of 2.135% when grid features, comprised of pixel density, pixel distribution and axial slant, were used in an HMM classifier. When adopting the same features in our study, our system achieved a MER of 2.295% (See Section 4.5.2.3) which is only slightly worse to the results obtained by Justino et al in [35]. However, our proposed system achieved better results when the grid features where combined with global features where a MER of 2.07% was achieved.

5.7 Conclusion

This chapter analysed and discussed in detail the obtained results in Chapter 4. A total of 39 experiments were carried out where the system was evaluated with combination of the three sets of features and using two alternatives of vector quantization. The best performance results were obtained when the system was evaluated with global and grid features (fixed size VQ codebook of 50 codewords), producing a TER of 4.08%.

The next chapter gives a summary of the entire study and outlines any limitations, recommendations and contributions.
Chapter 6  Conclusion
6.1 Summary
The objective of this work was to analyse the effectiveness of RBF neural networks in the field of offline handwritten signature verification.

This study implemented an entire offline handwritten signature verification system using a totally RBF neural network solution. The system’s methodology consisted of data acquisition, pre-processing, feature selection and extraction, classification and evaluation. The pre-processing and feature selection and extraction processes were a determinant factor in this study. Three groups of features namely global, grid and texture features were selected to extract information from signature samples. Both preprocessing and feature extraction processes required implementation of several algorithms (See Appendix F).

The focus of this study was to analyse the effectiveness of the proposed RBF single-layer architecture. The promising results obtained in Chapter 4 show the robustness of the proposed architecture. The algorithms implemented for pre-processing and feature extraction also helped to produce promising results. The powerful Gaussian functions used to map a multi-dimensional space into a linear space helped to keep mathematics simple (linear algebra). Additionally, the pseudo-inverse technique facilitated the training of the neural network as no gradient descent algorithms were required.

A TER of 4.08% consisting of an FRR of 1.58% and an FAR of 2.5% were the best results obtained when the system was evaluated with both global and grid features. In this case, each RBF neural network (one for each author) was fed by a number of training vectors (signature samples) each containing 592 features. A codebook of 50 codewords was also used in this scenario to perform vector quantization on the grid features.

6.2 Limitations
Since the system was evaluated on a collected signature database comprised of 2498 signature samples from 65 authors, the obtained results cannot be generalised to represent the general performance of the proposed architecture. However, the sample size is comparable to other studies in the field; this fact and the fact that the system was validated by splitting the 2498 signature samples into training and testing samples provide enough confidence in the results obtained and indicate that RBFNNs have promising performance in this field.

Another limitation is that the system’s performance was measured using only random signature forgeries. This is because no simple and skilled signature forgeries were obtained.

6.3 Contributions
To our knowledge, no other published work investigated the use of a totally RBF architecture solution for offline handwritten signature verification. The promising results obtained by this study may motivate other researchers to investigate further the use of RBFNN in this field.
6.4 Future Work and Recommendations

Future work may include a system evaluation with a larger database containing more than 65 authors. An interesting work would also be to evaluate the system with less signature samples for each author, so that the system will have less samples in the learning phase.

Since the proposed system was tested with only random signature forgeries, it will be interesting to test the system with simple and skilled forgeries. In this study, a grid and texture features were extracted from a grid of $12 \times 8$ segments. Thus, future works may investigate the performance of RBFNN with different grid dimensions. Additionally, a future work may also test the system by varying the number of codewords used for vector quantisation.

Other future works will use a cross-validation technique, such as leave-one-out (LOO), in order to obtain better estimates for the performance. Also, investigating the use of feature selection techniques such as principal component analysis by reducing the dimensionality of the feature space, will also be a challenging study.
Appendices
A. Project Description Form

CIS320 Project Description Form

One copy of this form is to be completed by each project student. The form must be signed by the student. The original is to be submitted with the examination entry form to the University of London. Any deviation from the proposal should be explained by the student in the report submitted. The student should keep a copy of the project description form.

To Be Completed By The Candidate

<table>
<thead>
<tr>
<th>Name: Mr. George Azzopardi</th>
<th>Registration No: U / 01 / 0316557</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title of Project: How effective are Radial Basis Function Neural Networks for Offline Handwritten Signature Verification?</td>
<td></td>
</tr>
</tbody>
</table>

1. Statement of Objectives
   a. What do you intend to achieve?

   The objective of this project is to investigate the use of Radial Basis Function Neural Networks (RBFNN) in offline handwritten signature verification and compare its performance to conventional Multi-Layer Perceptrons (MLP). The evaluation will be performed on a signature data set collected for this purpose.

   b. Why have you chosen the proposed project?

   Due to the dependence on electronic storage and transmission of information, the need for electronically verifying a person’s identity has been raised. Handwritten signatures are a traditional and conventional way for identity verification. A typical application would be a small device placed on the desk of a Bank teller used during encashment of cheques. The teller will decide upon the device output.

   RBFNNs are popular for their robustness in eliminating outliers and for the simple computation due to single layer architecture. For this reason, an RBFNN classifier is chosen for this study to analyse its potential with regards to such system. They are expected to perform better than MLP neural networks in this regard.

   Furthermore, little have been studied on RBFNNs in handwritten signature verification systems, in fact, only one paper was found during my literature review, which described an RBFNN approach, in which a number of MLPs are feeding an RBFNN (two stage approach) to have the final decision.

   c. Itemised results/documents you intend to deliver in the achievement of your objectives:

      1. A Literature Review of the currently used offline handwritten signature verification methods
      2. The consent forms used for the collection of signatures for this study
      3. A performance evaluation of the MLP, MLP-RBFNN and RBFNN approaches in terms of False Acceptance Rate (FAR), False Rejection Rate (FRR), Mean Error Rate (MER) and any other relevant performance metrics.
      4. A discussion and recommendations regarding the use of RBFNNs to offline handwritten signature verification.
      5. Conclusion of the entire study

2. The Methods to be used
   a. How you intend to achieve the objectives listed above:

      The objectives listed above will be achieved by:

      1. Start a literature search on the existing methods for offline handwritten signature verification in order to analyse and compare the methods used and results reached so far (using IEEE and http://scholar.google.com/). This will be followed by an interview with a Graphologist, who works at the Maltese Law Courts, to help identify the main features of an offline signature.
      2. A data set of 25 signatures will be collected from at least 40 authors, and after the scanning process, they will be stored in a relational database.
      3. Preprocessing will be performed using binarization and thinning techniques. The image result of processed signatures will be stored in the database too.
      4. Feature Extraction will be performed, where on the basis of literature review and graphologist’s knowledge, appropriate features (such as signature width and height and number of edge points) will be extracted and stored in the database.
5. Implementation will be performed by Java™ programming language using Java Advanced Imaging (JAI) package where necessary, and it will consist of the following 3 approaches:
   i. An MLP approach
   ii. A two-stage approach where a number of MLPs feed forward an RBFNN. This follows from the only study found using RBFNNs mentioned in (1b).
   iii. A single layer RBFNN approach

6. Testing and evaluation using established performance metrics (e.g. FAR, FRR, MER)

b. Why you are intending to do it this way:
   1. A literature review is required to obtain a good insight into current offline handwritten signature verification methods. The interview with a graphologist will provide direct expert knowledge on the art of signature verification.
   2. A data set of signatures will be collected because no public database was found available for this study.
   3. Preprocessing using binarization and thinning (also called skeletonization) will be required in order to reduce the amount of data in the signature image without losing the basic structural information.
   4. Feature Extraction is required to extract the relevant features from the signature that will capture the characteristics of the signature structure.
   5. The implementation will enable us to evaluate the performance of an RBFNN against other approaches (MLP and MLP-RBFNN) that were found to be used in offline handwritten signature verification. Java programming will be used as it is platform independent and also provides robustness in imaging processing through the Java Advanced Imaging package.
   6. FAR, FRR and MER are conventional and established metrics used for such studies. The lower the rates the better the systems. An optimum solution is one which produces 0% in both FAR and FRR.

c. Your strategy for getting started:
   In order to achieve the above-mentioned deliverables, the following strategy will be used:
   1. Study the theory of radial basis function neural networks
   2. Contact Data Protection Commissioner of Malta to ask for authorization for the collection of signatures.
   3. Collect a data set of 25 signatures from at least 40 authors
   4. Extends the literature review to identify the specific features to be extracted

3. The Work Plan
   a. A schedule showing key milestones in the project.

<table>
<thead>
<tr>
<th>Milestone</th>
<th>Target End Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Literature Review</td>
<td>31st January, 2006</td>
</tr>
<tr>
<td>Collection of signatures, scanning, storage and pre-processing</td>
<td>31st January, 2006</td>
</tr>
<tr>
<td>Implementation of signature features extraction</td>
<td>28th February, 2006</td>
</tr>
<tr>
<td>Implementation and Performance Evaluation of methods used</td>
<td>7th April, 2006</td>
</tr>
<tr>
<td>Discussion and Recommendations</td>
<td>15th April, 2006</td>
</tr>
<tr>
<td>Complete Report</td>
<td>30th April, 2006</td>
</tr>
</tbody>
</table>

   b. A production schedule for the report (i.e. when you will start writing and when it will be finished).

<table>
<thead>
<tr>
<th>Dissertation Section</th>
<th>Target Start Date</th>
<th>Target End Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Terms of Reference</td>
<td>1st December, 2005</td>
<td>5th February, 2006</td>
</tr>
<tr>
<td>Literature Review</td>
<td>15th January, 2006</td>
<td>15th February, 2006</td>
</tr>
<tr>
<td>Methods, Results and Evaluation</td>
<td>16th February, 2006</td>
<td>15th April, 2006</td>
</tr>
<tr>
<td>Discussion</td>
<td>10th April, 2006</td>
<td>20th April, 2006</td>
</tr>
<tr>
<td>Limitations</td>
<td>15th April, 2006</td>
<td>20th April, 2006</td>
</tr>
<tr>
<td>Conclusion</td>
<td>21st April, 2006</td>
<td>30th April, 2006</td>
</tr>
<tr>
<td>Evaluation</td>
<td>30th April, 2006</td>
<td>5th May, 2006</td>
</tr>
</tbody>
</table>

Additional Comments
Use this section to make extra comments on the proposal, on matters not covered above or where space is insufficient.

Signed
Name: Date:
B. Data Protection Commissioner Correspondence

B.1 Request

Dr. Roberta,

With regards to my dissertation data collection, I would like to confirm that by gathering the following personal information, no Data Protection rules will be violated, and no further authorisations are required except for the individual's consent.

1. Title
2. Name
3. Surname
4. Gender
5. Date of Birth
6. Job Title
7. Level of Education
8. City
9. Country
10. Telephone
11. Mobile No
12. Email
13. A sample of 25 - 40 handwritten signatures

The following information will be given to each individual to confirm his/her consent.

"Subject: BSc Computing and Information Systems - Dissertation

To whom it may concern,

Currently I am reading my final year in BSc. Computing & Information Systems with University of London, and I need to submit a dissertation as partial fulfilment of my degree. The aim of my dissertation is to analyse the effect of Radial Basis Functions, as classifiers, with respect to Offline Handwritten Signature Verification System.

In order to start my project, I need to create a database of signatures from different authors. The signatures obtained will be used for system training and testing purposes only, and eventually will be acknowledged publicly in my final report submitted to the university. The report will also be available to the public from St. Martin’s Institution of IT library located at Pieta.

I hereby request your authorisation to use a set of 25 - 40 written signatures together with your full name, date of birth, gender, job title, place and contacts. Once the signatures are scanned and stored in a database, the original written signatures will be destroyed."
The following is the individual's consent

"I have read and understood the information explained above concerning the mentioned project. I hereby give my consent to use my set of 25-40 signatures for the above-mentioned study.

Full Name (in BLOCK LETTERS) Signature"

Thanks in advance for your cooperation

Regards

George Azzopardi

B.2 Reply

Mr Azzopardi,

The list of data indicated in your mail does not include sensitive personal data and so no authorisation in terms of article 16 of the Act is required.

Having said that, consent must still be sought and full information given to the participants.

Regards and Good Luck

Dr Roberta Peresso BA, LLD
f/Data Protection Commissioner

Data Protection Commissioner
2, Airways House
High Street
Sliema SLM 16
Malta

Tel:(+356) 2328 7100
Fax:(+356) 2328 7198
Website: www.dataprotection.gov.mt
C. Data Acquisition Sample

C.1 Consent Form

Mr. George Azzopardi
“St. John”
Antonio Muscat Fenech Street
Qormi, QRM11

Tel: 21 447926 / 99845118
Email: geazzo@waldonet.net.mt

Date: December 2005

Subject: BSc Computing and Information Systems - Dissertation

To whom it may concern,

Currently I am reading my final year in BSc. Computing & Information Systems with University of London, and I need to submit a dissertation as partial fulfilment of my degree. The aim of my dissertation is to analyse the effect of Radial Basis Functions, as classifiers, with respect to Offline Handwritten Signature Verification System.

In order to start my project, I need to create a database of signatures from different authors. The signatures obtained will be used for system training and testing purposes only, and eventually will be acknowledged publicly in my final report submitted to the university. The report will also be available to the public from St. Martin’s Institution of IT library located at Pieta.

I hereby request your authorisation to use a set of 25-40 written signatures together with other personal information mentioned in the next page. Once the signatures are scanned and stored in a database, the original written signatures will be destroyed.

Regards

George Azzopardi

I have read and understood the information explained above concerning the mentioned project. I hereby give my consent to use my set of 25-40 signatures for the above-mentioned study.

Full Name (in BLOCK LETTERS)  Signature
C.2 Personal Details

Data Collection

Kindly fill in the following information.

In the next pages you are required to sign 40 signatures over a period of 5 days. Kindly use the different provided pens and sign accordingly.

Personal
Title:  
Surname:  
Other Names:  
Gender (mark ☑ where appropriate):  ☑ Male  ☐ Female  
Date of Birth (dd/mm/yyyy):  04/07/1981  
Job Title:  SENIOR PROGRAMMER

Level of Education
(mark ☑ where appropriate)

Primary  ☐  
Secondary  ☐  
Tertiary
Undergraduate
Diploma  ☑  Degree  ☑  
Postgraduate
Masters  ☐  PhD  ☐

Place
City:  QORMI  
Country:  MALTA

Contact
Tel:  21447926  
Mob:  99845118  
Email:  geazz@gmail.com.net.mt

Student No: U / 01 / 0316557  2  George Azzopardi
C.3 Signature Samples
Please sign within the provided boxes without overlapping the borderline.
D. Transcript of interview with Maltese Graphologist

The purpose of this interview is to understand how signature verification process is currently being done at the Malta Law Courts. The interviewee, who is a Maltese graphologist, was asked to suggest an adequate format to collect reference signatures and to highlight the most important characteristics of a handwritten signature.

D.1 Agenda

Interviewer: Mr. George Azzopardi
Interviewee: Mr. Joseph Gaffiero
Meeting Place: At the residency of Mr. Gaffiero in Sliema
Date: 17th October, 2005
Time: 15:00
Duration: 1 hour

Introduction

1. Can you please describe your role within the Maltese Courts?
2. How long have you been serving in this position?
3. How did you find yourself in this field?
4. Have you obtained any form of qualifications in this area?

Signature Verification Process

5. Could you kindly explain the process of signature verification?
6. Do you use some sort of computer systems in order to help with the verification process? Are there other tools which are used in this process?
7. What are the basic characteristics you look for in a signature?
8. If you are still unsure, what are the next characteristics you look for?
9. Are there any circumstances where you refer to someone else? If yes what is the procedure?

General Questions

10. How accurate do you think are the analysis tasks carried out?
11. Are there any characteristics that you think cannot be examined by a human expert, but can be possibly examined by a computer system?
12. Do you think human experts are better than computer systems in this field?

Recommendations

13. Which features in your opinion are essential to be implemented in a computer system to verify signatures?
14. I need to collect genuine signature samples as part of my study. What are your recommendations in this regard?

Other Questions

15. How popular is graphology?
D.2 Interview Report

The interview was carried out in Maltese. For the purpose of the dissertation the interview is being translated in English. Although a semi-structured interview tool was prepared beforehand, the interviewer allowed the conversation to unfold freely. An interpersonal relationship was established. An explanation about the purpose of the study and what was expected from him was immediately given.

*Interviewer: George Azzopardi – GA*

*Interviewee: Joseph Gaffiero – JG*

**Introduction**

1. **GA:** Can you please describe your role within the Maltese Courts?

   *JG:* I am a technical court referee, and my role is to do forensic analysis to determine authenticity of the questioned signature.

2. **GA:** How long have you been serving in this position?

   *JG:* First of all, I must say that I am an artist and I always was fascinated by handwritten signatures. It is almost 20 years now that I have been doing this work.

3. **GA:** How did you find yourself in this field?

   *JG:* As already said, I am artist and in my free time I like painting and even sculpting. Furthermore, I used to work as a branch manager with a Maltese banking institution. The signature forgery was always a problem especially in financial transactions were a handwritten signature is required for all transactions. This motivated me to start a new career where I can just focus in this forensic field.

4. **GA:** Have you obtained any form of qualifications in this area?

   *JG:* There are no specialised qualifications in this area. Signature verification is studied as part of a forensic course which covers all forensic aspects. My only qualification is the experience I gained in the area over the past twenty years.

**Signature Verification Process**

5. **GA:** Could you kindly explain the process of signature verification?

   *JG:* When a legal case concerning a suspected handwritten signature is in progress, I am called by the Maltese courts to analyse the questioned handwritten signature against the genuine signatures of the original owner.
The process is not trivial in view of the fact that I first need to obtain a set of genuine signatures for comparison purposes. This will be more difficult if the owner of the signature passed away and no genuine signatures are immediately available. In this case a special permission would be required to obtain the original signature from the original legal documents. These are supplied from a notary.

After detailed analysis and comparison, the suspected forger is asked to sign in front of me in order to observe the speed and hesitancy at the time of signing.

6. **GA:** Do you use some sort of computer systems in order to help with the verification process? Are there other tools which are used in this process?

   **JG:** No computer systems are used in this process. A magnifying glass and a microscope are the only tools used to analyse the signature.

7. **GA:** What are the basic characteristics you look for in a signature?

   **JG:** It depends on the shape of the signature, whether it has a lot of curves and whether it consists of a full name or just the initials. I usually start off by looking for discontinuities in the signature which determines hesitancy which usually ends up in a forged signature.

   Other basic characteristics which I usually analyse are the following.
   - Pressure - thickness
   - Size of the signature including proportionality.
   - Comparison of letter by letter
   - The position of the signature with respect to a provided baseline and frame
   - Starting point – the first dot of the signature
   - Curve analysis – usually hesitancy is observed in curves. Curves also can provide more information such as the pressure used for going up and that of going down the curve
   - Slope analysis – the inclination of the signature

8. **GA:** If you are still unsure, what are the next characteristics you look for?

   **JG:** If I am still undecided, I ask the suspect forger to sign in front of me (approximately ten signatures in all) in order to observe other characteristics, such as hesitancy, speed and stoppages used during the process of signing.

9. **GA:** Are there any circumstances where you refer to someone else? If yes what is the procedure?

   **JG:** Of course, this happens when the questioned signature looks very similar to the genuine signature. In this case I will refer the case to the other technical court referee (we are only two in Malta) at the disposition of the Maltese courts.
General Questions

10. GA: How accurate do you think are the analysis tasks carried out?

   JG: Of course we cannot say that they are 100%. However it is a fact that a human expert in this field is very good at identifying forgeries, but sometimes we fail to identify a forgery from a genuine signature.

11. GA: Are there any characteristics that you think cannot be examined by a human expert, but can be possibly examined by a computer system?

   JG: No, not that I am aware of. However, I am sure that since a computer has a lot of processing power, it might serve to analyse some characteristics in more detail and in less time.

12. GA: Do you think human experts are better than computer systems in this field?

   JG: I appreciate the fact that the level of detail analysed by computer system is much higher than human experts. As already explained, we use microscopes and magnifying glass in order to enlarge the signature image to look for more detail.

   However, I believe that a human expert, having a real sharp eye, can analyse signature characteristics such as curves and stoppages more effectively than a computer system.

Recommendations

13. GA: Which features in your opinion are essential to be implemented in a computer system to verify signatures?

   JG: I believe that all features mentioned earlier are essential. However, from my experience, I believe that hesitancy, curve analysis and orientation within the provided area are the most essential.

14. GA: I need to collect genuine signature samples as part of my study. What are your recommendations in this regard?

   JG: I would recommend developing a template having different sized frames. This template will be distributed to the signers. Some signers are affected by the orientation of the provided area. Sometimes the signature shape changes according to the provided frame even if the other characteristics remain constant. To give you an example if the signer is used to do wide curves, s/he will still do wide curves even if the size of the signature area is varied.
Other Questions

15. GA: How popular is graphology?

JG: Graphology is very popular in forensic and psychiatry fields. In Malta, graphology is still in its infancy. For instance, in foreign countries, job applications are specifically requested to be submitted in the applicants’ handwriting. These are forwarded to graphologists to determine the individual’s personality. It might determine the ideal candidates for the job.

Mr. Gaffiero was very pleased with the interest shown in this field. He immediately made me feel home by offering coffee and some biscuits. He is very enthusiastic to see the obtained my results. Moreover, he asked whether it would be possible to show him the implemented application and to give him a copy of my dissertation.

Interview Analysis

The answers provided my Mr. Gaffiero were further analysed, in order to be placed in the appropriate context of the study.

It could be noted that the characteristics mentioned in question 7 refer to the offline signature characteristics, where the expert (human or computer system) has only the signature image as an input without any knowledge of the signing process. However, in question 8, Mr. Gaffiero replied that the suspected signer is requested to sign 10 signatures in front of the expert. In this case, the expert is examining online characteristics such as speed, stoppages and starting point.
E. Entity Relationship (ER) Diagram

The following ER diagram shows the architecture of the database used for the study of offline signature verification. It contains information about the Author, Signatures, Global Features, Grid Features and Texture Features. Oracle Database 10g Enterprise Edition Release 10.2.0.1.0 is the relational database management system (RDBMS) used.

Figure 75 - Entity Relationship (ER) Diagram
F. Summary of the Main Implemented Algorithms

<table>
<thead>
<tr>
<th>No</th>
<th>Name</th>
<th>Reference</th>
<th>Section used</th>
<th>Scope</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Data Area Cropping</td>
<td>Gonzalez C. and Wintz [23] (cited by [8])</td>
<td>Pre-Processing</td>
<td>To remove extra background from the handwritten signature image</td>
</tr>
<tr>
<td>2</td>
<td>Thinning</td>
<td>Quek and Zhou [58]</td>
<td>Pre-Processing</td>
<td>To reduce the amount of data by storing a skeleton of the image without losing information about the structure</td>
</tr>
<tr>
<td>3</td>
<td>Pure Width and Pure Height</td>
<td>Qi and Hunt [57]</td>
<td>Global Feature Extraction</td>
<td>To remove the horizontal and vertical blank pixels respectively</td>
</tr>
<tr>
<td>4</td>
<td>Global and Local Slant Angles</td>
<td>Qi and Hunt [57]</td>
<td>Global Feature Extraction</td>
<td>To find the signature angle of inclination globally and locally</td>
</tr>
<tr>
<td>5</td>
<td>K-Means</td>
<td>Bishop [12]</td>
<td>Classification</td>
<td>To cluster data points in ( k ) clusters</td>
</tr>
<tr>
<td>6</td>
<td>Determination of number of clusters in K-Means</td>
<td>Ray and Turi in [61]</td>
<td>Classification</td>
<td>To find the optimal number of clusters required to cluster the given data points</td>
</tr>
<tr>
<td>7</td>
<td>Vector Quantization</td>
<td></td>
<td>Classification</td>
<td>After using K-Means to cluster the column vectors, each column vector is then replaced by the centroid of the respective cluster.</td>
</tr>
<tr>
<td>8</td>
<td>RBF Network</td>
<td>Haykin [28]</td>
<td>Classification</td>
<td>A single-layer architecture used to train the signature samples</td>
</tr>
</tbody>
</table>

Table 28 - Summary of the main implemented algorithms
G. Program Implementation
H. CD Contents

Further to this report, a CD was created containing other information that could not be included in this report. The following are the contents of the CD.

H.1 Dissertation Document

The following two document formats of the dissertation are available on the CD

1. D:\GA_CIS320\Dissertation\GA_CIS320Project.doc
2. D:\GA_CIS320\Dissertation\GA_CIS320Project.pdf

H.2 Implementation

A full implementation is available on the cd consisting of:

1. Java source files
   - D:\GA_CIS320\Implementation\OHSV\src\n
2. Java class files
   - D:\GA_CIS320\Implementation\OHSV\classes\n
3. A full documentation of the implementation generated in HTML format using javadocs
   - D:\GA_CIS320\Implementation\OHSV\javadoc\index.html

4. Other Java packages used
   - D:\GA_CIS320\Implementation\Java Packages\JAI\n   - D:\GA_CIS320\Implementation\Java Packages\JAMA\n   - D:\GA_CIS320\Implementation\Java Packages\JavaStatistics\n   - D:\GA_CIS320\Implementation\Java Packages\jexcelapi\n
H.3 Database

Oracle Database 10g Enterprise Edition Release 10.2.0.1.0 was used as an RDBMS to store the acquired signature database. A dump file namely EXPDAT.DMP was exported containing all database objects and is available on the CD (D:\GA_CIS320\Database\EXPDAT.DMP). However, the fields containing authors’ personal information and signature samples are set to null on purpose to protect the authors’ privacy. So, this data can only be used to execute the training and testing of the entire system. The following are the required steps to import the database

1. Install Oracle 10g Enterprise Edition Release 10.2.0.1.0
2. Create a database namely ORCL
3. Enable user SCOTT with a password TIGER
4. Load a DOS shell command prompt
5. Type the following command and press enter
   \texttt{D:GA\_CIS320\Database\import.bat SCOTT TIGER ORCL}

   The above steps will import the signature database used for this study.

\textbf{H.4 Results}

All results obtained by this study are available in the following zipped file:

- \texttt{D:GA\_CIS320\Results\Results.zip}
Bibliography


[32] International Association for Biometrics (IAFB) and International Computer Security Association (ICSA), ‘1999 Glossary of Biometric Terms’, URL: [http://www.afb.org.uk/docs/glossary.htm](http://www.afb.org.uk/docs/glossary.htm) [cited 15/03/2006].


Evaluation

I am very satisfied with the study carried out during the last months. I managed to implement an offline signature verification system tested for random forgeries. Although the original proposal was slightly deviated, as explained in Section 2.9.1, I managed to achieve promising results with a mixture of signature features. The various verification scenarios using different combination of features and signature samples helped to understand the behaviour and robustness of single-layer RBF architecture.

The research process was an essential process where I understood the problem of signature verification and analysed what have been covered and where the researchers are heading.

This study helped me to broaden my knowledge in several areas. For instance RBF neural networks were totally new for my knowledge and it took a number of days to understand how RBF architecture is implemented. Furthermore, Image Processing was another area where I lacked experience. Although, I am familiar with Java™ development, it required quite an effort to understand and apply Java Advanced Imaging API for this study.

The systematic approach proposed by the project guide helped me to complete the dissertation in a methodical way. My self-discipline was another ingredient which helped to focus and stick to my project plan, although there were some occasions where I delayed slightly a specific task.

The dissertation helped me to mature both technically and behavioural. From the technical aspect I broadened my technical knowledge especially in the area of image processing. On the other hand, from the behavioural aspect, I think I have improved significantly my writing skills and the way I approach new areas of studies.

I am very satisfied with my final report where I managed to document a detailed proposal of an innovative way to tackle the problem of offline signature verification. This study gave me a lot of satisfaction and it motivates me to further my studies in this field.