CERTIFICATE

This is to certify that the project report entitled **"Off-Line Cursive Handwritten Words Recognition using Soft Computing Techniques"** is a bonafide record of work carried out by **Mr. Amit Choudhary (2K12/CSE/28)** under my guidance and supervision in partial fulfillment of the requirement for the award of the degree of Master of Technology in Computer Science & Engineering from Delhi Technological University, Delhi.

Vinod Kumar
(Supervisor)
Associate professor,
Department of Computer Engineering,
Delhi Technological University
Delhi-110042

DECLARATION

I hereby declare that the Major Project-II Report entitled **"Off-Line Cursive Handwritten Words Recognition using Soft Computing Techniques"** which is being submitted to Delhi Technological University is a bonafide work carried out by me. The material contained in the report has not been submitted to any other university or institute for award of any degree or diploma. My indebtedness to other works has been duly acknowledged at relevant places.

Amit Choudhary

Roll No. 2K12/CSE/28

M.Tech (Computer Science and Engineering),
Department of Computer Science and Engineering,
Delhi Technological University,
Delhi-110042

ACKNOWLEDGEMENT

I feel besieged to spell few names those were the source of inspiration behind this project work and their help and support made me to accomplish this demanding task.

First and foremost, with a profound sense of gratitude, I acknowledge the valuable scrupulous guidance rendered by my supervisor **Mr. Vinod Kumar**, Associate Professor, DTU, who has not been just a guide but also an inspiring teacher, an eminent academician and an incredible human being. It has been an honor for me to be his student. To me, he is not just the supervisor of my project report but instead I look upon him as an architect of my academic destiny. Although it is very difficult to translate the feeling of regards and respect into words, yet I wish to extend my indebtedness to my supervisor whose motivating force behind me has given richness to my modest endeavor.

I would like to express my sincere thanks and gratefulness to **Dr. Rajni Jindal,** HOD, Computer Science and Engineering, DTU for her generous and erudite guidance and support. As the Head of Department, she has created an environment which encourages academic work and research activities. I express my deep sense of gratitude to her for the everlasting inspirations and encouragement given by her during my project work.

I would like to take this opportunity to extend my deep sense of gratitude and indebtedness to **Mr. Manoj Kuamr**, Associate Professor, DTU, who has been the motivating force and the source of inspiration for me during the project work.

I want to extend my special thanks to **Mr. R. K. Yadav**, Assistant Professor, DTU, for his encouragement and guidance throughout the project work.

I am also thankful to the faculty members, non-teaching staff and library staff of Delhi Technological University, Delhi for their help throughout the research work.

I acknowledge the continuous motivation and support provided by my family and friends during the completion of this project work.

Last but not the least, I am thankful to God who has always been the invisible divine force behind my achievements and I hope that He will continue to do so forever…

(Amit Choudhary)

ABSTRACT

Handwriting is a natural way to communicate and record information. A large amount of important historical data is written on papers. Machine simulation to recognize handwriting has opened new horizons to improve human-computer interface and perform repetitive task of reading by computers. Despite of more than four decades of intensive research, off-line unconstrained cursive handwriting recognition is still an open problem and an active area of research these days. Currently, accuracy of off-line cursive handwritten word recognition schemes is below acceptable level and is computationally expensive for real world applications. In a cursive handwritten word, touched characters are common and are the main cause of low segmentation and recognition accuracy.

This project report presents superior approaches for character image preprocessing, untouched as well as touched character segmentation and feature extraction for the purpose of handwritten words recognition. The performance evaluation of the two neural network classifiers has been done by fusion of the two feature extraction techniques. The techniques to optimize the training parameters of a back-propagation MLP are evaluated and common situations during BP learning with possible causes and potential remedies are also presented.

Preprocessing techniques include Grayscale conversion, Binarization, Global Thresholding and Background Noise Removal, Thinning and Skeletonization, Foreground Noise Removal, Character Image Cropping and Size Normailzation. Character segmentation from the word image is also a focus of this thesis because poor segmentation contributes to inaccuracy in character and word recognition processes. In this regard, two improved segmentation techniques are proposed and evaluated. The first technique is based on connected component analysis and is proposed to segment untouched characters in a word image and in the second technique, a heuristic vertical dissection based approach is proposed to segment touched characters in a word image. For feature extraction; Binarization technique and Projection Profile Techniques are proposed to extract features of a handwritten character / numeral image. For character

recognition, two variants of artificial neural networks, namely, Feed Forward Back-Propagation neural network and Radial Basis Function Neural Network are proposed and their performances are analyzed in terms of accuracy, speed and computational complexity.

Various techniques and strategies for optimizing the recognition accuracy of the classifier are also proposed in this research. All the experiments are evaluated on a local database of handwritten characters and digits. Favorable accuracy for each phase of the OCR system is achieved and reported in this research with high speed and accuracy with minimum computational complexity.

TABLE OF CONTENTS

DECLARATION ii
ACKNOWLEDGEMENT iii
ABSTRACT v
TABLE OF CONTENTS vii
LIST OF FIGURES x
CHAPTER-1 1
1 Introduction 1
1.1 Introduction to Pattern and the Act of Recognition 1
1.2 Introduction to Artificial Neural Networks 1
1.3 Applications of Artificial Neural Networks 2
1.4 Introduction to Handwriting Recognition 2
1.4.1 On-Line Handwriting Recognition 3
1.4.2 Off-Line Handwriting Recognition 3
1.5 Historical Development of an OCR System 3
1.6 Motivation 3
1.7 Objectives 4
1.8 Research Confinements and Scope of Improvement 4
1.9 Research Contribution 5
1.10 Organization of the Report 6
CHAPTER-2 7
2 Review of Literature 7
2.1 Introduction 7
2.2 Typical Handwriting Recognition System 8
2.3 Preprocessing 8
2.3.1 Thresholding 9
2.3.2 Noise Removal 9
2.3.3 Size Normalization 9
2.3.4 Skew Correction 9
2.3.5 Thinning and Skeletonization 9
2.4 Segmentation 9
2.5 Feature Extraction 11
2.6 Recognition 11

2.7 Conclusion....................13
CHAPTER-3.................... 14
3 Segmentation Techniques for Cursive Handwritten Words.................... 14
3.1 Introduction....................14
3.2 Challenges in the Proposed Segmentation Techniques....................14
3.3 Untouched Character Detection and Extraction from Off-Line Printed Text and Images....................15
3.3.1 Proposed Character Extraction Technique.................... 15
3.3.2 Implementation and Methodology.................... 16
3.3.3 Experimental Results and Analysis.................... 18
3.3.4 Conclusion.................... 20
3.4 Touched Character Segmentation from Off-Line Cursive Handwritten Words of Varying Length....................20
3.4.1 Proposed Technique and Methodologies.................... 21
3.4.2 Preparation of Handwritten Words Local Database.................... 21
3.4.3 Image Acquisition.................... 22
3.4.4 Preprocessing.................... 22
3.4.4.1 Thresholding and Binarization..................... 22
3.4.4.2 Skeltonization and Image De-Noising.................... 23
3.4.4.3 Cropping.................... 24
3.4.5 Proposed Segmentation Technique.................... 24
3.4.5.1 Overview.................... 24
3.4.5.2 Methodology.................... 24
3.4.5.3 The Problem of Over-segmentation.................... 25
3.4.5.4 Solving the Over-Segmentation Problem.................... 25
3.4.5.5 Implementation.................... 26
3.4.5.6 Result Analysis.................... 27
3.4.6 Conclusion.................... 28
CHAPTER-4.................... 30
4 Implementation of Back-Propagation Neural Network Classifier System for Off-Line Cursive Handwritten Character Recognition using Features Extracted from Binarization Technique.................... 30
4.1 Introduction....................30
4.2 Challenges in the Proposed Recognition Process....................30
4.3 Overall OCR System Design....................30

4.4 Character Image Acquisition 31
4.5 Local Database Preparation 31
4.6 Preprocessing 32
4.6.1 Grayscale Conversion 32
4.6.2 Binarization Technique 33
4.6.3 Noise Removal 34
4.6.4 Cropping 35
4.6.5 Size Normalization 36
4.7 Feature Extraction 36
4.8 Sample Preparation for Neural Network Training 37
4.9 Classification and Recognition Process 38
4.9.1 Methodology and ANN Architecture 38
4.9.2 Experimental Conditions 40
4.10 Implementation and Functional Details 40
4.11 Discussion of Results 44
4.12 Conclusion 45
CHAPTER-5 47
5 Implementation of RBF Neural Network Classifier System for Off-Line Cursive Handwritten Digit Recognition using Projection Profile Features 47
5.1 Introduction 47
5.2 Proposed Recognition System 47
5.3 Image Acquisition and Sample Preparation 47
5.4 Preprocessing 48
5.5 Feature Extraction 50
5.6 Classification and Recognition 51
5.6.1 RBF Neural Network as a Classifier 52
5.6.2 Training Sample Preparation 53
5.6.3 Implementation and Functional Details 53
5.7 Discussion of Results 55
5.8 Conclusion 58
CHAPTER-6 59
6 Conclusion and Future Scope 59
6.1 Conclusion 59
6.2 Future Scope 60
REFERENCES 61

LIST OF FIGURES

Fig.2.1 Typical Segmentation-Based Handwriting Recognition System 8
Fig.2.2 Categorization of Segmentation Based and Segmentation Free Approaches....... 10
Fig. 3.1 Printed and Handwritten Word Image Samples 16
Fig.3.2 Original Scanned Input Image having a Logo Image and Text........................... 17
Fig.3.3 The Input Image after Background Noise Removal and Contrast Adjustment.... 17
Fig.3.4 Extraction of Characters from the Input Image 18
Fig.3.5 Original Image and the Preprocessed Handwritten Word Image 19
Fig.3.6 Extracted Character Image where the Dot '.' of Character 'i' is Not Extracted .. 19
Fig.3.7 Original Image and the Preprocessed Digits Image 20
Fig.3.8 Two Connected Components are extracted for Character '5' 20
Fig.3.9 Handwritten Word Image Samples having Touched Characters.......................... 22
Fig.3.10 Input Scanned Handwritten Word Image 22
Fig.3.11 Word Image in Grayscale Format 23
Fig.3.12 Word Images in Binary Form 23
Fig.3.13 Word Image after Thinning 23
Fig.3.14 Noise Detection in Word Images.. 23
Fig.3.15 Word Image after Noise Removal 24
Fig.3.16 Cropped Word Images.. 24
Fig.3.17 Word Image Preprocessing (a) Input Scanned Word Images; (b) Word Images after Gray Scale Intensity Threshold; (c) Word Images in Binary Format; (d) Word Images after Thinning; (e) Cropped Word Images after Noise Removal......................... 26
Fig.3.18 Word Image Segmentation (a) Pre-processed Word Images; (b) Inverted Binary Images; (c) RGB Images; (d) Over-segmentation in Images; (e) Image after removing Over-segmentations; (f) Final Segmented Output Word Images 27
Fig.3.19 Word Images Showing All Type of Segmentation Errors................................. 28
Fig.4.1 Schematic Diagram of the Proposed Character Recognition System................... 31
Fig.4.2 Character Image Samples Contributed by a Single Writer.................................. 32
Fig.4.3 Input Scanned Handwritten Character Image.. 33
Fig.4.4 Handwritten Character Image in Grayscale Format 33
Fig.4.5 Handwritten Character Image in Binary Format 34
Fig.4.6 Noisy Handwritten Character Image 34
Fig.4.7 Noise Detection in Handwritten Character Image... 35
Fig.4.8 Handwritten Character Image after Noise Removal.. 35
Fig.4.9 Cropped Handwritten Character Images 36
Fig.4.10 Resized Handwritten Character Images 36
Fig.4.11(a) Binary Image of Character 'c' (b) Resized Binary Image of Character 'c'; (c) Binary Matrix representation and (d) Reshaped Binary Matrix or Feature Vector of Character 'c'.. 37

Fig.4.12 Matrix representation of Input Sample 38
Fig.4.13 Feed Forward Neural Network with one Hidden Layer 39
Fig.4.14 Architecture of the Neural Network used in the Recognition System 39
Fig.4.15 Development Environment for Neural Network Toolbox 41
Fig.4.16 Structure of the Network 42
Fig.4.17 Training Process of the Network 42
Fig.4.18 The Variation of MSE with the Training Epochs 43
Fig.4.19 A Confusion Matrix Representing the Performance of the Neural Network Classifier 44
Fig.4.20 Three Dimensional Plot of Confusion Matrix Representing the Performance of the Neural Network Classifier 45
Fig.5.1 Digit Samples Contributed by a Writer 48
Fig.5.2 Input Scanned Digit Image 48
Fig.5.3 Image after Background Noise Removal 49
Fig.5.4 Size Normalized Digit Image 49
Fig.5.5 Projection Directions for a 3×3 Image Pattern 50
Fig.5.6 Projection Profile of Digit '3' in Vertical, Horizontal, Left-Diagonal and Right-Diagonal Orientations 51
Fig.5.7 A General RBF Neural Network with One Output 52
Fig.5.8 Matrix Representation of Input Digit Sample of Size 94×10 53
Fig.5.9 The Development Environment for Neural Network Toolbox 54
Fig.5.10 RBF Network Creation Wizard 54
Fig.5.11 Structure of the RBF Network 55
Fig.5.12 RBF Network Performance with Training Epochs 55
Fig.5.13 Two-Dimensional Confusion Matrix Plot in MATLAB Environment 56
Fig.5.14Three-Dimensional Confusion Matrix Plot in MATLAB Environment 57

LIST OF TABLES

Table 2.1 Comparison of Segmentation Results 10

Table 2.2 Performance Comparison of Script Recognition Accuracy 12

Table 3.1 Segmentation Result of the Proposed Vertical Segmentation Approach 27

Table 4.1 Experimental Conditions during the Recognition Experiment 40

Table 5.1Confusion Matrix representing the Performance of the Classifier 56

CHAPTER-1

1 Introduction

1.1 Introduction to Pattern and the Act of Recognition

Pattern Recognition is an emerging and a very active branch of Artificial Intelligence dealing with the detection and identification of input pattern observations. Pattern Recognition can be defined as: “The act of inputting raw data and taking an action based on the category of the data”. Anything that has been perceived by the sensory units of a human body is termed as a pattern. Special information which is understood by our brain and the brain recognizes the input source is called a pattern.

Pattern recognition is a usual and ordinary act in our daily routine. Whatever we see, listen or touch any object, human brain tries its characteristics to be associated with once already stored in the brain. It recognizes the input pattern when a perfect match is discovered.

1.2 Introduction to Artificial Neural Networks

Artificial Neural Network (ANN) has been developed by the researchers inspired by the functionality, efficiency and capability of biological neural network present in human brain. All the activities performed by a human brain are simulated into a computer machine with the help of a complicated architecture of small functional units called Artificial Neurons. The interconnection of all these neurons is termed as Artificial Neural Network (ANN).

ANN tends to find the solution of a given problem in the same way as done by human brain by mimicking the logical reasoning and thought process of humans. Nodes available in the Artificial Neural Networks are interconnected by adjustable weights which can be updated to train the ANN.

1.3 Applications of Artificial Neural Networks

Artificial neural networks are extensively being used in the fields of “Classification”, “Signal Processing”, “Speech Recognition”, “Function Approximation”, “Intelligent Control”, “Financial Forecasting”, “Condition Monitoring”, “Process Monitoring and Control”, “Neuro-Forecasting”, “Pattern Analysis” and “Handwriting Recognition”.

1.4 Introduction to Handwriting Recognition

The importance of the piece of paper cannot be ignored in enhancing the people’s memory and in facilitating communication between people. This handwriting’s concept is very old and attributed by many civilizations and cultural ages. However, the solitary purpose is to facilitate communication and expand human memory. According to Plamondon and Srihari (2000), the terms “Handwriting" and “Handwriting Recognition” has been defined as:

“Handwriting is the task of transforming a language represented in its spatial form of graphical marks into its symbolic representation".

“Handwriting Recognition is a process which allows computers to recognize written or printed characters such as numbers or letters and to change them into a form that the computer can use for editing and searching.”

Machine simulation created a special attention for the scientists and consequently many researchers started efforts to mimic human behavior into machines to develop an Optical Character Recognition (OCR) system for recognizing printed or handwritten material. Capability of a computer system to extract and understand the input data from photos, paper document images etc. is called handwriting recognition.

In daily routine, people generally write on papers as it is easily available and is cost effective also. There is a great demand to transform this handwritten text into digital form so that the text can be digitally organized and searched as per requirement. Online and Offline handwriting are the two different domains in which Handwriting recognition has been divided.

1.4.1 On-Line Handwriting Recognition

It implicates the automatically conversion of editable text from the words written on a digitizer etc. Movement of pen tip such as moving the pen tip upwards or movement of pen tip downwards on a touch sensitive surface is picked up by a sensor unit.

1.4.2 Off-Line Handwriting Recognition

Off-line recognition of handwriting deals with automatically converting the printed text images into editable text which is used within computers. Offline recognition of handwriting is also termed as Optical Character Recognition because an optical device (camera/scanner) converts the input image into a bit pattern. The bit pattern data is then used for recognition task for machine printed as well as handwritten text.

Offline recognizers are beneficial because they can be used for recognizing historical documents which were written a long time back.

1.5 Historical Development of an OCR System

Historically, Tausheck in Germany is the first to obtain patent for OCR in 1929, closely followed by Handel in 1933. Later in 1951, David Shepard invents a robot reader-writer, followed by the invention of a prototype machine by Jacob Rabinow in 1954. However, these machines can only read uppercase printed characters with very slow speed of about one minute per character (Srihari, 1993). In early sixties, a number of well-known companies, including IBM, start research into OCR. Consequently, IBM comes out with new OCR product and is used for reading special and artificial fonts.

1.6 Motivation

The main reasons for continuing research in this area among the researchers and motivation behind this research are:

- Off-line cursive handwritten text recognition has many applications in real life.

- Machine simulation challenges such as preprocessing and proper segmentation during text analysis & recognition.
- Enhancement in the recognition accuracy by detection and removal of noise from the handwriting samples.

There is an intense boost in cursive handwriting recognition area; still it has been an active research direction till now. A fully functional automatic system is to be designed to explore new strategies and techniques for improved character segmentation and recognition which intern will improve accuracy and speed of the recognition system.

1.7 Objectives

Off-line cursive handwriting recognition has been the utmost difficult and thought-provoking problem in pattern recognition domain. Handwriting recognition involves reading script from static surface. Despite of achieving high recognition accuracy for machine printed word recognition, the same accuracy has not been attained for unconstrained handwriting recognition due to huge amount of variations present.

So the aim of this work is to do segmentation of touching characters in a handwritten word accurately and to recognize off-line cursive characters in handwritten text. Accordingly, to accomplish this aim, the main objectives of this research are:

- To develop and enhance preprocessing techniques for off-line cursive characters in handwritten text.
- To develop and investigate improved segmentation technique(s) for handwritten word images.
- To propose feature extraction technique(s) to recognize segmented characters and digits using MLP and RBF Networks.

1.8 Research Confinements and Scope of Improvement

Defining the research scope is crucial to attain the defined objectives and covered the following aspects:

- Global Thresholding technique is adopted.

- All experiments are conducted on a local database.
- Broken characters and those touching at two or more points are out of scope.
- Overlapped characters are out of scope.
- Skew removal and slant correction are out of scope as all the handwriting samples in the local database are free from skew and slant.
- All the experiments are conducted for handwritten words written in Roman Script.

1.9 Research Contribution

The contributions of this research are classified into following categories:

(a) Preprocessing

- A connected component based approach is proposed for foreground noise removal.

(b) Segmentation

- A character detection and extraction methodology has been proposed for segmenting the foreground character components (which are not touching each other) from captured handwritten word images or printed text on a colored background with noise and other unwanted image objects.
- Another character segmentation technique is proposed which enhances over-segmentation after thinning the contour of input word image to single pixel wide. Between any two characters, two or more than two consecutive segmentation columns are merged into a single column in the next step.

(c) Feature Extraction

- Two feature extraction techniques “Binarization” and “Projection Profile” are proposed to improve the recognition accuracy.

(d) Recognition

- A fully functional cursive handwritten word recognition system is depicted incorporating proposed preprocessing, segmentation and character recognition techniques.

1.10 Organization of the Report

The report consists of six chapters. Chapter 1 presents Introduction, historical development of OCR system, motivation, research objective, research scope and the contribution of the work done. Chapter 2 provides the literature review in brief. Chapter 3 describes two improved character segmentation strategies based on "connected component analysis" and "vertical dissection technique". Chapter 4 describes the binarization feature extraction and MLP classifier for recognizing the off-line unconstrained characters from the handwritten images. Chapter 5 describes the projection profile features and the RBF network as a classifier to recognize the off-line cursive handwritten digits. Chapter 6 concludes the achievements of this research and the overall off-line cursive handwriting recognition system. It also suggests plans and recommendation for work to be done in future.

CHAPTER-2

2 Review of Literature

2.1 Introduction

This chapter briefs the state-of-the-art in “handwriting recognition” research and techniques for “preprocessing”, “segmentation”, “feature extraction” and “recognition” of free style handwritten words.

Unconstrained handwriting recognition has been a stimulating research field for the last few decades. Many researchers all over the world has been involved for over a half a century and achieved amazing results for machine printed characters exceeding 95%. Remarkable improvement in recognition results for handwritten characters has been achieved exceeding 90% (Alginahi, 2010).

Current state-of-the-art in the field of “Off-line cursive handwritten words recognition” has been presented in this chapter. In recent years, significant growth has been reported in this field. A huge number of research papers have been published in the literature with focus on novel strategies for classification task of handwritten characters, numerals and words (“Fujisawa et al., 1992; Wang et al., 2005; Britto Jr et al., 2004; Gader et al., 1997; Blumenstein et al., 2004; Suen and Tan, 2005; Marinai et al., 2005; Liu and Fujisawa, 2005; Yanikoglu and Sandon, 1998; Dimauro et al., 1998; Xiao, X. and Leedham, G. 2000; Chiang, 1998; Martin et al., 1993; Eastwood et al., 1997; Srihari, 2006; Gilloux, 1993; Plamondon and Srihari, 2000; Suen et al., 1993; Cho, 1997; Casey and Lecolinet, 1996; Dunn and Wang, 1992; Lu, 1995; Lu and Shridhar, 1996; Elliman and Lancaster, 1990”).

In literature, few researchers achieved exciting recognition accuracy for isolated digits/characters but due to the complicated shape of handwritten words and various challenges during segmentation of word images, the segmentation and classification results has not been found reasonable ("Verma et al., 2001; Gunter and Bunke, 2004; Vinciarelli et al., 2003; Verma et al., 2004; Arica and Yarman-Vural, 2002; Camastra and Vinciarelli, 2003; Hanmandlu et al., 2003; Günter and Bunke, 2005; Viard-Gaudin et al., 2005; Schambach, 2005; Chevalier et al., 2005; Lee and Coelho, 2005; Gatos et al., 2006a; Koerich et al., 2006; Xu et al., 2003; Wen et al., 2007; Kapp et al., 2007; Blumenstein and Verma, 2001; Gang et al., 2002; Blumenstein et al., 2003; Verma, 2003; Fan and Verma, 2002").

2.2 Typical Handwriting Recognition System

Various steps involved in a "handwriting recognition system" include (a) "Digitization/Image acquisition", (b) "Preprocessing", (c) "Segmentation" (d) "Feature Extraction" and (e) "Recognition/Classification" as shown in Fig.2.1.

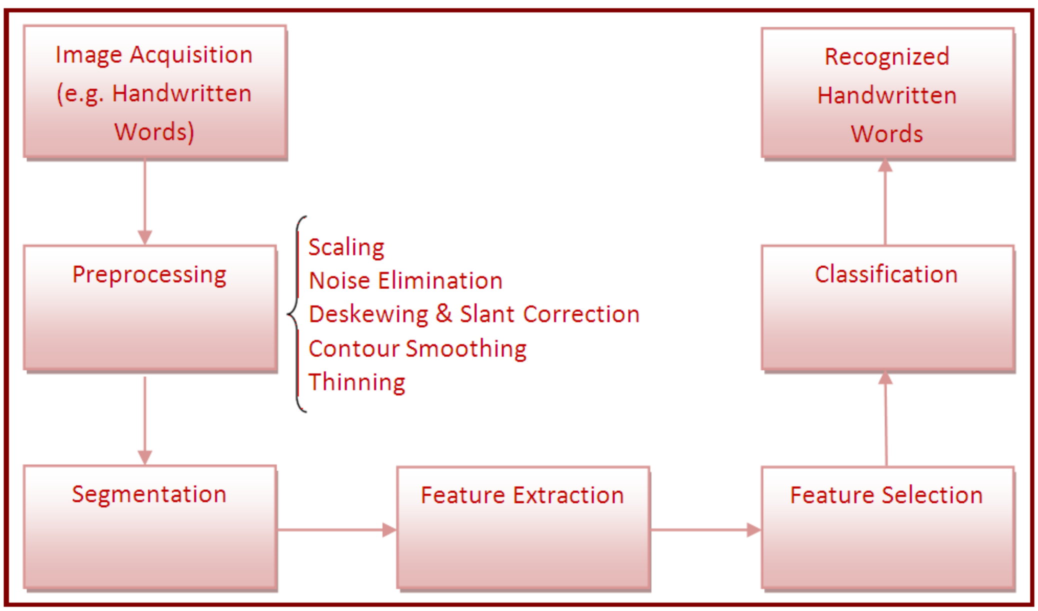

Fig.2.1 Typical Segmentation-Based Handwriting Recognition System

2.3 Preprocessing

The main objective behind preprocessing is to remove the invariabilities existing in word images. While scanning the handwritten word images, the quality may be ruined as the noise is introduced due to dust or due to colored background.

2.3.1 Thresholding

By comparing many global preprocessing techniques, Fischer confirmed “the Otsu method (Otsu, 1979) is preferred in document image processing (Fischer, 2000) and the Otsu method (Otsu, 1979) is one of the widely used techniques used to convert a grey-level image into a binary image then calculates the optimum threshold separating those two classes so that their combined spread (intra-class variance) is minimal (Alginahi, 2010)”.

2.3.2 Noise Removal

Noise may easily be picked up while capturing a picture or scanning an image (Verma and Blumenstein, 2008). Many researchers have worked in the area of noise removal (“Madhvanath et al., 1999; Dimauro et al., 1997; Chen et al., 1992; Kim et al., 1999”).

2.3.3 Size Normalization

Scaling is very important to deliver images of same sizes (Burges et al., 1992; Verma and Blumenstein, 2008).

2.3.4 Skew Correction

Some skew cannot be avoided while scanning manually (“Sarfraz and Rasheed, 2008; Sadri and Cheriet, 2009; Saba et al., 2011”).

2.3.5 Thinning and Skeletonization

Thinning erodes a character image until it becomes one-pixel wide. It produces skeleton of the character image to ease the recognition process. (Russ, 2007; Alginahi Y, 2010; Davies, 2005).

2.4 Segmentation

Segmentation techniques available in the literature can be classified into 3 types named “Explicit Segmentation”, “Implicit Segmentation” and “Holistic Segmentation” approaches as displayed in Fig.2.2.

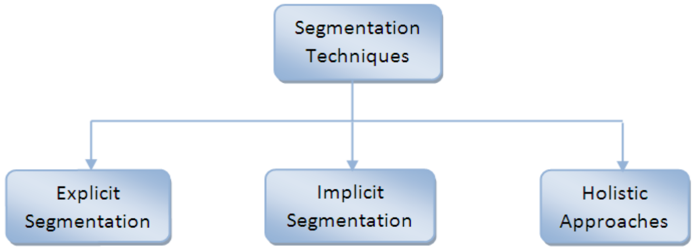

Fig.2.2 Categorization of Segmentation Based and Segmentation Free Approaches

A comparison of segmentation results available in the literature has been summarized in Table 2.1.

Table 2.1 Comparison of Segmentation Results

Author	Segmentation Approach	Segmentation Rate (%)	Comments
Blumenstein and Verma (1997)	ANN+conventional Method	81.21	800 Words
Verma and Gader (2000)	Feature based+ANN	76.52	Words number not mentioned
Verma et al. (2001)	Fusion of multiple Word recognition Techniques	86	317 words used for Testing
Blumenstein and Verma (2001)	Feature based+ANN	78.85	Words number not Mentioned
Verma (2002)	Feature based+ANN	84.87	300 test words only
Cheng and Blumenstein (2005a)	Ligature detection+ ANN	84.19	317 test words
Samrajya et al. (2006)	Hypergraph+ ligature analysis	Not mentioned	Not mentioned
Rehman and Dzulkifli (2008)	Ligature and shape analysis	88.21	2,936 words

The segmentation algorithms in the literature has been designed for segmenting touching character present in the handwritten words of English language and may not work well if applied to some other languages such as Arabic or Chinese. (“Plamondon and Srihari, 2000; Blumenstein and Verma, 2001; Vinciarelli, 2002; Gang et al., 2002; Koerich et al., 2003; Bortolozzi et al., 2005; Rehman and Dzulkifli, 2008; Saba, 2012”).

2.5 Feature Extraction

Extracting better features are directly correlated with the better recognition results. Projection Profile features (Desai, A., 2010) proved their importance even if the input samples are lacking some preprocessing techniques such as smoothing or thinning.

During Feature Extraction, beneficial information is extracted from the patterns and this information is used in the recognition process. (“Mitrpanont and Limkonglap, 2007; Blumenstein et al.,2007; Verma, 2003; Verma et al., 2004; Koerich et al., 2003; Kim et al., 2000; Vamvakas et al., 2007; Kang and Kim, 2004; Cavalin et al., 2006”)

2.6 Recognition

In a typical OCR system, this step is the most important. The classifier selected to do the classification also plays a crucial role in recognition process. Statistical and intelligent techniques are the two types of recognition techniques found in the literature.

Many effective recognition methods are based on statistical classifiers i.e. k-nearest–neighbor (K-NN), Bayesian classifier, template matching etc. (Liu and Fujisawa, 2005). Analytical approaches for recognition of characters/digits have also been reported by the researchers in the literature (“Fujisawa, 2005; Gatos et al., 2006b”). Recently, ANN classifiers are verified as an influential and fruitful for character/word recognition (“Verma et al., 2004; Blumenstein et al., 2007”).

A Summary has been prepared from the review of work done by various researchers in the literature and is shown in Table 2.2.

Table 2.2 Performance Comparison of Script Recognition Accuracy

Author	Classifier	Lexicon Size (in words)	Problem Domain	Recognition Rate (%)
Guillevic and Suen (1998)	HMM/KNN	30	LA words(ENG)	86.7
Chiang (1998)	NN	100	USPS database mail	87.4
Kim et al. (2000)	HMM/ MLP	32	LA words	92.2
Oliveira et al. (2002)	MLP	12	Numerical strings	87.2
Kundu and Chen (2002)	HMM	100	Postal words	88.2
Koch et al. (2004)	MLP	1,000	Letters (FR)	67.8
Günter and Bunke (2004)	HMM+Ensembled methods		IAM	71.58
Günter and Bunke (2005)	HMM+Ensembled methods		IAM	75.61–82.28
Koerich et al. (2005)	HMM		SRTP	77.62–99.29
Schambach (2005)	HMM		Siemens	60
Koerich et al. (2006)	HMM		SRTP	78
Gatos et al. (2006a)	K-NN	3,799	IAM	81.05
Gatos et al. (2006b)	SVM		IAM	87.68
Tomoyuki et al. (2007)	Posterior probability/ DP	1,646	City names (European countries)	80.2

2.7 Conclusion

In this chapter, a state-of-the-art in off-line cursive script recognition has been presented and the drawn conclusion is “The research is almost matured in the area of numeral recognition however the problem of cursive character recognition remains very much an open problem and it is mainly due to the presence of noisy, broken, multi-stroke, incomplete and ambiguous characters in scanned word images & to handle this type of problem, new feature extraction/selection techniques and multistage classifiers are desired”.

CHAPTER-3

3 Segmentation Techniques for Cursive Handwritten Words

3.1 Introduction

The smallest unit of a language is a single character and the separation of individual characters is very important step of any OCR (Optical Character Recognition) System. The OCR system converts the document image obtained after scanning into an editable form. Accuracy of OCR system is dependent on the segmentation technique employed.

Character segmentation & recognition are still an open problem for the researchers towards developing an efficient OCR system. Three steps in an OCR system are preprocessing, segmentation and recognition (Camastra, 2003; Fujisawa, 2005).

3.2 Challenges in the Proposed Segmentation Techniques

The difficulties during the segmentation process can be listed as:

- Two or more characters in a word can be touching each other in a word image.
- Some characters (e.g. ‘u’ and ‘v’) in a handwritten word image can have similar contours.
- Some characters can give the illusion of presence of two similar characters e.g. ‘w’ can be segmented into two ‘v’ and ’v’ characters.
- Two consecutive handwritten characters ‘i’ and ‘i’ may not get segmented and give the illusion of presence of character ‘u’.

This chapter proposes two techniques for segmentation and extraction of individual characters from handwritten word images or printer word images.

3.3 Untouched Character Detection and Extraction from Off-Line Printed Text and Images

A robust segmentation technique is proposed for the purpose of segmenting characters which are not touching or connected to each other in the input scanned handwritten word images. The input word images can have any number of characters and can have some images around the handwritten or printed text. The segmentation technique here involves the connected component analysis of the foreground components (objects) in the binary or black and white input image of the printed/ handwritten words. Along with the words, there can be some images. The number of characters in these words can vary i.e. the word length is variable. Various cursive font styles of different sizes can be used here. Excellent results are achieved which confirms the strength of the proposed untouched character detection & extraction procedure.

3.3.1 Proposed Character Extraction Technique

The proposed segmentation technique selects the foreground objects/regions available in the scanned word image or text meeting the norm of having foreground area greater than a particular threshold value. The foreground objects in the input scanned image having area more than 50 pixels and lesser than 1000 pixels are legal character objects. The foreground objects having area below 50 pixels are considered to be noise whereas the foreground objects with area more than 1000 pixels are considered as unwanted pictures or logo printed in the input document image / banner. Proposed character detection and extraction technique involves following steps:

- Step 1. The document picture (having logo/ images and handwritten/ typed words) is captured for extracting individual characters.
- Step 2. The captured image is converted into binary format.
- Step 3. The connected components in the binary image are identified with the help of “bwlabel” function available in MATLAB after inverting the binary image.
- Step 4. All foreground connected components (objects) are extracted by MATLAB’s “regionprops” method of MATLAB is used for identification and extraction of all possible objects (connected components) in the foreground.

Step 5. For each connected component identified in the captured picture, steps 5 to 7 are repeated.

Step 6. An identified connected component will be termed as valid object if its area is more than 50 pixels and less than 1000 pixels. A minimum possible rectangular region having minimum area and enclosing the whole object (connected component) is marked on every identified connected component.

Step 7. Each and every small image component having a complete single connected component enclosed with in a rectangular area is inverted. This inverted sub-image is presented as an individual segmented image of a single character.

3.3.2 Implementation and Methodology

Some pictures (having logo/ images and handwritten/ typed words) are captured using a camera/scanner. Font Size and font color in these captured images can be different. Also, these images may be printed on backgrounds with noise/color. Some scanned word images having handwritten or printed words are shown in Fig.3.1.

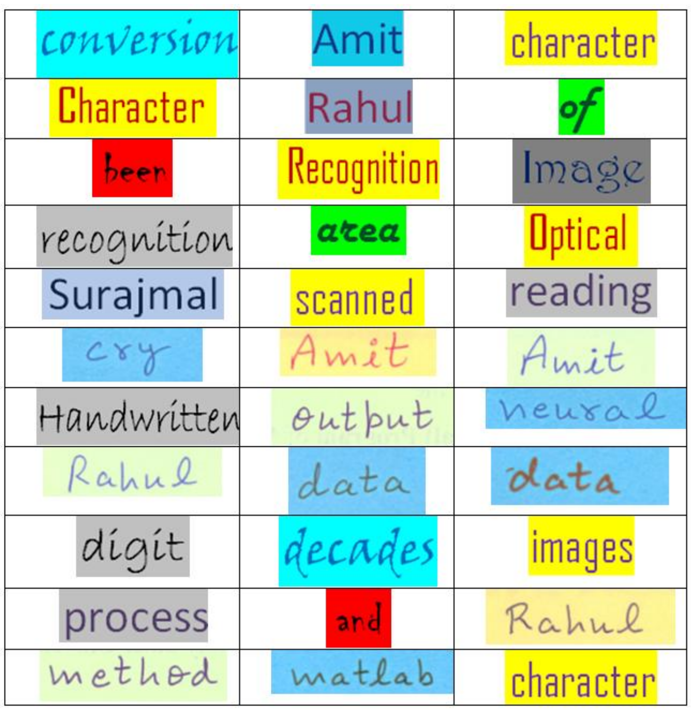

Fig. 3.1 Printed and Handwritten Word Image Samples

The captured images are having a noisy/colored background. Elimination of this background noise/color is necessary for improving the picture quality before further processing. It can be observed from Fig. 3.1 that the handwritten words are written with different colored ink and machine printed words are printed with different colored ink. To overcome the issues raised due to different colors, it is necessary to do the contrast adjustment. Fig.3.2 displays the originally captured image holding a logo and some printed text whereas Fig.3.3 displays the resultant image obtained after removing the background noise and post binarization. For binarization process, the method employed here is called the grayscale intensity threshold.

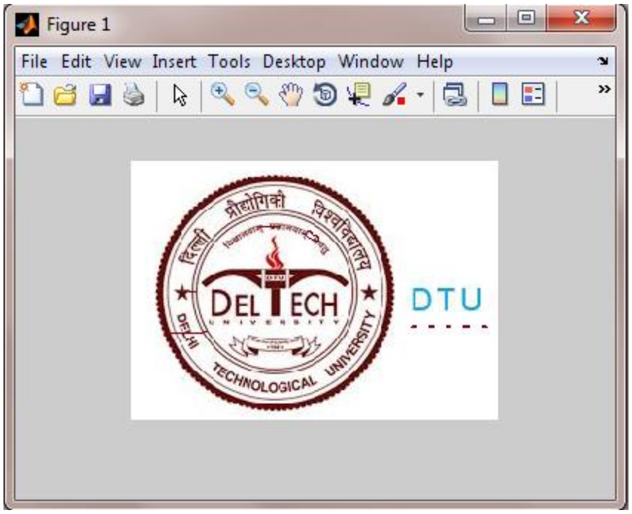

Fig.3.2 Original Scanned Input Image having a Logo Image and Text

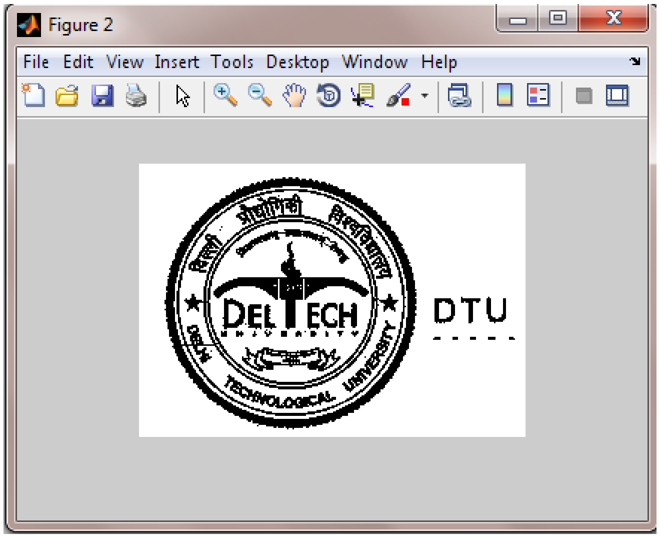

Fig.3.3 The Input Image after Background Noise Removal and Contrast Adjustment

The character extraction technique has been applied on the captured image shown in Fig.3.3. This image has been traced vertically from top to bottom starting from upper left corner to identify all the valid connected compomemts depending upon their rectangular area in the foreground. An identified connected component will be termed as valid object if its area is more than 50 pixels and less than 1000 pixels and are enclosing individual segmented character sub-image as shown in Fig.3.4.

Fig.3.4 Extraction of Characters from the Input Image

It is important to note that these individual character images have been cropped very well. If a recognition process has also to be carried out (as in a typical OCR system), these character images can be further pre-processed and can be used for training an ANN (Artificial Neural Network) classifier.

3.3.3 Experimental Results and Analysis

The proposed technique ensures to extract every possible character sub-image by valid rectangular box boundaries as shown in Fig.3.4. A connected component is said to be valid depending upon their rectangular area in the foreground. To become valid, it must have its area lying between maximum and minimum limits specified as per the extraction technique.

This is the reason that the dot(.) in character “i” was not extracted because it was not identified as a valid connected component as the area of its bounding box is below minimum limit (50 pixels) as depicted in Fig.3.6.

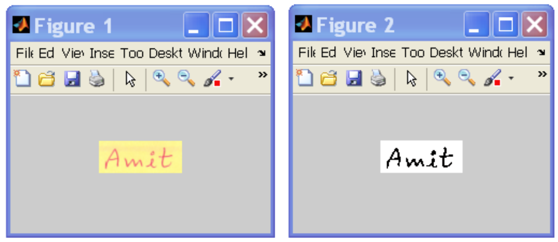

Fig.3.5 Original Image and the Preprocessed Handwritten Word Image

Fig.3.6 Extracted Character Image where the Dot '.' of Character 'i' is Not Extracted

The dot (.) in character "i" can be extracted and can be identified as a valid connected component if the least area requirement limit is further reduced to become less than 50 pixels but there are chances that some unwanted noise may picked up from the input scanned text image. The criterion to set maximum and minimum area limit for deciding a valid connection component can be customized as per application requirement. For noise free input images, the lower limit for area requirement may be fixed much lower than 50.

It can also be noted that this character extraction techniques fails when the characters in the input scanned image are broken i.e. when they are split into two or more connected components as shown in Fig.3.7. For perfect extraction, a character must be represented by a single connected component. It is displayed in Fig.3.8 that sub-image of character image "5" is split into 2 connected components where both of them are valid as they followed the minimum and maximum area requirement.

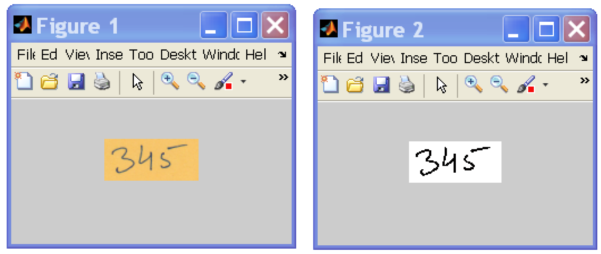

Fig.3.7 Original Image and the Preprocessed Digits Image

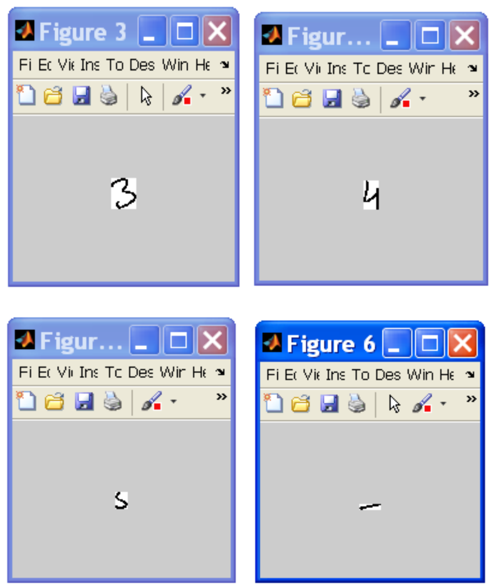

Fig.3.8 Two Connected Components are extracted for Character ‘5’

3.3.4 Conclusion

Excellent character extraction results are obtained by this technique. Apart from having a few drawbacks in case when character images are broken, the proposed technique can be employed in various applications such as road signboard recognition and vehicle number plate recognition etc. where an OCR system is trained and tested from these extracted images.

3.4 Touched Character Segmentation from Off-Line Cursive Handwritten Words of Varying Length

Segmentation of individual characters from a scanned word image is the most critical step of a typical OCR (Optical Character Recognition) System. A robust segmentation algorithm is proposed here. The word images are segmented into individual characters after skew angle correction and the thinning process, to get the single pixel stroke width. Ligatures of the touching characters are detected by keeping in view the geometrical shape of the English alphabets. The proposed vertical segmentation technique is used to cut individual characters from the handwritten cursive words. The proposed algorithm delivers excellent segmentation accuracy when tested on a local database.

3.4.1 Proposed Technique and Methodologies

Now a day, researchers are trying to introduce human brain's intelligence and capability into a computer system to recognize the information written on paper. In an OCR system, good character recognition accuracy can be achieved if the characters in the handwritten script are well segmented. Many researchers had already achieved very good segmentation results (Tan, J. et al., 2012) but the scope of improvement is always there and superior segmentation results are always awaited. Technological Advancements during the last 40 years in the area of document and character recognition is presented (Fujisawa, H., 2008). A new technique to recognize handwritten as well as typewritten English text is presented (Saeed and Albakoor, 2009). It does not require the thinning process and it delivered 80% accuracy. The slant and skew correction were not performed by the authors during preprocessing of the word images.

3.4.2 Preparation of Handwritten Words Local Database

To demonstrate the proposed segmentation algorithm, handwritten word samples written on colored or noisy background has been collected from 10 different persons aged between 15-40 years. From the collection of handwriting samples, we have selected 200 handwritten words randomly to perform the proposed experiment. Fig.3.9 displays few samples from the local handwritten word images database.

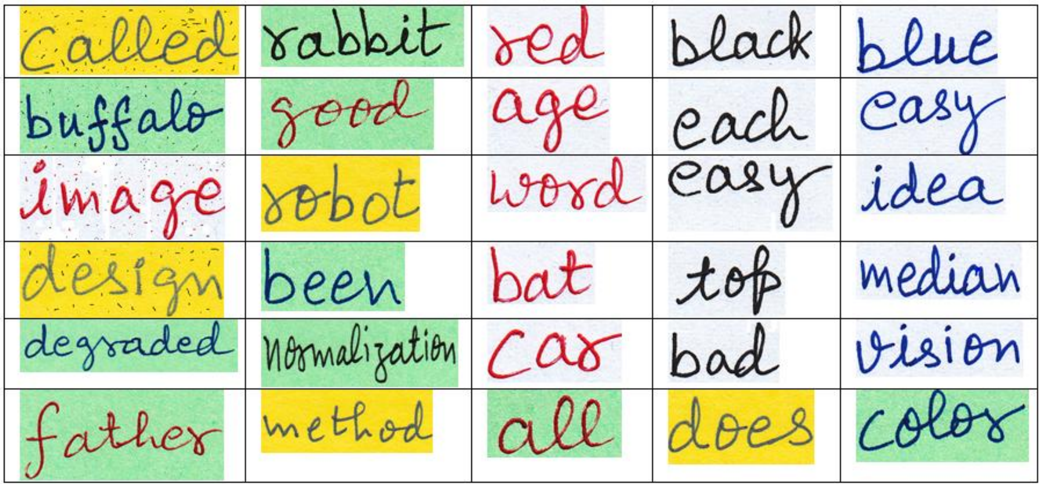

Fig.3.9 Handwritten Word Image Samples having Touched Characters

3.4.3 Image Acquisition

In image acquisition, a digital photo camera or a scanner is generally used to capture the handwritten word images and these images are saved in .bmp or .jpg file format for pre-processing. Fig.3.10shows three such image samples from the database.

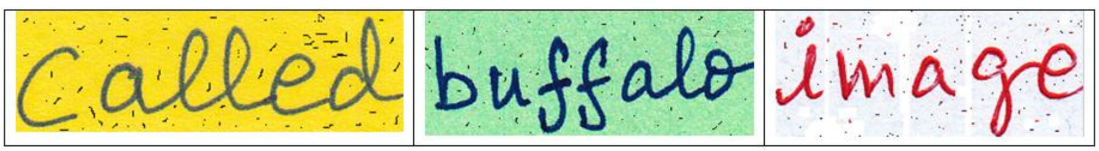

Fig.3.10 Input Scanned Handwritten Word Image

3.4.4 Preprocessing

The main objective behind preprocessing is to remove the invariabilities existing in word images. While scanning the handwritten word images, the quality may be ruined as the noise is introduced due to dust or due to colored background.

3.4.4.1 Thresholding and Binarization.

Thresholding is necessary so that the problems can be avoided due to usage of pen of different colored ink on colored and noisy surfaces. Fig.3.11shows three such grayscale images obtained after thresholding. The grayscale images are then transformed to the binary matrix form in which a 0 represents a black pixel in the foreground and a 1 represents a white pixel in the background. Fig.3.12 displays such binary images.

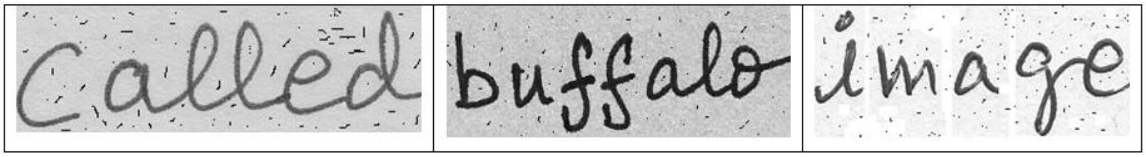

Fig.3.11 Word Image in Grayscale Format

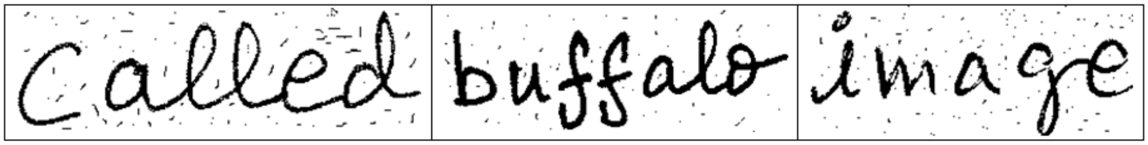

Fig.3.12 Word Images in Binary Form

3.4.4.2 Skeltonization and Image De-Noising

As the pens of different stroke width can be used by the different writers, a lot of unevenness may exist. After the thinning process, all the handwritten word images were made to have stroke-width of 1 pixel each. Three such image samples following the skeletonization process are displayed in Fig.3.13.

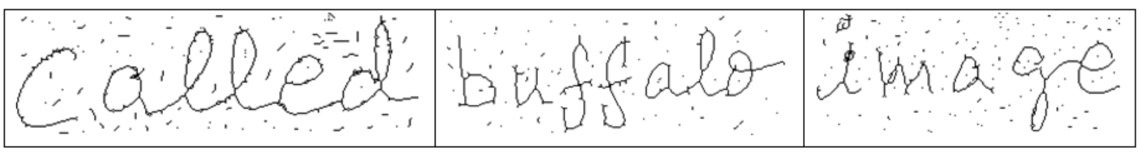

Fig.3.13 Word Image after Thinning

In "De-Noising" preprocessing stage, the noise (small foreground components and dots) induced in the image scanning process have been optimally eliminated. Only the noise dots and other foreground components have been removed in this step while retaining the character components as shown in Fig.3.14. Noise free images are shown in Fig.3.15.

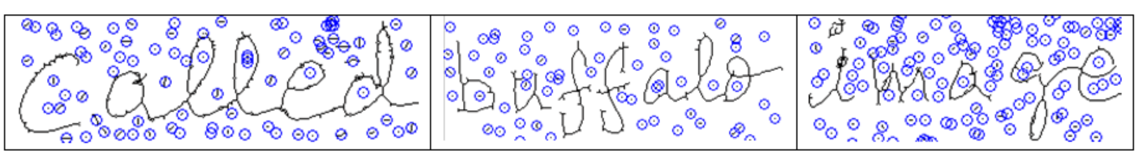

Fig.3.14 Noise Detection in Word Images

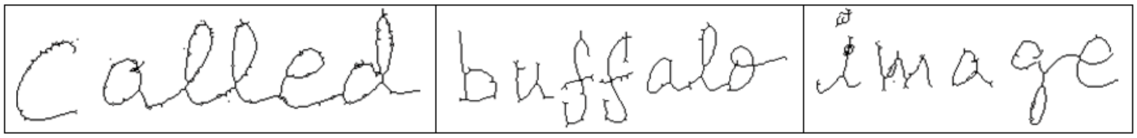

Fig.3.15 Word Image after Noise Removal

3.4.4.3 Cropping

The images after de-noising has been cropped to remove the extra space available around the rectangular region enclosing the handwritten noise free word image (Marti and Bunke 2002). Fig.3.16 shows such cropped sample images.

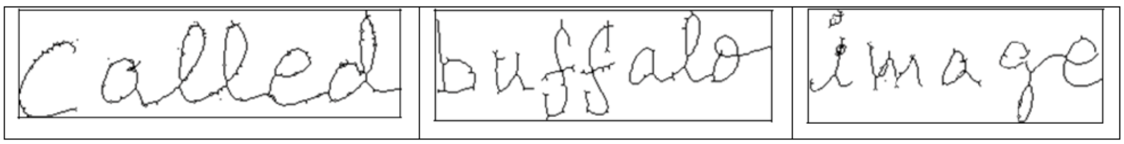

Fig.3.16 Cropped Word Images

3.4.5 Proposed Segmentation Technique

The projected segmentation algorithm has been designed for segmenting touching character present in the handwritten words of English language and may not work well if applied to some other languages such as Arabic or Chinese.

3.4.5.1 Overview

English language has Closed as well as Open characters. Closed characters have a semi-loop or a loop such as 'g', 'o', 'p', 's', 'a', 'b', 'c', 'd', 'e', etc. Open characters don't have any semi-loop or loop e.g. 'u', 'v', 'w', 'm', 'n' etc. Discriminating between ligatures and character segment is very hard in open characters. Ligature may be defined as a link between two or more consecutive characters used to join them. In written English language words, two 'i' characters side by side may look like 'u' and vise versa. Successive 'n' & 'i' may appear as 'm'. Character 'w' may give the illusion of presence of two characters 'i' & 'u'.

3.4.5.2 Methodology

After inverting the handwritten word image, the number of white pixels has been counted in each column scanning the image from top to bottom. The columns having 1 or 0 as the count of white foreground pixels are termed as CSC (Candidate Segmentation

Columns) and their positions have been noted. Fig.3.18(d) shows all such acknowledged columns.

3.4.5.3 The Problem of Over-segmentation

Several successive CSCs have been grouped together at various places in the handwritten word image and resulting in a situation called ‘over-segmentation’ and is displayed in Fig.3.18(d). There are three situations in which this problem of over-segmentation occurs. First, when there is a gap between two successive characters and for each column that lay in this gap, the count of the number of white pixels is 0. Second, when there is a ligature between two characters and the sum of white pixels is 1 for all columns through such ligatures in the whole word image. Finally, when there exist characters such as ‘u’, ‘m’, ‘n’, ‘w’ etc. which contains loop or semi-loop and the count of all the white foreground pixels for each column which crosses the ligatures-within-characters is also 1. Hence such types of characters are over-segmented.

3.4.5.4 Solving the Over-Segmentation Problem

In the situations, when there is a gap between successive characters, each and every CSC in this gap will have 0 white pixels. By taking mean of all CSCs lying in that gap and merging all the CSCs to a sole column, over-segmentation problem has been solved. In other situations, when ligature-within-character is present (e.g. characters ‘u’, ‘v’, ‘m’, ‘w’ etc.) or a ligature connecting two successive characters; a mean of all those CSCs in a group are calculated which are within a distance below threshold range and these CSCs are merged to a sole segmentation column.

In horizontal direction, the least gap between successive CSCs is called threshold range and its value is selected in such a manner that it should be less than the thinnest available character’s width such as ‘l’, ‘i’ etc. By repetitive experiments performed many times, threshold’s value is selected as ‘8’. Hence, all the CSCs that are within the 8 pixels range distance from another CSC would be merged into a single segmentation column.

3.4.5.5 Implementation

The handwritten word images obtained after various pre-processing steps as shown in Fig.3.18(a) has been complemented and taken as input to the segmentation algorithm. By inverting the input black & white images, black pixels form the background and white pixels form the foreground has been displayed in Fig.3.18(b). White pixels have been represented by 1 and it is now easy to count the number of white pixels in each and every vertical column of the binary handwritten word images. Now, this binary image is converted to the RGB color arrangement and is displayed in Fig.3.18(c). It is convenient if we show CSCs in any color (say red) other than black & white as shown in Fig.3.18(d). It can be clearly seen that every column, whose total count of white pixels is zero or one, vertically dissects the word image and has been termed as a CSC (Candidate Segmentation Column). All CSCs lying within the threshold range of 8 pixels from one another, are fused together to draw a single column representing that particular group of CSCs and is called Segmentation Column and is indicated by the Fig.3.18(e). Now, the image is then inverted again to get the white background and black foreground for the final segmented handwritten word image as displayed in Fig.3.18(f).

Fig.3.17 Word Image Preprocessing (a) Input Scanned Word Images; (b) Word Images after Gray Scale Intensity Threshold; (c) Word Images in Binary Format; (d) Word Images after Thinning; (e) Cropped Word Images after Noise Removal

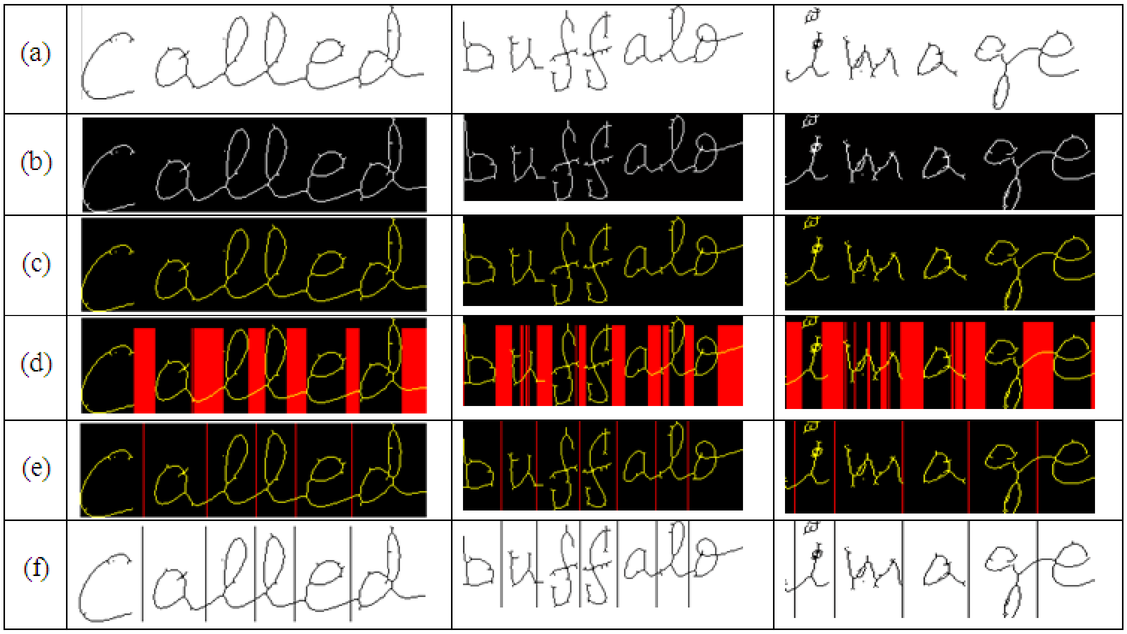

Fig.3.18 Word Image Segmentation (a) Pre-processed Word Images; (b) Inverted Binary Images; (c) RGB Images; (d) Over-segmentation in Images; (e) Image after removing Over-segmentations; (f) Final Segmented Output Word Images

3.4.5.6 Result Analysis

A random selection of 200 word images contributed by 10 different writers was used in this experiment. To evaluate the proposed segmentation technique, 3 types of errors were considered i.e. number of bad-segmented, over-segmented and miss-segmented words out of a total of 200 words used in the experiment.

Table 3.1 Segmentation Result of the Proposed Vertical Segmentation Approach

Total Number of Handwritten Words	Total Correctly Segmented Words (%)	Total Incorrectly Segmented Words (%)	Number of Words with various Segmentation Errors		
			Over-Segmented	Miss-Segmented	Bad-Segmented
200	167 (83.5%)	33 (16.5%)	14	5	24

Table-3.1 shows that 167 words were segmented correctly and 33 words were segmented incorrectly. Some incorrectly segmented words were bad-segmented as well as over-segmented and are counted in each type of error category while displaying the results in the Table-3.1. This is why 14+5+24 $\neq$ 33.

In the same way, for some word images, right point of segmentation has been missed and found to be shifted to another place which results in a bad segmentation point. Some "over-segmented" or "miss-segmented" or "bad-segmented" word images has been shown in Fig.3.19.

Fig.3.19 Word Images Showing All Type of Segmentation Errors

Comparing the results attained by the proposed segmentation technique with the results of other segmentation techniques developed by other researchers in the literature, is not so easy because different researchers presented their segmentation results under different constraints and also they used different types of databases. Some researchers made the assumption that the word images are noise free while some researchers gathered the word image samples from different number of contributors. Although, some authors (Marti and Bunke, 2002; Hull, J.J, 1994) used popular benchmark databases like IAM and CEDAR but they selected different number of handwritten word images from these databases and they even rejected some particular complicated word images from the database as per their personal choices.

3.4.6 Conclusion

The proposed technique ensures to dissect each and every possible character boundary by over-segmenting the sample word image enough number of times. Another strategy is also adopted that detects groups of many candidate segmentation points that are lying between any two successive characters and then clubs them into a single segmentation point. Whenever a word image contains untouched characters, accurate segmentation is guaranteed by the proposed technique. It performs very well to dissect ligatures connecting two successive closed characters. This technique sometimes over-segments the open characters because the ligature-within-characters looks like ligature connecting two characters. The segmentation accuracy of 83.5% delivered by the proposed segmentation technique is quiet excellent but the scope of improvement is always there. In future work, there is a need to improve some of the pre-processing techniques e.g. thinning etc.

CHAPTER-4

4 Implementation of Back-Propagation Neural Network Classifier System for Off-Line Cursive Handwritten Character Recognition using Features Extracted from Binarization Technique

4.1 Introduction

Out of the various steps involved in the process i.e. “preprocessing”, “feature extraction” and “classification”, the most crucial step towards this automation is “Feature Extraction”. Initially preprocessing is done for improving extracted features. The main objective behind preprocessing is to remove the invariabilities existing in character images.

4.2 Challenges in the Proposed Recognition Process

Challenges that came across during the recognition of handwritten character are described below:-

- Different writers have different style of handwriting and shape of handwritten characters.
- Size of characters written by different people is different.
- The shape of characters written by a single writer may be different.
- The background of the handwritten character images may be colored or can have noise.
- Color of the characters may be different depending on the color of the ink used to write the character.

4.3 Overall OCR System Design

Following steps are involved in the proposed character recognition system and are shown in Fig.4.1.

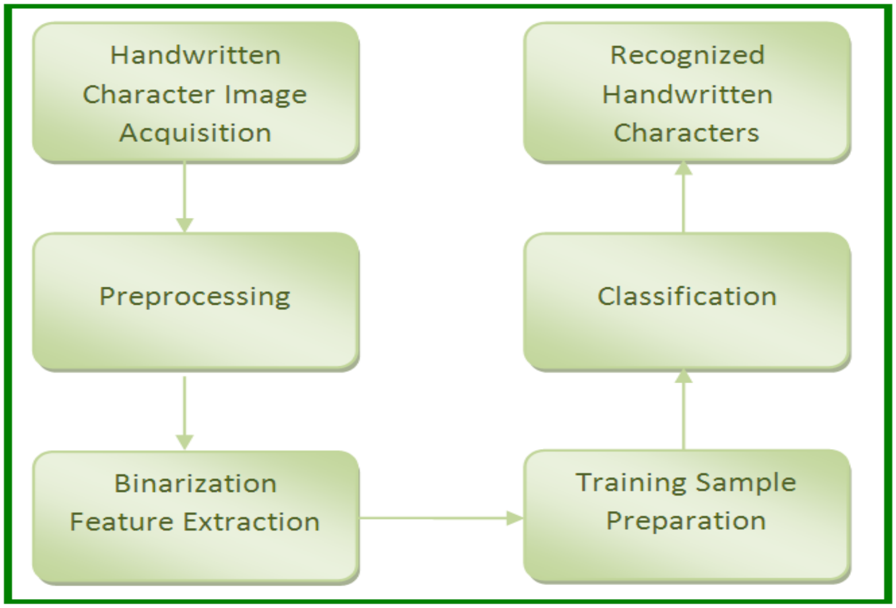

Fig.4.1 Schematic Diagram of the Proposed Character Recognition System

4.4 Character Image Acquisition

In image acquisition, a digital photo camera or a scanner is generally used to capture the handwritten word images and these images are saved in .bmp or .jpg file format for pre-processing.

4.5 Local Database Preparation

Five different authors aged between 15 to 50 years contributed five samples each where a sample consists of an entire English alphabet i.e. a -z. Hence, 650 (5×5×26=650) samples of handwritten character images have been collected. An additional 650 character image samples have been collected by segmenting the individual characters from the word images.

Thus, a data base of total collected 1300 characters has been used for feature extraction and recognition experimental setup. Five characters samples contributed by a single author are shown in Fig.4.2.

Fig.4.2 Character Image Samples Contributed by a Single Writer

4.6 Preprocessing

The main objective behind preprocessing is to remove the invariabilities existing in character images. While scanning the handwritten word images, the quality may be ruined as the noise is introduced due to dust or due to colored background.

4.6.1 Grayscale Conversion

Black and white images in which there are many shades of Gary ranging from black to white are called gray scale images. These images are different from binary images as there only two shades i.e. black and white.

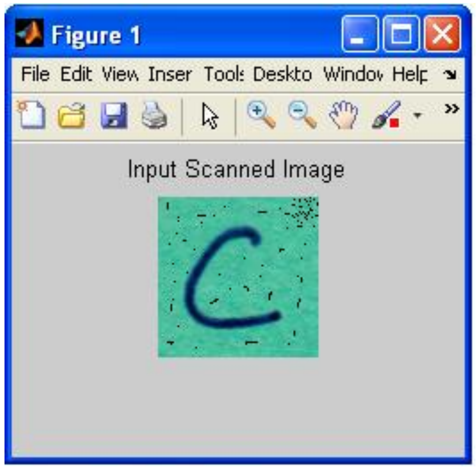

Fig.4.3 Input Scanned Handwritten Character Image

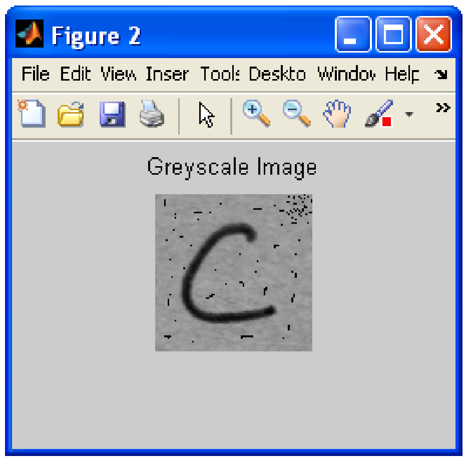

Fig.4.4 Handwritten Character Image in Grayscale Format

During this preprocessing stage, “rgb2gray” function of MATLAB is used to convert the input handwritten character image from the database as shown in Fig.4.3 is transformed into grayscale image as shown in Fig. 4.4. The issues rose due to the different colors of the background and the ink itself are eliminated here.

4.6.2 Binarization Technique

The grayscale images are then transformed to the binary matrix form in which a 0 represents a black pixel in the foreground and a 1 represents a white pixel in the background. Fig.4.5 displays such binary image.

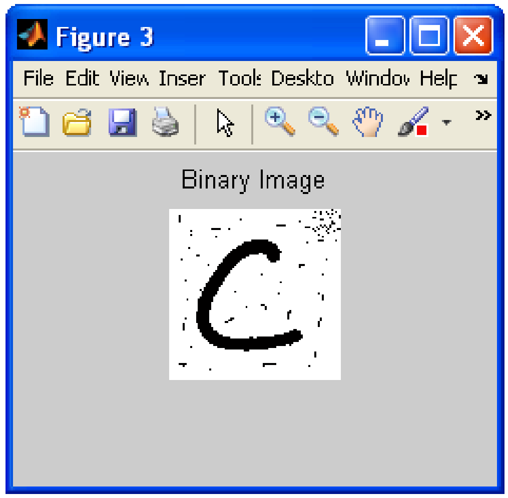

Fig.4.5 Handwritten Character Image in Binary Format

4.6.3 Noise Removal

In De-Noising preprocessing stage, the noise (small foreground components and dots) induced in the image scanning process is optimally eliminated by using MATLAB's "bwareaopen" method. Only the noise dots and other foreground components have been removed in this step while retaining the character components as shown in Fig.4.6. For highlighting the removed pixels, two MATLAB methods "bwlabel" and "regionprops" has been used for highlighting the noisy pixels as shown in Fig.4.7. Noise free images are shown in Fig.4.8.

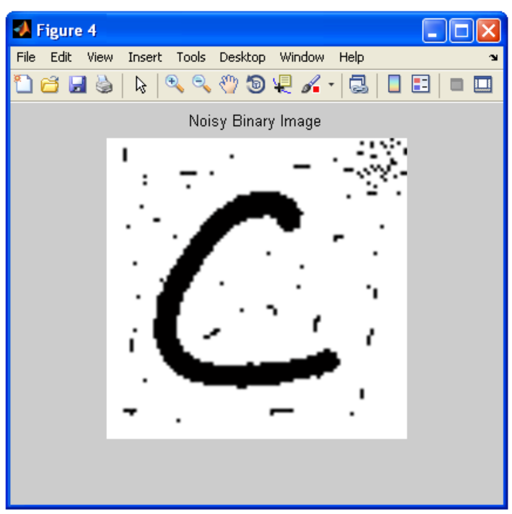

Fig.4.6 Noisy Handwritten Character Image

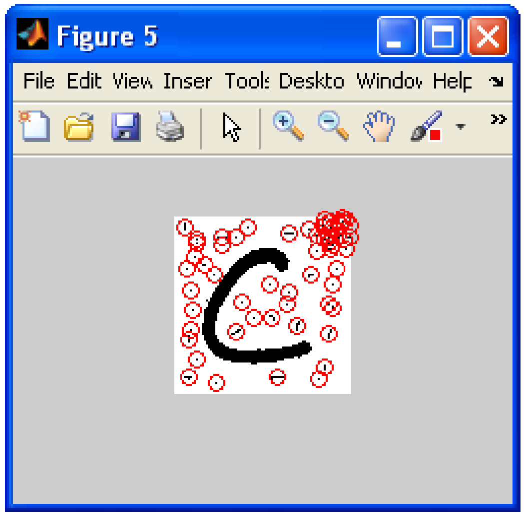

Fig.4.7 Noise Detection in Handwritten Character Image

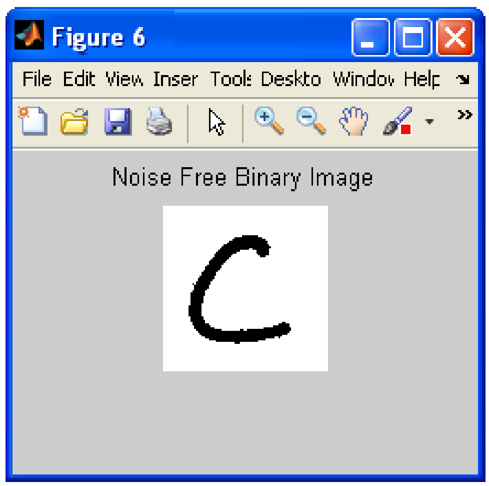

Fig.4.8 Handwritten Character Image after Noise Removal

4.6.4 Cropping

The images after de-noising are cropped to remove the extra space available around the rectangular region enclosing the handwritten noise free character image. MATLAB's "imcorp" method produces a minimum possible rectangular area which contains whole character image. Fig.4.9 shows a cropped character image.

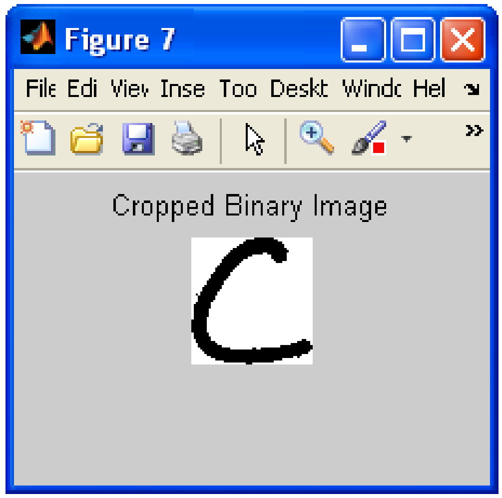

Fig.4.9 Cropped Handwritten Character Images

4.6.5 Size Normalization

It is mandatory for our experiment that all the input character images must be size normalized i.e. they must be same in size. MATLAB's "imresize" method is employed to reconstruct all the sample images with a size of 15×12. Fig. 4.10 shows resized character image.

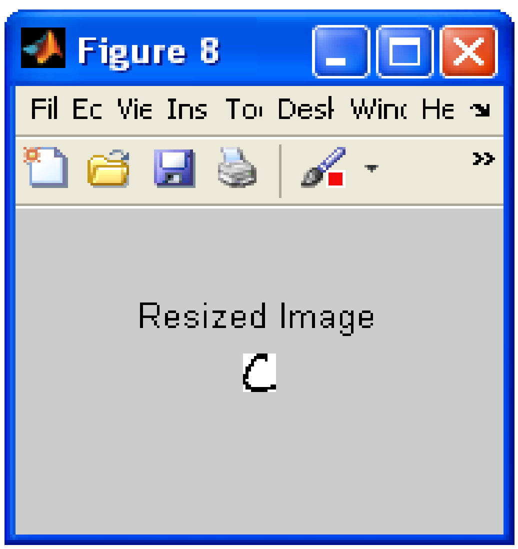

Fig.4.10 Resized Handwritten Character Images

4.7 Feature Extraction

In this process, beneficial information is extracted from the patterns and this information is used in the recognition process.

Fig.4.11(a) shows the original binary image of ‘c’ which is resized and shown in Fig. 4.11(b) in the form of 15×12pixel matrix. In Fig.4.11(c) matrix form of the image is displayed in which a ‘0’ represents a white pixel and a ‘1’ denoted a black pixel. In Fig. 4.11(d) a single column binary matrix of 180 × 1 is displayed which is obtained from reshaping the matrix of 15×12 using MATLAB’s “reshape” method. This single column matrix is termed as feature vector of the original image of character ‘c’ as displayed in Fig.4.11(a).

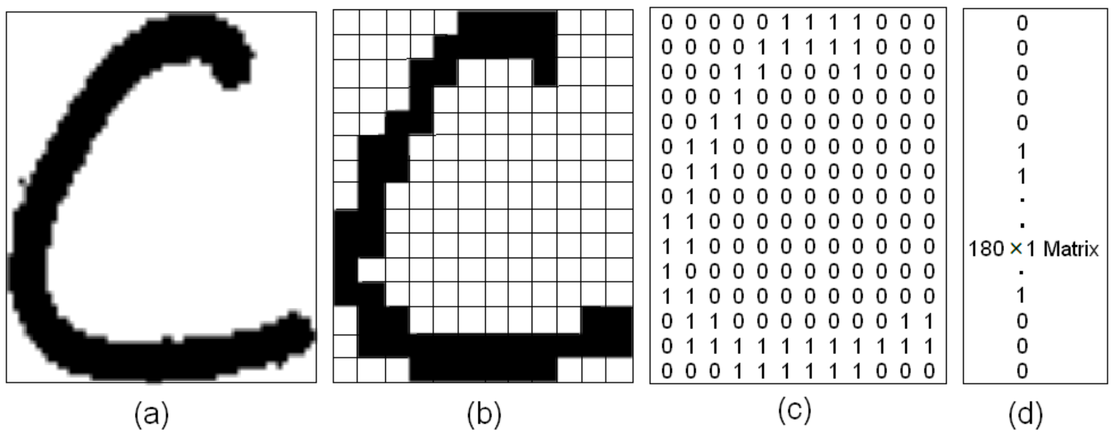

Fig.4.11(a) Binary Image of Character ‘c’ (b) Resized Binary Image of Character ‘c’; (c) Binary Matrix representation and (d) Reshaped Binary Matrix or Feature Vector of Character ‘c’.

4.8 Sample Preparation for Neural Network Training

A single column matrix sized 180 × 1 is termed as feature vector of the original image of character ‘c’ as displayed as the third column in the matrix shown in Fig.4.11(a). A sample of size 180 × 26 has been constructed by combining all the single column feature vectors of individual sizes 180 × 1 each representing all 26 characters available in the English alphabet. Such a complete sample is shown in Fig.4.12.

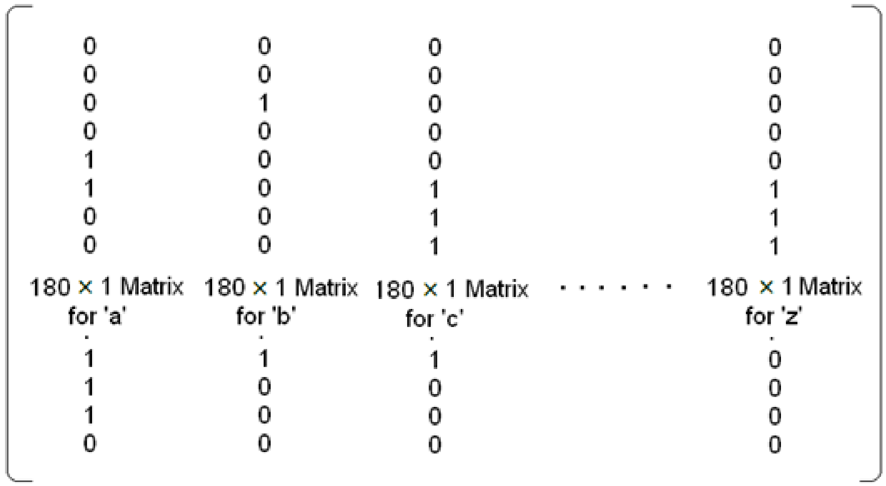

Fig.4.12 Matrix representation of Input Sample

Here, feature vector representing a character ‘a’ is placed in first column; ‘b’ is represented by second column and so on. During creation of samples, 1300 character images have been collected where an author contributed 5 samples. A sample consists of all the 26 characters (a-z) from English alphabet. Training is performed by these samples after preprocessing.

4.9 Classification and Recognition Process

In a typical OCR System, this step in the most important and is the final step. Accuracy of an OCR System is dependent on the techniques involved during preprocessing and the superiority of the feature set extracted from the sample input for recognition process. The classifier selected to do the classification also plays an important role in recognition process. It is concluded from the review of work done by various researchers that Artificial Neural Networks can be selected as the superior tool among other soft computing techniques for the task of classification.

4.9.1 Methodology and ANN Architecture

A Multilayer feed forward artificial neural network is selected for the classification task in which all the nodes in hidden and output layers have “tansig” activation function which is a differentiable and non-linear activation function. Linear function is used in the units of the input layer. The learning of ANN involves the backpropagation learning rule.

Backpropagation (of errors) is a logical technique for the training of a feed forward neural network which uses delta learning rule based on gradient descent. Weights associated with the Interconnecting nodes are adjusted during the training process of the network based on the difference between actual appearing outputs and the expected outputs. This weight updating process continues till some acceptable level of error is reached during network convergence.

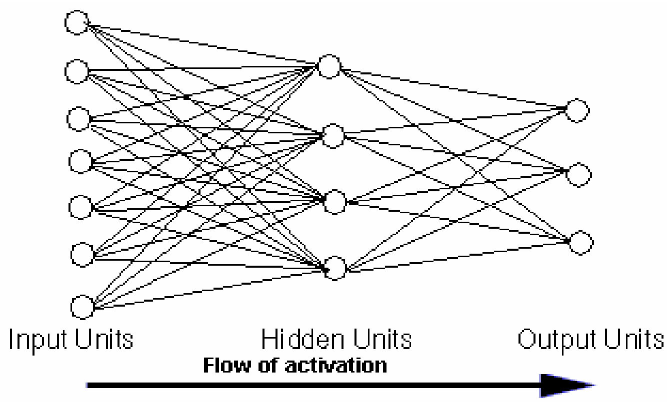

Fig.4.13 Feed Forward Neural Network with one Hidden Layer

During feed forward, signals propagates forward and are sent back in reverse direction during backpropagation. The processing units in hidden and output layers have "tansig" activation function which is a differentiable and non-linear activation function. Linear function is used in the units of the input layer as displayed in Fig.4.14.

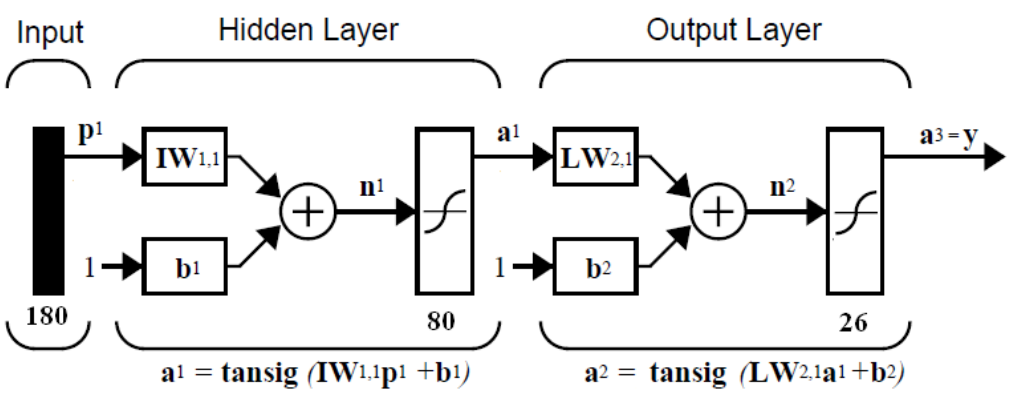

Fig.4.14 Architecture of the Neural Network used in the Recognition System

4.9.2 Experimental Conditions

Suitable parameter values for training a neural network are defined by trial and error method as there is no standard rule to define them. Table 4.1mentions various neural network training parameters set in the proposed character recognition experiment.

Table 4.1 Experimental Conditions during the Recognition Experiment

PARAMETERS	VALUE
Input Layer	
No. of Input neurons	180
Activation Function	Linear
Hidden Layer	
No. of neurons	80
Activation Function	'tansig'
Learning Rule	Momentum
Output Layer	
No. of Output neurons	26
Activation Function	'tansig'
Learning Rule	Momentum
Learning Constant	0.01
Acceptable Error Level (MSE)	0.001
Momentum Term (α)	0.90
Maximum Epochs	100000
Termination Conditions	Based on minimum Mean Square Error or maximum number of epochs allowed
Initial Weights and biased term values	Randomly generated values between 0 and 1
Number of Hidden Layers (N_{HL})	1

4.10 Implementation and Functional Details

Here, 180 is the number of neurons in the input layer because each character has a feature vector of length 180. For output layer, 26 is the number of neurons fixed as there are 26 characters available in English Alphabet.

To judge the optimal count of hidden units in the hidden layer is very tricky. Too few or too many of them will lead to under-fitting or over-fitting of the network respectively. By adopting trial and error technique, the count for hidden units has been selected as 80 for optimal utilization of system resources.

MATLAB's neural network toolbox has been used for the network training and its development environment can be seen from Fig.4.15.

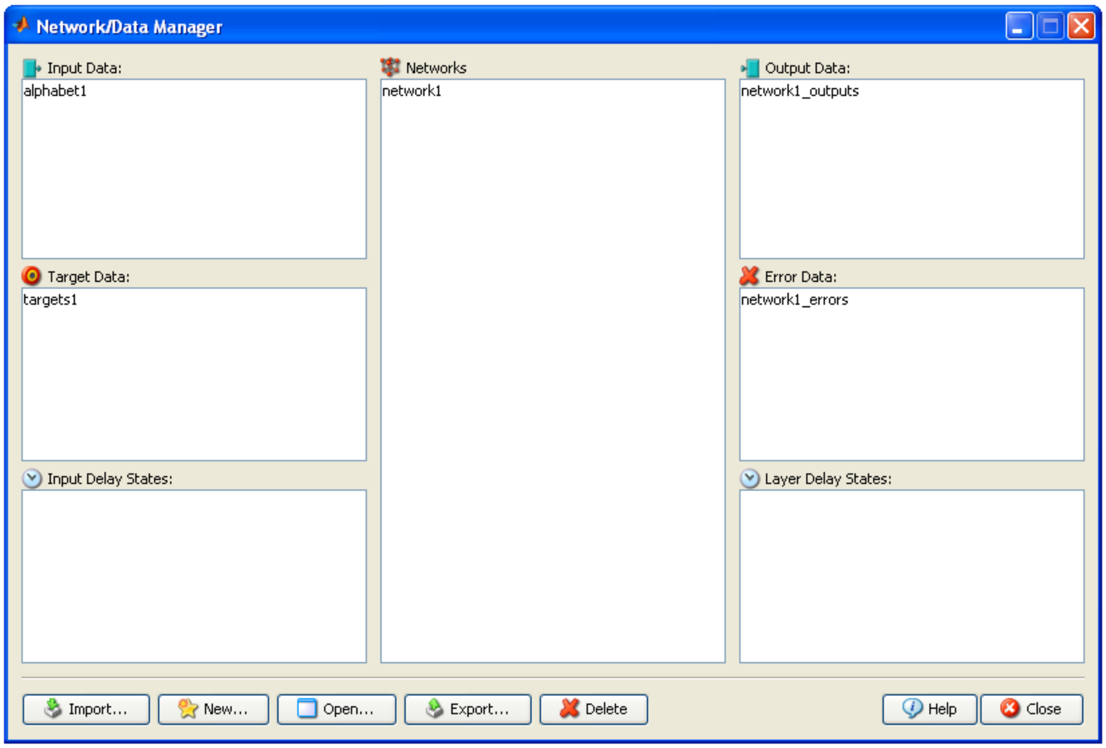

Fig.4.15 Development Environment for Neural Network Toolbox

As displayed in Fig.4.15, "alphabet1" has been applied as input for training and "target1" has been applied as expected outcome because backpropagation algorithm assumes supervised learning. Hidden and output layer units have "tansig" activation function as displayed in Fig.4.16.

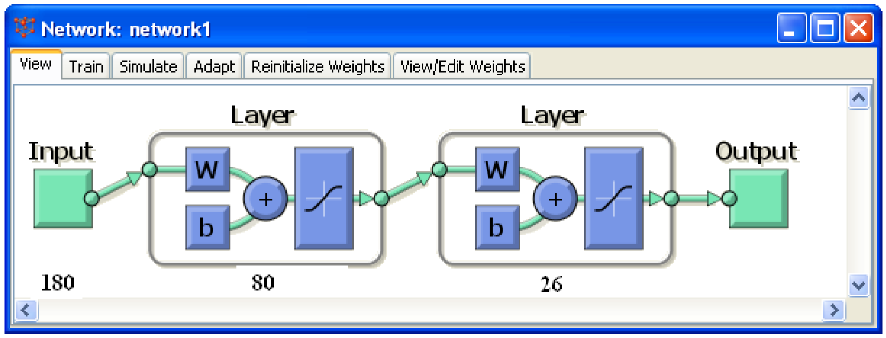

Fig.4.16 Structure of the Network

Fig.4.17 depicts the training process of the network using adaptive learning function "traingdx" where mean square function (MSE) being the "Cost Function". Network training will come to a halt whenever the MSE value reaches 0.001 which is set as acceptable level of error present during the training process. Low Performance value (0.000865) represents proper training of network has been completed and is shown in Fig.4.17.

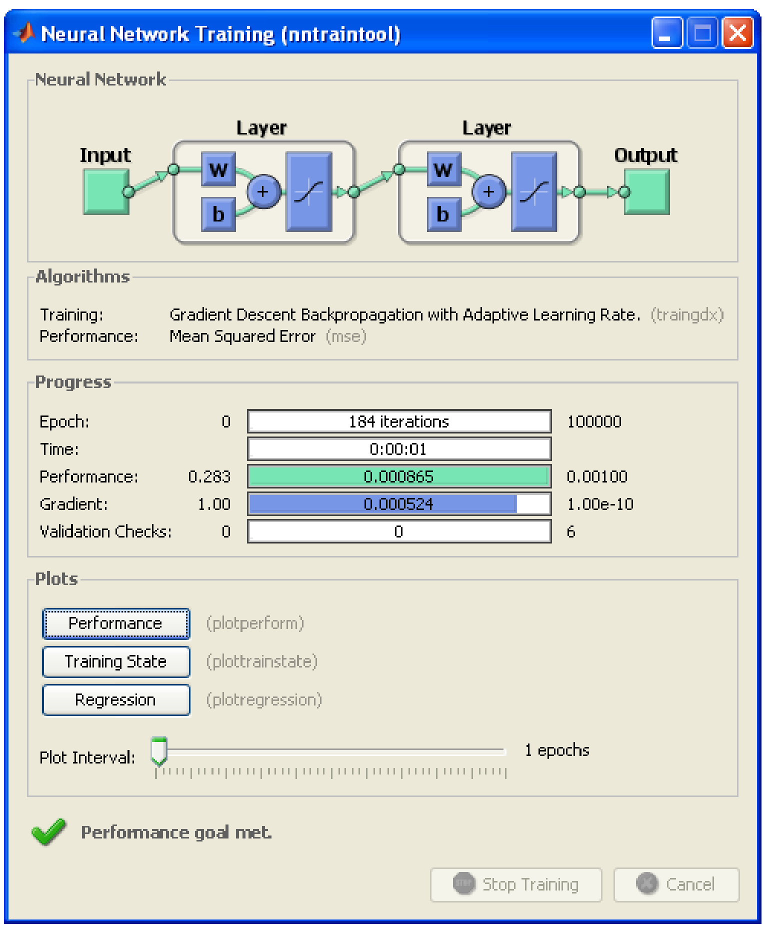

Fig.4.17 Training Process of the Network

Performance of the network also depends on the number of training epochs required by the network during its training. Too few training epochs will result in lack of proper training and too large will result in over training and unwanted wastage of time and resources.

As displayed in Fig.4.17, the maximum limit for training epoch has been fixed at 100000. Beyond this maximum allowed limit of the number of epochs, the network training will automatically stop.

A training sample consists of 26 handwritten characters (a-z) and the proposed neural network has been trained with 50 such samples i.e. 1300 (50×26=1300) samples have been participated in the training process. As shown in Fig.4.18, network has been trained with sample "alphabet1" and in 184 epochs, it has successfully converged to the training goal.

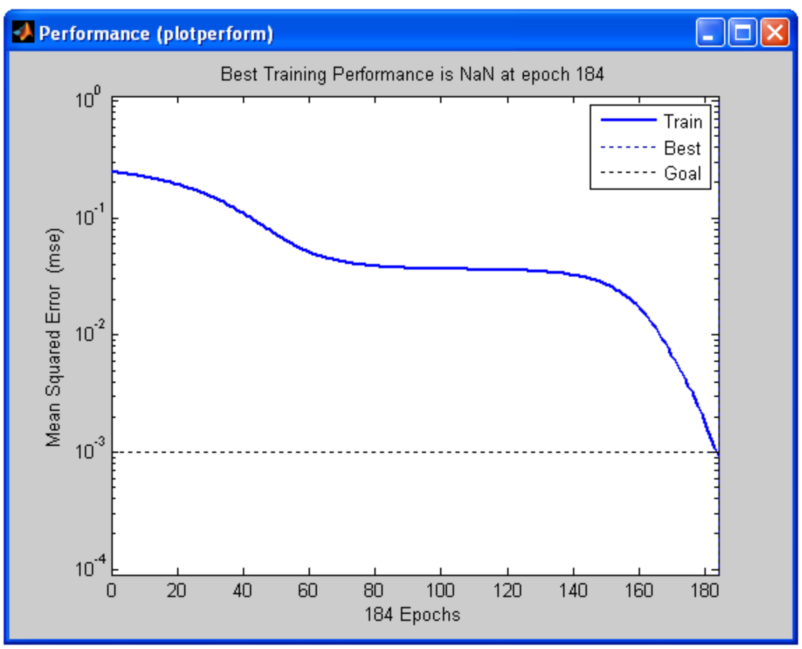

Fig.4.18 The Variation of MSE with the Training Epochs

4.11 Discussion of Results

The neural network has been exposed to 50 different samples. Each character at the input will put a '1' at that neuron in the output layer in which the maximum trust is shown and rest neuron's result into '0' status. The output is a binary matrix of size 26×26 because each character has 26×1 output vector. The first 26×1 column stores the first character's recognition output; the following column will be for next character and so on for 26 characters (a sample). For each character the 26×1 vector will contain value '1' at only one place. For example, character 'a' if correctly recognized, will result in [1, 0, 0, 0 …all …0], character 'b' will result in [0, 1, 0, 0 … all …0] and so on.

Alphabets	a	b	c	d	e	f	g	h	I	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	Success (%)
a	43	0	0	0	0	0	0	1	0	0	0	0	2	8	5	0	0	0	0	0	0	3	0	0	0	0	86
b	0	49	0	2	0	0	0	0	0	0	3	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	98
c	0	0	44	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	88
d	0	1	0	38	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	76
e	2	0	4	0	44	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	88
f	0	0	0	0	0	46	0	0	0	0	0	3	0	0	0	0	0	0	0	1	0	0	0	0	0	1	92
g	0	0	0	0	0	0	39	0	0	10	0	0	0	0	0	1	4	0	0	0	0	0	0	0	13	0	78
h	0	0	0	0	0	0	0	49	0	0	5	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	98
i	0	0	0	0	0	0	0	0	37	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	74
j	0	0	0	0	0	0	1	0	5	29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	58
k	0	0	0	0	0	0	0	0	0	0	42	2	0	0	0	0	0	0	0	1	0	0	0	0	1	0	84
l	0	0	0	1	0	1	0	0	3	0	0	39	0	0	0	0	0	0	0	0	0	0	0	0	0	0	78
m	0	0	0	0	0	0	0	0	0	0	0	0	41	2	0	0	0	1	1	0	0	0	0	0	0	0	82
n	0	0	0	0	0	0	0	0	0	0	0	0	0	38	0	0	0	0	1	0	0	0	0	0	0	0	76
o	5	0	1	0	1	0	0	0	0	0	0	0	0	0	43	0	0	0	1	0	0	0	0	0	0	0	86
p	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	44	0	0	0	10	0	0	0	0	0	0	88
q	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	46	0	0	0	0	0	0	0	1	0	92
r	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	49	0	0	0	0	0	0	0	0	98
s	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	46	0	0	0	0	0	0	0	92
t	0	0	0	4	0	3	0	0	5	0	0	6	0	0	0	0	0	0	0	38	0	0	0	0	0	2	76
u	0	0	1	0	1	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	48	5	0	0	0	0	96
v	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	40	0	0	0	0	80
w	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	50	0	0	0	100
x	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	1	0	0	0	0	50	0	1	100
y	0	0	0	0	0	0	8	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	35	0	70
z	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	46	92
Overall Character Recognition Accuracy =																											85.62 %

Output Class

Target Class

Fig.4.19 A Confusion Matrix Representing the Performance of the Neural Network Classifier

Fig.4.19 shows a confusion matrix which displays recognition results and indicates the confusion amongst the various classified character images during testing the correctness of the network.

As shown in Fig.4.19, character “a” is classified 43 times correctly out of 50. Two times it is miss classified as “e” and five times it is miss classified as “o”. Similarly, character “z” is recognized46 times in a correct way. One time it is miss classified as “f”, two times as “t” and one time as “x”. Total trials for each type of characters were fifty.

The last column in Fig.4.19 shows the recognition accuracy of each character (a-z) and the last row shows the average character recognition accuracy of all the characters and is found to be 85.62% which is satisfactory for this experiment. Fig.4.20 shows the 3-D plot of this confusion matrix drawn by MATLAB.

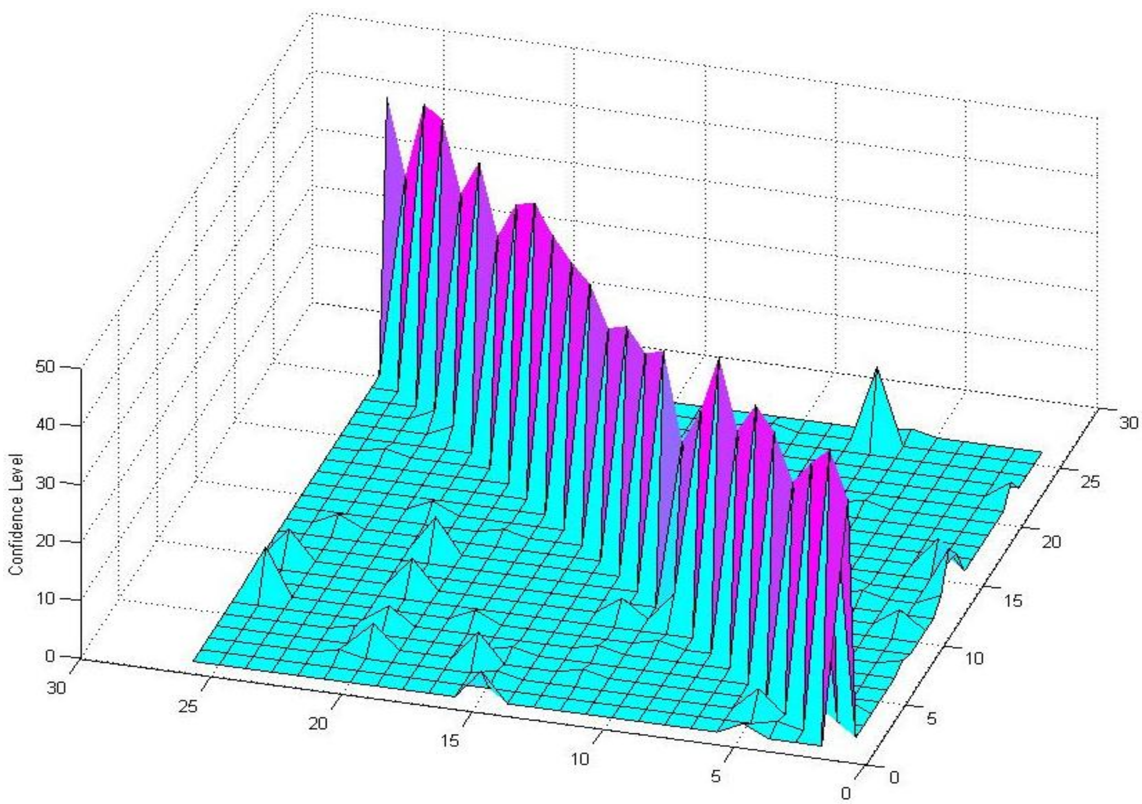

Fig.4.20 Three Dimensional Plot of Confusion Matrix Representing the Performance of the Neural Network Classifier

4.12 Conclusion

In the proposed experiment, the MLP has been designed with 80 units in hidden layer & “tansig” transformation function in the nodes of output as well as hidden layers. For applying character images as input to the network classifier, all character images were reshaped in 180×1 matrix after being resized to 15× 12 during training sample preparation.

The feature vector extracted using binarization technique for each character is of length 180; hence, the number of units in input layer has been fixed to 180. There exist 26 output classes in which each input character sample is to be classified so the number of nodes in output layer has been fixed to 26. Remarkable recognition accuracy has been achieved by using binarization features of “gradient descent approach” of backpropagation algorithm.

The recognition accuracy of 85.61% was achieved which has been considered excellent but can be improved further by improving the “preprocessing techniques”, “feature extraction techniques”, “training sample quality” and the “classifier itself” such as “a radial basis function (RBF) neural network” etc. can be tried as a classifier for the handwritten character recognition experiment.

CHAPTER-5

5 Implementation of RBF Neural Network Classifier System for Off-Line Cursive Handwritten Digit Recognition using Projection Profile Features

5.1 Introduction

Man Machine interface can be improved by simulating the reading process of humans which is termed as OCR automation. Out of the various steps involved in the process i.e. “preprocessing”, “feature extraction” and “classification”, the most important step towards this automation is “Feature Extraction”. Initially preprocessing is done so as to improve the quality of the extracted features. The main objective behind preprocessing is to remove the invariabilities existing in digit images. In this work, the main focus is on feature extraction and classification.

5.2 Proposed Recognition System

Here, in this work it is proposed to develop a handwritten or machine printed digits classification system. “Projection Profile Features” i.e. features in vertical direction, horizontal direction and towards laft & right daigonals are extracted for each input digit image. RBF (Radial Basis Function) network has been employed for the classification job and delivered outstanding recognition results.

Handwritten digit samples were collected from 10 contributors. Five sets of ten digits (0-9) have been collected from each author. A database a prepared in which has 500 ($10\times5\times10=500$) samples and were involved in the training of the chosen RBF network.

5.3 Image Acquisition and Sample Preparation

A digital photo camera or a scanner is generally used to capture the handwritten digit images and these images are saved in .bmp or .jpg file format for pre-processing.

500 digit samples have been contributed by 10 persons. A sample data set of 50 digit images contributed by a writer is shown in Fig.5.1.

Fig.5.1 Digit Samples Contributed by a Writer

5.4 Preprocessing

The main objective behind preprocessing is to remove the invariabilities existing in digit images. While scanning the handwritten digit images, the quality may be ruined as the noise is introduced due to dust or due to colored background. One sample image taken from our database has been displayed in Fig.5.2.

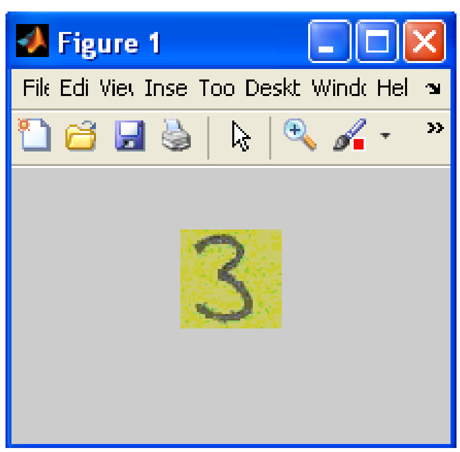

Fig.5.2 Input Scanned Digit Image

The noise that was introduced due to colored background has been removed by using "gray scale intensity threshold" during binarization. In this image, a "0" represents a black pixel in the foreground and a 1 represents a white pixel in the background. The resulting image is shown in Fig.5.3.

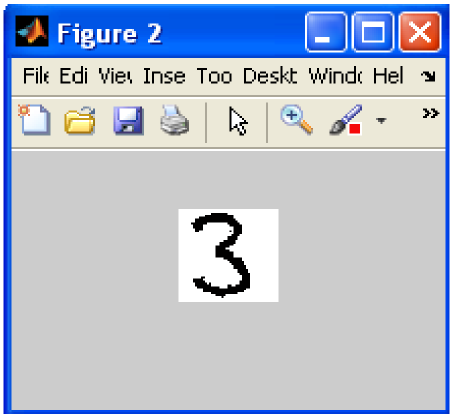

Fig.5.3 Image after Background Noise Removal

It is mandatory for our experiment that all the input digit images must be size normalized i.e. they must be same in size. MATLAB's "imresize" method is employed to reconstruct all the sample images with a size of 16×16. Fig. 5.4 shows resized digit image.

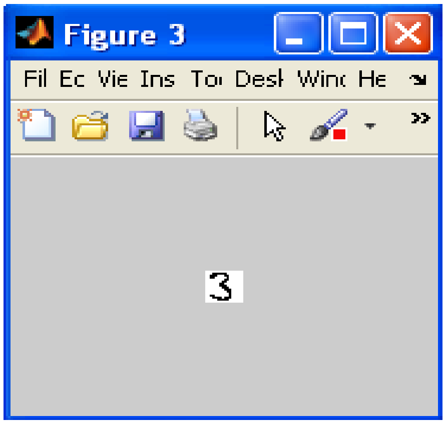

Fig.5.4 Size Normalized Digit Image

5.5 Feature Extraction

Extracting better features are directly correlated with the better recognition results. Projection Profile features proved their importance even if the input samples are lacking some preprocessing techniques such as smoothing or thinning. In fact, thinning has been kept away from this experiment as there are chances that some crucial information regarding number of black pixels and their respective positions may be lost.

The projection of the sample image is drawn in four directions by traversing the image in these directions and counting the number of black pixels individually in these four directions (Desai, 2010) as shown in Fig.5.5.

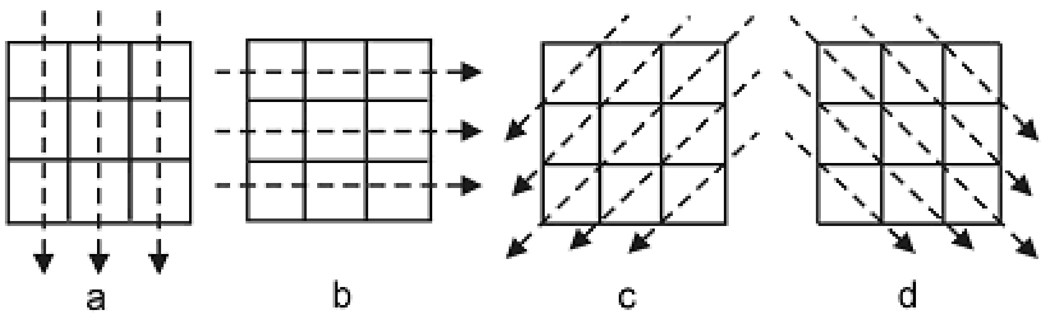

Fig.5.5 Projection Directions for a 3×3 Image Pattern

As the size of the sample image is 16×16 pixels, 16 columns and 16 rows are there. Fig.5.6(a) shows the column wise count of black pixels at the bottom of each column. Fig.5.6(b) shows the row wise count of black pixels at right side of each row. In the same manner, the count of black pixels for left and right diagonals has been shown in Fig.5.6(c) & Fig.5.6(d) respectively.

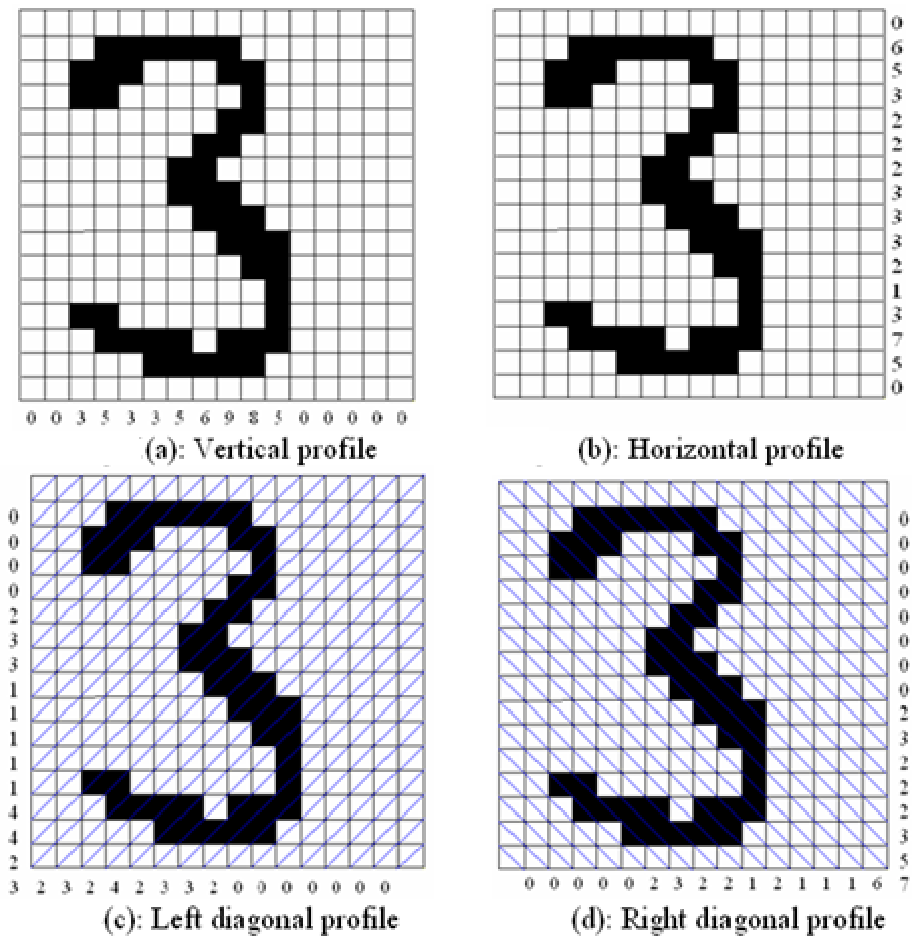

Fig.5.6 Projection Profile of Digit '3' in Vertical, Horizontal, Left-Diagonal and Right-Diagonal Orientations

A single feature vector can be obtained by combining projection profiles of all the four orientations. So, after concatenating all the four profiles, projection feature vector for digit "3" will be of length 94 (16+16+31+31=94) and can be written as:

{0 0 3 5 3 3 5 6 9 8 5 0 0 0 0 0

0 6 5 3 2 2 2 3 3 3 2 1 3 7 5 0

0 0 0 0 2 3 3 1 1 1 1 1 4 4 2 3 2 3 2 4 2 3 3 2 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 2 3 2 2 2 3 5 7 6 1 1 1 2 1 2 2 3 2 0 0 0 0 0 }

5.6 Classification and Recognition

The classifier selected to do the classification also plays a crucial role in recognition process.

5.6.1 RBF Neural Network as a Classifier

A RBF Network (RBFNN) is selected for the classification task in which nodes in hidden layer have bell shaped Guassian activation function is used which is a differentiable and non-linear activation function as shown in Fig.5.7. Linear function is used in the units of the output layer.

The main advantage of RBFNN is that gets trained very fast and a few input samples are required for its training.

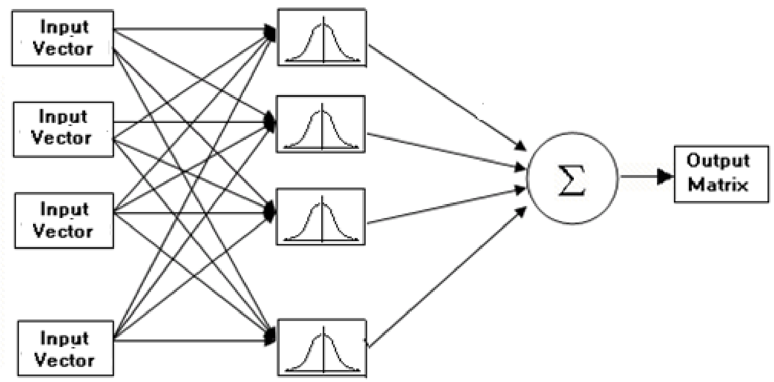

Fig.5.7 A General RBF Neural Network with One Output

The bell shaped radial basis function used is a Gaussian Function represented by:-

$$\phi(x_i) = e^{-\frac{\left\| x-\mu_i \right\|^2}{2\sigma_i^2}}$$

where

"φ" is a "Gaussian function",

"x" is called "input of neuron i",

"μ_i" is called "basis of neuron i"&

"σ_i"is the "amplitude of neuron i".

"Radial Basis Function network" is competent enough to deliver comparable or even better classification results as compared to "Multilayer feed-forward neural network".

5.6.2 Training Sample Preparation

A single digit image can be denoted by a projection feature vector having a length of 94. Fig. 5.8 shows the collection of projection feature vectors of 10 digits (0-9). First column represents the digit "0" and second column represents the digit "1" and so on. The collective feature vectors of all the 10 sample digits are shown as a 94×10 matrix shown in Fig. 5.8.

0	0	0	0		0
0	0	0	6		3
0	1	5	5		4
2	1	2	3		2
7	2	2	2		4
6	2	1	2		2
5	2	1	2		3
4	2	2	3		6
:	:	:	:		:
94 × 1 Matrix for digit 0	94 × 1 Matrix for digit 1	94 × 1 Matrix for digit 2	94 × 1 Matrix for digit 3	. . . . .	94 × 1 Matrix for digit 9
:	:	:	:		:
0	1	2	2		0
0	0	1	3		0
0	0	1	2		0
0	0	0	0		0
0	0	0	0		0
0	0	0	0		0
0	0	0	0		0
0	0	0	0		0

Fig.5.8 Matrix Representation of Input Digit Sample of Size 94×10

5.6.3 Implementation and Functional Details

Radial Basis Function has been used in the hidden nodes of the network employed in this work. Here, 94 is the number of neurons in the input layer because each character has a feature vector of length 94. For output layer, 10 is the number of neurons fixed as there are 10 digits.

The development environment of MATLAB's NN toolbox used for training the RBF network is shown in Fig.5.9.

Learning in RBF Network is supervised in nature. In network named “network1”, training input sample “P” is applied as input data and “T” is the target data. Actual output is “network1_outputs” and actual error is represented by “network1_errors”.

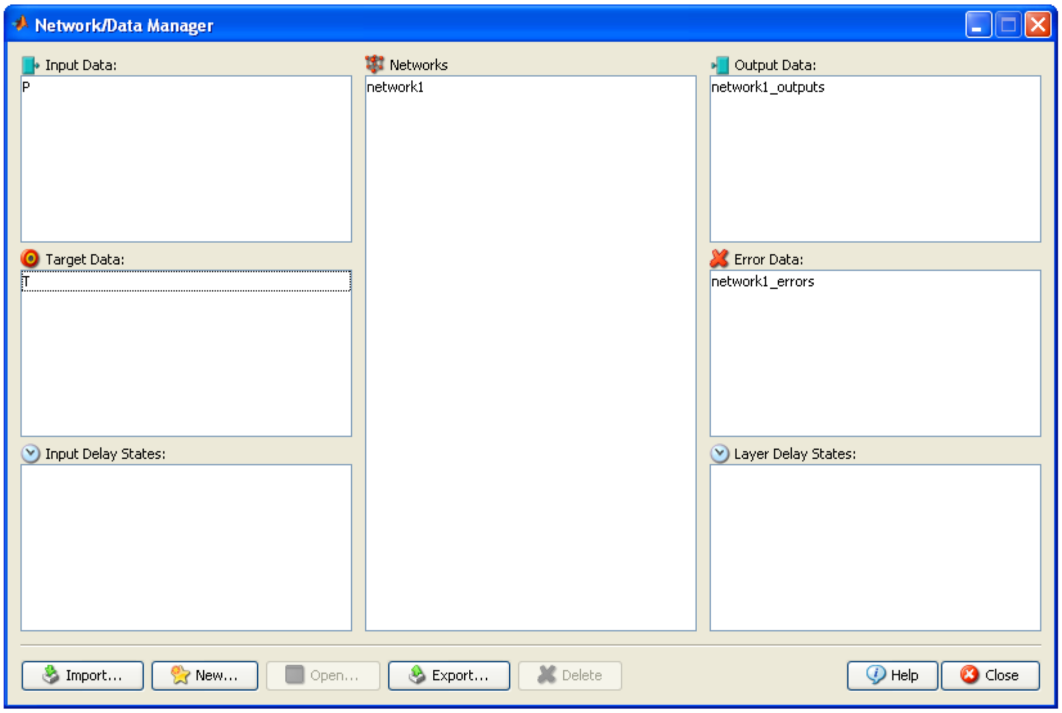

Fig.5.9 The Development Environment for Neural Network Toolbox

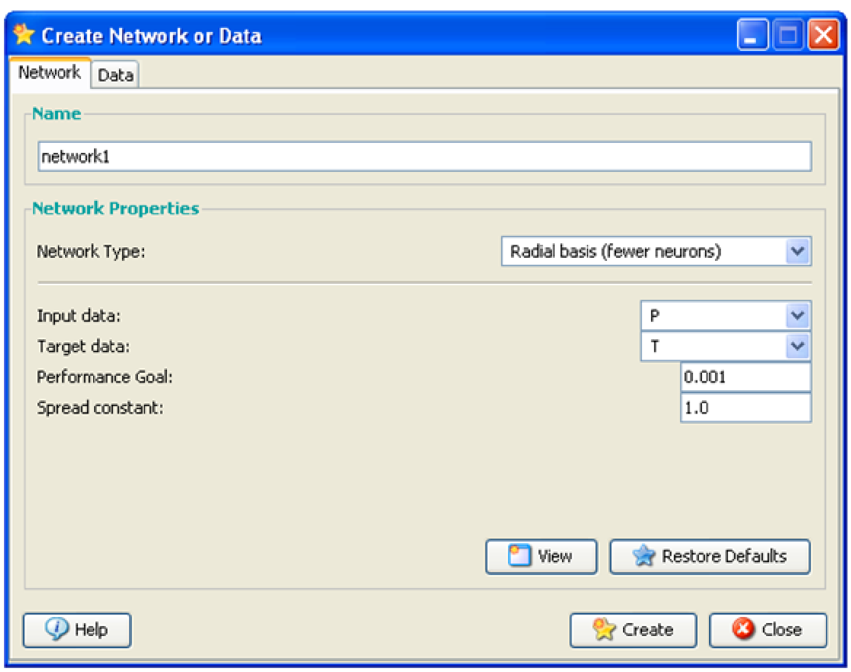

Fig.5.10 RBF Network Creation Wizard

RBF network's training parameters has been initialized as shown in Fig.5.10. This wizard is used to define the performance goal. Whenever the error value drops below the set performance goal of 0.001, the training process automatically stops.

Fig.5.11 represents the RBF network's structure. Output layer nodes have "linear activation function" and input layer units have "Gaussian activation function (F)".

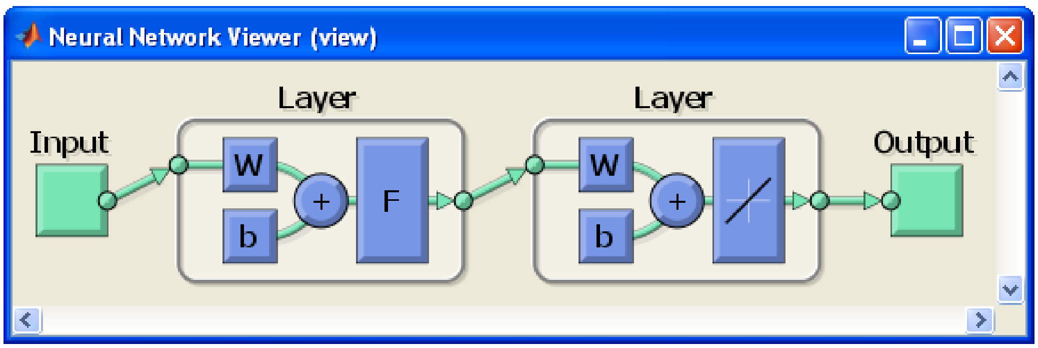

Fig.5.11 Structure of the RBF Network

5.7 Discussion of Results

MSE plot during the training is shown in Fig.5.12. Network training will come to a halt whenever the MSE value reaches 0.001 which is set as acceptable level of error present during the training process.

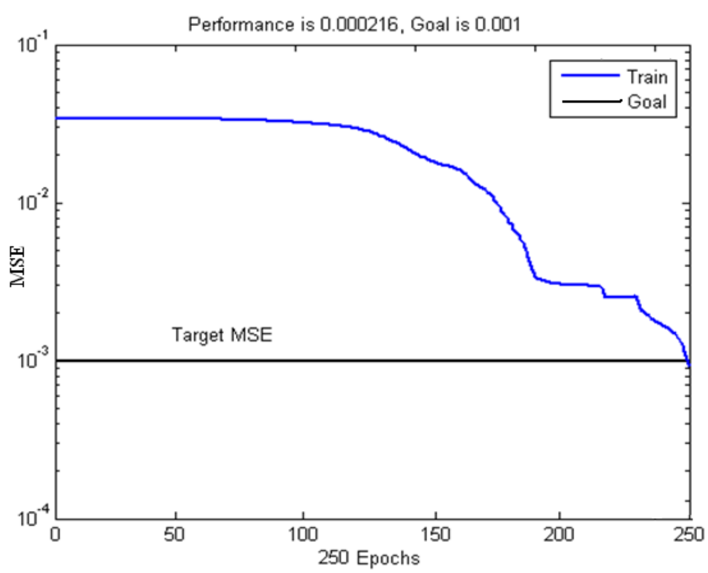

Fig.5.12 RBF Network Performance with Training Epochs

Table 5.1 shows confusion matrix which displays recognition results and indicates the confusion amongst the classified digits during testing the network's accuracy.

Table 5.1Confusion Matrix representing the Performance of the Classifier

Digit	0	1	2	3	4	5	6	7	8	9	Success (%)
0	50	0	0	0	0	0	0	0	0	0	100
1	0	50	0	0	0	0	0	0	0	0	100
2	0	0	50	0	0	0	0	0	0	0	100
3	0	0	0	50	0	0	0	0	0	0	100
4	0	0	0	0	50	0	0	0	0	0	100
5	0	0	0	0	0	50	0	0	0	0	100
6	0	0	0	0	0	0	50	0	0	0	100
7	0	0	0	0	0	0	0	50	0	0	100
8	0	0	0	0	0	0	0	0	50	0	100
9	0	0	0	0	0	0	0	0	0	50	100
						Overall Recognition Accuracy					**100 %**

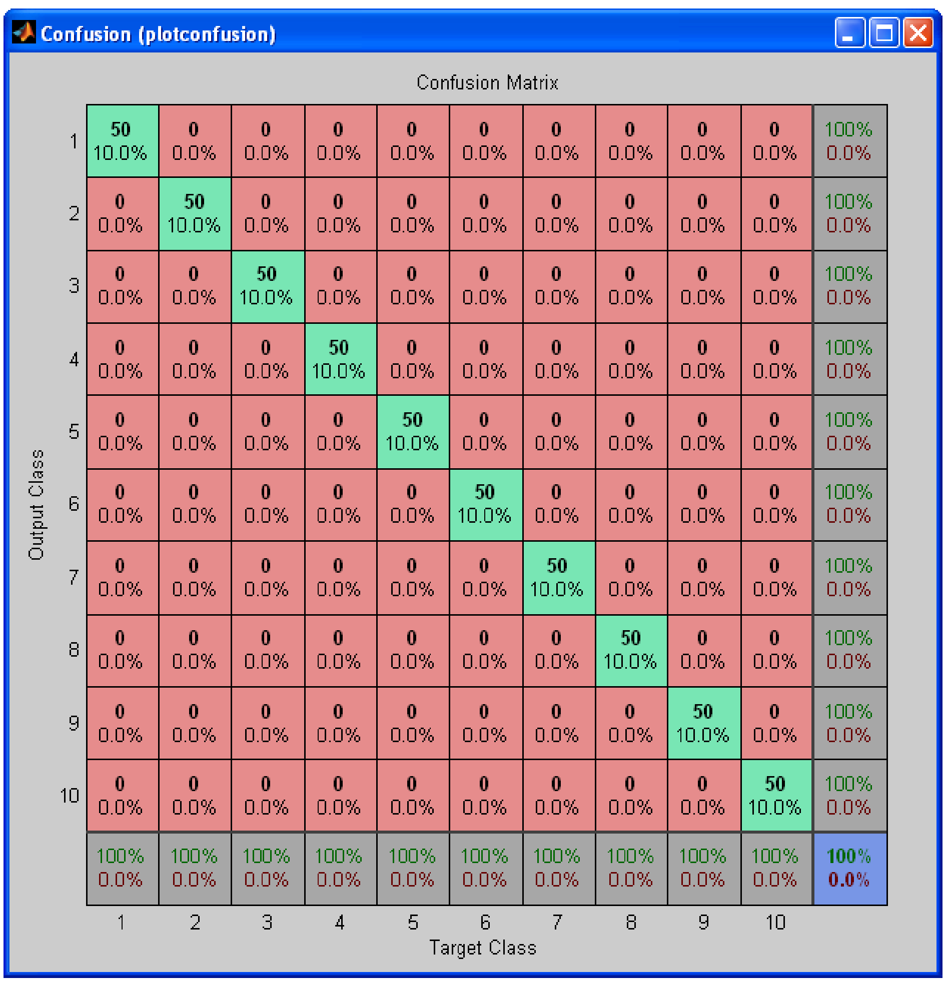

Fig.5.13 Two-Dimensional Confusion Matrix Plot in MATLAB Environment

As shown in Table 5.1, digit “0” is classified 50 times correctly out of 50. Total trials for each type of characters were fifty. All the digits presented to the network are classified correctly every time. None of the digits are miss-classified even a single time. The last column in Table 5.1 shows the recognition accuracy of each digit (0-9) and the last row shows the average digit recognition accuracy of all the digits and is found to be 100 % which is outstanding for this experiment.

Fig.5.13 shows the confusion matrix plot drawn by MATLAB. As indicated by this confusion matrix, all the handwritten digits are recognized correctly and the RBF network delivered 100% recognition accuracy. Here, the output and target class ‘1’ means the digit ‘0’ and the output and target class ‘10’ means the digit ‘9’.

Fig.5.14 shows the 3-D plot of this confusion matrix drawn by MATLAB.

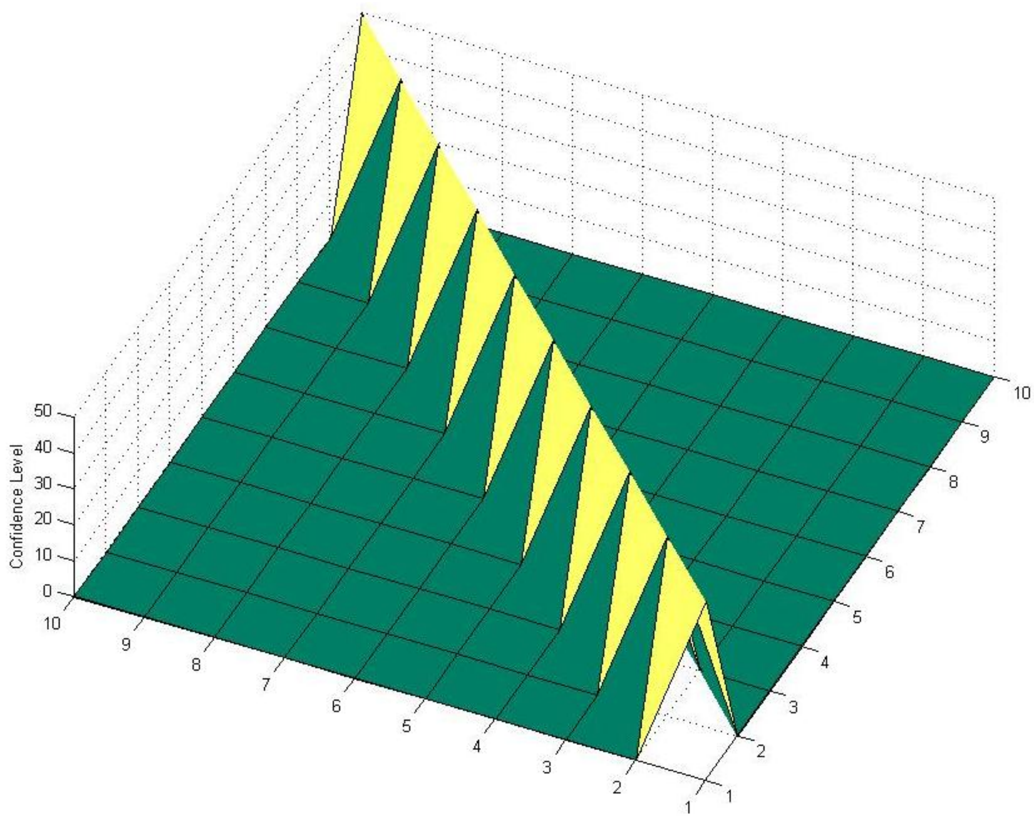

Fig.5.14Three-Dimensional Confusion Matrix Plot in MATLAB Environment

5.8 Conclusion

By using the feature vectors based on projection profiles in vertical, horizontal, left & right diagonal directions and the RBF network delivered 100% recognition accuracy which is remarkable for this experiment. This accuracy of 100% is obtained for a local data base having 500 digit samples. In future, apart from selecting the best training parameters for the network; preprocessing and feature extraction techniques can be improved further to achieve same level of accuracy for a huge dataset of isolated sample digit images also.

CHAPTER-6

6 Conclusion and Future Scope

6.1 Conclusion

The focus of this report was to explore preprocessing, untouched and touched character segmentation and feature extraction techniques for “off-line cursive handwritten words recognition”. MLP & RBF classifiers have been employed for recognition task and various techniques are suggested to optimize the neural network classifier for further improvement in the recognition accuracy.

The first character segmentation technique proposed in this report is based upon “connected component analysis” and is employed to segment untouched characters. Some shortcomings were identified in situations where a captured image of handwritten character is broken i.e. when it is not totally connected.

The second character segmentation technique proposed in this report is basically a vertical dissection based segmentation technique used to segment touched characters within a word. It guarantees to segment correctly whenever the characters with in a word are not touched. Ligatures connecting “Closed Characters” were also segmented in an excellent manner. In case of “Open Characters”, over-segmentation issue was minimized.

The use of binarization features and back-propagation network delivered admirable recognition correctness of 85.62%. By using projection features and RBF network delivered outstanding recognition accuracy of 100% for recognition of handwritten digits.

6.2 Future Scope

Performance of pattern classifiers can be improved further if the best features are selected from all the extracted features by using various other Soft Computing techniques apart from “Artificial Neural Networks” such as “Genetic Algorithms”, “Fuzzy Systems” or their combination. Genetic algorithms are based on “the survival of the fittest” and are the most efficient feature selection techniques to select best features from the extracted feature set of the character/digit image. It can also be investigated for optimal weight initialization during Neural Network Training. Fuzzy Set theory can be involved for the recognition of noisy, broken or incomplete character patterns having imprecise information.

Apart from MLP & RBF networks, some another potential neural network classifiers such as “Hopfield Network”, “Hidden Markov Model (HMM)” & “Support Vector Machine (SVM)” can be examined in this domain of research. Different classifiers have their own advantages and disadvantages. Therefore, the advantages of each individual method can be inherited in a single classifier system built by the fusion of various other classifiers working at different stages in this system.

Hence, Genetic Algorithms and Fuzzy Set Theory for extraction and selection of features along with the fusion of classifiers may become an interesting research direction for future.

REFERENCES

Alginahi, Y. (2010). "Preprocessing Techniques in Character Recognition", Character Recognition, Minoru Mori (Ed.), ISBN: 978-953-307-105-3, InTechopen Publishers, pp. 1-20.

Arica, N., Yarman-Vural, F. T. (2002). "Optical Character Recognition for Cursive Handwriting", IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(6), pp. 801-813.

Blumenstein, M. & Verma, B. (2001). "Analysis of Segmentation Performance on the CEDAR Benchmark Database", in proceedings of the 6th International Conference on Document Analysis and Recognition, Seattle, USA: IEEE Computer Society Press, pp. 1142-1146.

Blumenstein, M., Liu, X. Y. & Verma, B. (2004). "A Modified Direction Feature for Cursive Character Recognition", in proceedings of the International Joint Conference on Neural Networks, Budapest, Hungary: IEEE Computer Society Press, pp. 2983-2987.

Blumenstein, M., Liu, X. Y. & Verma, B. (2007). "An investigation of the modified direction feature for cursive character recognition", Pattern Recognition, 40, pp. 376–388.

Blumenstein, M., Verma, B. (1997). "A segmentation algorithm used in conjunction with artificial neural networks for the recognition of real-world postal addresses", in proceedings of 2nd online world conference on soft computing in engineering design and manufacturing, pp. 184-191.

Bortolozzi, F., de Souza, Britto. A., Jr, Oliveira, L.S. & Morita, M. (2005). "Recent advances in handwriting recognition", in Pal U, Parui SK, Chaudhuri BB (eds) Document analysis, pp. 1–30.

Britto, A. Jr., Sabourin, R., Bortolozzi, F. & Suen, C. Y. (2004). "Foreground and background information in an HMM based method for recognition of isolated characters and numeral strings", in proceedings of the 9th international workshop on frontiers in handwriting recognition, pp. 371–376.

References

Burges, C. J. C., Be, J. I. & Nohl, C. R. (1992). "Recognition of Handwritten Cursive Postal Words using Neural Networks", Proceedings of the 5th USPS Advanced Technology Conference, pp. 117-124.

Camastra, F. & Vinciarelli, A. (2003). "Combining neural gas and learning vector quantization for cursive character recognition", Neurocomputing, 51, pp. 147-159.

Casey, R. G., & Lecolinet, E. (1996). "A survey of Methods and Strategies in Character Segmentation", IEEE Trans Pattern Anal Mach Intell, 18 (7), pp. 690–706.

Cavalin, P. R., Britto, A. S., Bortolozzi, F., Sabourin, R. & Oliveira, L. S. (2006). "An implicit segmentation based method for recognition of handwritten strings of characters", in proceedings of ACM symposium on applied computing, pp. 836–840.

Chen, M. Y., Kundu, A., Zhou, J. & Srihari, S. N. (1992). "Off-Line Handwritten Word Recognition using Hidden Markov Model", in proceedings of the 5th USPS Advanced Technology Conference, pp. 563-579.

Cheng, C. K., Blumenstein, M. (2005a). "The neural based segmentation of cursive words using enhanced heuristics", in proceedings of the 8th international conference on document analysis and recognition, 2, pp. 650–654.

Chiang, J-H. (1998). "A Hybrid Neural Model in Handwritten Word Recognition", Neural Networks, 11(2), pp. 337-346.

Cho, S. B. (1997). "Neural-Network Classifiers for Recognizing Totally Unconstrained Handwritten Numerals", IEEE Transactions on Neural Networks, 8(1), pp. 43-53.

Davies, E. (2005). "Machine Vision – Theory Algorithms Practicalities", Third Edition, Morgan Kaufmann Publishers, ISBN 13: 978-0-12-206093-9, ISBN-10: 0-12-206093-8, San Francisco, CA, USA.

Desai, A. A. (2010). "Gujarati handwritten numeral optical character recognition through neural network", Pattern Recognition, Vol. 43, pp. 2582-2589.

Dimauro, G., Impedovo, S., Pirlo, G. & Salzo, A. (1997). "Removing Underlines from Handwritten Text: An Experimental Investigation", Progress in Handwriting Recognition, World Scientific Publishing, pp. 497-501.

References

Dimauro, G., Impedovo, S., Pirlo, G. & Salzo, A. (1998). "An Advanced Segmentation Technique for Cursive Word Recognition", Advances in Handwriting Recognition, World Scientific Publishing, pp. 255-264.

Dunn, C. E., Wang, P. S. P. (1992). "Character segmenting techniques for handwritten text—a survey", in proceedings of 11th international conference on pattern recognition, 2, pp. 577–591.

Eastwood, B., Jennings, A. & Harvey, A. (1997). "A Feature Based Neural Network Segmenter for Handwritten Words, In proceedings of the 1st International Conference on Computational Intelligence and Multimedia Applications, Gold Coast, Australia, pp. 286-290.

Elliman, D. G., Lancaster, I. T. (1990). "A Review of Segmentation and Contextual Analysis Techniques for Text Recognition", Pattern Recognition, 23(3-4), pp. 337-346.

Fujisawa, H., Nakano, Y. & Kurino, K. (1992). "Segmentation methods for character recognition: from segmentation to document structure analysis", Proc IEEE, 80(7), pp. 1079–1092.

Fujisawa, H., (2008). "Forty years of research in character and document recognition-an industrial perspective", Pattern Recognition, 41, pp. 2435 – 2446.

Gader, P. D., Mohamed, M. & Chiang, J. H. (1997). "Handwritten Word Recognition with Character and Inter-Character Neural Networks", IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics, 27(1), pp. 158-164.

Gang, L., Verma, B. & Kulkarni, S. (2002). "Experimental Analysis of Neural Network Based Feature Extractors for Cursive Handwriting Recognition", Proceedings of the IEEE World Congress on Computational Intelligence, Hawaii, USA: IEEE Computer Society Press, pp. 2837-2841.

Gatos, B., Pratikakis, I. & Perantonis, S. J. (2006a). "Hybrid off-line cursive handwriting word recognition", in proceedings of 18th international conference on pattern recognition (ICPR'06), vol 2, pp. 998–1002.

Gatos, B., Pratikakis, I., Kesidis, A. L. & Perantonis, S. J. (2006b). "Efficient off-line cursive handwriting word recognition", Proceedings of the 10th international workshop on frontiers in handwriting recognition.

References

Gilloux, M. (1993). "Research into the New Generation of Character and Mailing Address Recognition Systems at the French Post Office Research Centre", Pattern Recognition Letters, 14(4), pp. 267-276.

Gilloux, M., Lemarie, B. & Leroux, M. (1995b). "A hybrid radial basis function/hidden Morkov model handwritten word recognition system", in international conference on document analysis and recognition, Montreal, pp. 394–397.

Guillevic, D., Suen, C. Y. (1998). "HMM-KNN word recognition engine for bank check processing", Proceedings of international conference on pattern recognition, Brisbane, pp. 1526–1529.

Günter, S., Bunke, H. (2004). "Feature selection algorithms for the generation of multiple classier systems and their application to handwritten word recognition", Pattern Recognit Lett, 25(11), pp. 1323–1336.

Günter, S., Bunke, H. (2005). "Off-line cursive handwriting recognition using multiple classifier systems". On the influence of vocabulary, ensemble, and training set size, Optics Lasers Eng, 43(3–5), pp. 437–454.

Hanmandlu, M., Murali, K. R. M., Chakraborty, S., Goyal, S. & Choudhury, D. R. (2003). "Unconstrained handwritten character recognition based on fuzzy logic", Pattern Recognition, 36(3), pp. 603-623.

Hull, J.J.,(1994). "A database for handwritten text recognition", IEEE Trans. Pattern Anal. Mach. Intelligence, 16, pp. 550–554.

Kang, K. W., Kim, J. H. (2004), "Utilization of hierarchical, stochastic relationship modeling for Hangul character recognition", IEEE Trans Pattern Anal Mach Intell., 26(9), pp. 1185–1196.

Kim, G., Govindaraju, V. & Srihari, S. N. (1999). "Architecture for Handwritten Text Recognition Systems", Advances in Handwriting Recognition, World Scientific Publishing, pp. 163-182.

Kim, J. H., Kim, K. K., Suen, C. Y. (2000). "An HMM-MLP hybrid model for cursive script recognition", Pattern Anal Appl, 3, pp. 314–324.

References

Knerr, S., Anisimov, V., Baret, O., Gorski, N., Price, D. & Simon, J. C. (1997). “The AIA inter-check system-Courtesy amount and legal amount recognition for French Checks”, Automatic bank cheque processing, pp. 43–86.

Koch, Paquet, T., Heutte, L. (2004). “Combination of contextual information for handwritten word recognition”, in 9th International workshop on frontiers in handwriting recognition, Kokubunji, pp. 468–473.

Koerich, A. L., Britto, A., Oliveira, L. E. S. & Sabourin, R. (2006). “Fusing high- and low-level features for handwritten word recognition”, in proceedings of the 10th international workshop on frontiers in handwriting recognition, pp. 85-97.

Koerich, A. L., Sabourin, R. & Suen, C. Y. (2003). “Large vocabulary off-line handwriting recognition: a survey”, Pattern Anal Appl, 6(2), pp. 97–121.

Koerich, A. L., Sabourin, R. & Suen, C.Y, (2005). “Recognition and verification of unconstrained handwritten words”, IEEE Trans Pattern Anal Mach Intell, 27(10), pp.1509–1522.

Kundu, Y. H., Chen, M. (2002). “Alternatives to variable duration HMM in handwriting recognition”, IEEE Trans Pattern Anal Mach Intell, 20(11), pp. 1275–1280.

Liu, C. L., Fujisawa, H. (2005). “Classification and Learning for Character Recognition: Comparison of Methods and Remaining Problems”, Proceedings of the International Workshop on Neural Networks and Learning in Document Analysis and Recognition, Seoul, Korea: IEEE Computer Society Press, pp. 5-7.

Fischer, S. (2000). “Digital Image Processing: Skewing and Thresholding”, Master of Science thesis, University of New South Wales, Sydney, Australia.

Fujisawa, H. (2005). “Handwritten digit recognition using state-of-the-art techniques”, in proceedings of 8th international workshop on frontiers of handwriting recognition, pp. 320–325.

Lu, Y., Shridhar, M. (1996). “Character segmentation in handwritten words—an overview”, Pattern Recognition, 29(1), pp. 77–96.

Madhvanath, S., Kleinberg, E. & Govindaraju, V. (1999). “Holistic Verification of Handwritten Phrases”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 21(12), pp. 1344-1356.

References

Marinai, S., Gori, M. & Soda, G. (2005). “Artificial Neural Networks for Document Analysis and Recognition”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(1), pp. 23-35.

Marti, U., Bunke, H., (2002). “The IAM database: An English sentence database for off-line handwriting recognition”, Int. J. Doc. Anal. Recognition, 15, pp. 65–90.

Martin, G. L., Rashid, M. & Pittman, J. A. (1993). “Integrated Segmentation and Recognition through Exhaustive Scans or Learned Saccadic Jumps”, International Journal on Pattern Recognition and Artificial Intelligence, 7(4), pp. 831-847.

Mitrpanont, J. L., Limkonglap, U. (2007). “Using contour analysis to improve feature extraction in Thai handwritten character recognition systems”, in 7th IEEE international conference on computer and information technology, CIT 2007, pp. 668–673.

Oliveira, L. S., Sabourin, R., Bortolozzi, F. & Suen, C. Y. (2002). “Automatic recognition of handwritten numerical strings: a recognition and verification strategy”, IEEE Trans Pattern Anal Mach Intell, 24(11), pp. 1438–1454.

Otsu, N. (1979). “A threshold selection method from gray level histogram”, IEEE Transaction on system, Man, Cybernetics, Vol. 19, pp. 62–66.

Plamondon, R., Srihari, S. N. (2000). “On-Line and Off-Line Handwriting Recognition: A Comprehensive Survey”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(1), pp. 63-84.

Rehman, A., Dzulkifli, M. (2008). “A simple segmentation approach for unconstrained cursive handwritten words in conjunction with the neural network”, Int J Image Process, 2(3), pp. 29–35.

Russ, J. (2007). “The Image Processing Handbook”, Fifth Edition, CRC Press, Boca Raton, FL, USA.

Saba, T. (2012). “Off-line cursive touched script non-linear segmentation”, Ph.D Dissertation.

Saba, T., Sulong, G. & Rehman, A. (2011). “Document image analysis: issues, comparison of methods and remaining problems”, Artificial Intelligence Review, 35, pp. 101-118.

References

Sadri, J., Cheriet, M. (2009). "A new approach for skew correction of documents based on particle swarm optimization", in proceedings of 10th international conference on document analysis and recognition, ICDAR '09, pp. 1066–1070.

Saeed, K., Albakoor, M., (2009). "Region growing based segmentation algorithm for typewritten and handwritten text recognition", Applied Soft Computing, 9, pp. 608–617.

Samrajya, P., Lakshmi, M. & Swaroop. A. (2006). "Segmentation of cursive handwritten words using Hypergraph", TENCON, IEEE region 10 Conference, pp. 1-4.

Sarfraz, M., Rasheed, Z. (2008). "Skew estimation and correction of text using bounding box", in proceedings of 5th international conference on computer graphics, imaging and visualization, (CGIV '08), pp. 259–264.

Srihari, S. N. (1993). "Recognition of Handwritten and Machine-printed Text for Postal Address Interpretation", Pattern Recognition Letters, 14(4), pp. 291-302.

Srihari, S. N. (2006). "Automatic Handwriting Recognition", Encyclopedia of Language & Linguistics, 2nd Edition, Elsevier.

Suen, C. Y., Legault, R., Nadal, C., Cheriet, M. & Lam, L. (1993). "Building a New Generation of Handwriting Recognition Systems", Pattern Recognition Letters, 14(4), pp. 305-315.

Suen, C. Y., Tan, J. (2005). "Analysis of errors of handwritten digits made by a multitude of classifiers", Pattern Recognition Letters, 26(3), pp. 369-379.

Tan, J. et. al (2012). "A new handwritten character segmentation method based on nonlinear clustering", Neurocomputing, 89, pp. 213–219.

Tomoyuki, H., Takuma, A. & Bunpei, I. (2007). "An analytic word recognition algorithm using a posteriori probability", in proceedings of the 9th international conference on document analysis and recognition, 2, pp. 669–673.

Vamvakas, G., Gatos, B., Pratikakis, I., Stamatopoulos, N., Roniotis, A. & Perantonis, S. J. (2007). "Hybrid off-line OCR for isolated handwritten Greek characters", In Proceedings of 4th IASTED international conference on signal processing, pattern recognition and applications, pp. 197–202.

References

Verma, B. (2002). "A contour character extraction approach in conjunction with a neural confidence fusion technique for the segmentation of handwriting recognition", in proceeding of the 9th international conference on neural information processing, 5, pp. 2459–2463.

Verma, B. (2003). "A Contour Code Feature Based Segmentation For Handwriting Recognition", In Proceedings of the 7th International Conference on Document Analysis and Recognition, Edinburgh, UK: IEEE Computer Society Press, pp. 1203-1207.

Verma, B., Blumenstein, M. & Ghosh, M. (2004). "A novel approach for structural feature extraction: contour vs.direction", Pattern Recognit Lett, 25(9), pp.975–988.

Verma, B., Blumenstein, M. & Ghosh, M. (2004). A novel approach for structural feature extraction: contour vs. direction, Pattern Recognition Letters, 25(9), 975-988.

Verma, B., Blumenstein, M. (2008). "Pattern Recognition Technologies and Applications: Recent Advances", Information Science Reference (An Imprint of IGI Global Publications), Hershey, New York, pp. 1-16.

Verma, B., Gader, P. & Chen, W. (2001). "Fusion of multiple handwritten word recognition techniques. Pattern Recognit Lett, 22(9), pp. 991–998.

Verma, B., Gader, P. (2000). "Fusion of multiple handwritten word recognition techniques", in neural networks for signal processing X, 2000. Proceedings of the 2000 IEEE Signal Processing Society Workshop, 2, pp. 926–934.

Viard-Gaudin, C., Lallican, P. M. & Knerr, S. (2005). "Recognition-directed recovering of temporal information from handwriting images", Pattern Recognit Lett, 26(16), pp. 2537–2548.

Vinciarelli, A. (2002). "A survey on off-line cursive word recognition", Pattern Recognit, 35(7), pp. 1433–1446.

Vinciarelli, A., Bengio, S. & Bunke, H. (2003). "Off-line Recognition of Large Vocabulary Cursive Handwritten Text", Proceedings of the 7th International Conference on Document Analysis and Recognition, Edinburgh, UK: IEEE Computer Society Press, pp. 1101-1107.

Wang, X., Ding, X. & Liu, C. (2005). "Gabor filters-based feature extraction for character recognition", Pattern Recognition, 38(3), pp. 369-379.

References

Xiao, X. & Leedham, G. (2000). "Knowledge-based Cursive Script Segmentation", Pattern Recognition Letters, 21(10), pp. 945-954.

Yanikoglu, B. & Sandon, P. A. (1998). "Segmentation of Off-Line Cursive Handwriting using Linear Programming", Pattern Recognition, 31(12), pp. 1825-1833.