Off-Line Handwritten Character Recognition of Manipuri Script

A Thesis submitted
for the award of degree of
Doctor of Philosophy

By:

Th.Tangkeshwar Singh
Reg. No. 9020358

Under the Guidance of:

Dr. Seema Bawa
Professor, CSED,
Thapar University,
Patiala

Dr. P.K Bansal
Director General
Quest Group of Institutions,
Mohali, Punjab

Prof. Renu Vig
Director UIET,
Punjab University,
Chandigarh

Computer Science and Engineering Department
THAPAR UNIVERSITY, PATIALA-147 004, INDIA
August, 2017

CERTIFICATE

I hereby certify that the work which is being presented in this thesis, "Off-Line Handwritten Character Recognition of Manipuri Script", for the award of degree of "Doctor of Philosophy", submitted to the Department of Computer Science and Engineering, Thapar University, Patiala, is an authentic record of my own work, carried out under the supervision of Prof. Seema Bawa, Prof. Renu Vig and Prof. P.K. Bansal. It refers to other researchers' works, which have been duly listed in the reference section.

The matter presented in this thesis has not been submitted in part or full to any other institute or university, for the award of any other degree.

Th. Tangkeshwar Singh

(Th. Tangkeshwar Singh)
Reg. No. 9020358

This is verified that the above statement made by the candidate is correct and true to the best of my knowledge.

Seema Bawa

(Prof. Seema Bawa)

Supervisor

Prof. Seema Bawa
Computer Science and
& Engineering
Department,
Thapar University
Patiala

P.K.Bansal

(Prof. P.K.Bansal)

Supervisor

Dr. P.K. Bansal
Director General,
Quest Group of
Institutions,
Mohali

Renu Vig

(Prof. Renu Vig)

Supervisor

Prof. Renu Vig
Director,
UIET,
Punjab University
Chandigarh

ABSTRACT

As computers become increasingly integrated into everyday life and since we want computers to be genuinely intelligent and to interact naturally with us, therefore the benefits of automatic recognition of handwritten digits and characters are obvious.

This thesis initially presents a brief report on linguistic survey of Manipuri script along with historical background and revival movement of the Script. Many aspects of handwritten digit and character recognition research for English as well as some Indian scripts have also been reviewed.

In this thesis, Handwritten Character Recognition System of Manipuri Script, HCRMS, has been presented for segmenting lines, words, non-touching characters and isolated digits and recognizing the handwritten isolated digits and non-touching characters. Analysis of different strategies for segmenting non-touching characters of handwritten word in different zones with the analysis of vertical, horizontal projection profiles and connected component analysis are given. After size normalization of the extracted component, probability features based on the size and slant invariant signatures features are extracted. The handwriting recognition results for digits and characters using K-L divergence with probabilistic features are presented. Fuzzy feature extraction technique from the resized component with zoning is also presented.

Then Hybrid feature is proposed by combining the two feature sets for the better recognition rates of the characters using feed forward backpropagation neural network.

The experimental results show that the choice of the features affects the performance of the classifier and the proposed hybrid feature set gives better recognition rate. The generalization of the recognition process has been improved with the size and slant invariant signatures features of the probabilistic feature method. Experimental results indicate that the proposed recognition system performs well and is robust to the writing variations that exist between persons and for a single person at different instances, thus being promising for user independent character recognition and tolerant to random noise degradations of the characters.

ACKNOWLEDGEMENTS

First and foremost I would like to express my profound appreciation and gratitude to my supervisor Prof. Seema Bawa, Department of Computer Science & Engineering, for her continuous motivations, guidance, encouragement and assistance at so many levels throughout my study at Thapar University. This work would not have been possible without her valuable support.

I would also like to express my profound appreciation and gratitude to my supervisor Prof. Renu Vig, Director, UIET, Punjab University, for her continuous motivations, guidance, and encouragement and particularly for giving me generously of her time and knowledge and correcting my errors.

I would also like to express my profound appreciation and gratitude to my supervisor Dr. P.K. Bansal, Director General, Quest Group of Institutions, Mohali, Punjab, for his continuous guidance, encouragement, assistance and particularly for his valuable advice and observations.

I would like to express my gratitude to the late Prof. H.S. Kasana for supervising me during the second quarter of this thesis.

I would like to express my profound gratitude to Prof. C. Amuba Singh, Former Vice-Chancellor, Manipur University, Canchipur, Imphal and Prof. H.N.K Sarma, Former Vice-Chancellor, Manipur University, and Prof. Adya Prasad Pandey , Vice Chancellor, Manipur University, for their valuable advice, support and encouragement. I also would like to express my best regards to Prof. N. Rajmuhon Singh, Dean, School of Mathematical and Physical Sciences, Manipur University.

My best regards to Prof. Prakash Gopalan, Director, Thapar University for valuable support and encouragement.

I earnestly acknowledge Thapar fraternity for providing resourceful research environment at Thapar University. My sincere gratitude towards the Doctoral Committee, Prof. R.K. Sharma, Dean of Faculty Affairs and Prof. A.K. Chatterjee, ECE Department and Dr. Maninder Singh, Associate Professor & Head, Department of Computer Science & Engineering, who persistently monitored the research progress, ensured quality work and pegged their invaluable suggestions.

I also would like to express my best regards to Dr.R.S Kaler, Deputy Director and Dr. O.P Pandey, Dean, Research and Sponsored Projects, for their valuable support and encouragement.

I would like to thank all academic, administrative and technical staff from the Department of Computer Science and Engineering (CSED), Thapar University, for their encouragement, assistance and friendship throughout my candidature. I would also like to thank everyone else in the Research Lab. CSED for providing an ideal environment, both socially and technically, in which to conduct research.

I would also like to thank all academic, administrative, technical staff and students from the Department of Computer Science, Manipur University for their encouragement, assistance and friendship and particularly for helping in building local database and I would like to thank all members of MIT, Imphal for their valuable support.

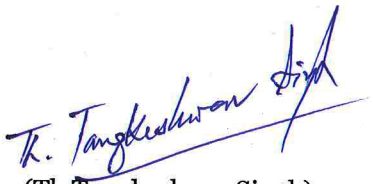
I wish to extend my appreciation to all my juniors and seniors for their valuable contributions in many different ways.

I would also like to express my sincere gratitude to all the senior citizens, and brothers for providing me documents related to Manipuri Script. I am thankful to all friends for friendship, encouragement and supports.

Last, but not the least, I am heartily thankful to all my family members for their love, encouragement and supports. I remain, indebted to my father late Th. Rajendra Singh and my mother late Th. Taruni Devi, for their love and encouragement.

Finally, I thank my wife Th. Binarani Devi and our son Syber Thokchom for their love and care from the core of their heart.

Above all, I am grateful to Almighty for guiding me and blessing me.



(Th.Tangkeshwar Singh)

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

1.1 Handwriting recognition - general presentation

Handwriting form of a language is used in notebooks, personnel letters, on envelopes, cheques etc. Handwriting has been the most natural mode of collecting, storing, and transmitting information. The concept of handwriting has existed for millennia, for the sole purpose of expanding human memory and facilitating communication. To imitate the human ability to read and recognize printed and handwritten character is the focus of research areas of many researchers from academic circles for many decades. The fundamental characteristics of handwriting are three-folds. (a) It consists of artificial graphical marks on a surface; (b) Its purpose is to communicate something; (b) This purpose is achieved by virtue of the mark's conventional relation to language [1]. Much of culture and civilization may be attributed to the advent of handwriting.

The interest devoted to this field is not explained only by the exciting challenges involved, but also the huge benefits that a system, designed in the context of a commercial application, could bring. Taking into account the possible importance of these documents, the benefits of automatic recognition of handwritten texts are obvious and now it serves not only for communication among humans but also serves for communication of humans and machines. Fuelled by the curiosity to uncover the secrets of the human mind, many researchers began to focus their attention on attempting to mimic intelligent behavior. One such example was the attempt to imitate the human ability to read and recognize printed and handwritten matter.

As technology has advanced and proliferated, computers have become an inescapable part of daily life. They promise to enhance communication and increase efficiency. As computers become increasingly integrated into everyday life, there has been a major push to make them as accessible and

"user friendly" as possible. Clearly, ease of communication between humans and computers will have a direct effect on the ability of computers to become a universal tool, as indispensable and ubiquitous as the mobile phone or the automobile. Ubicomp systems where computer interfaces that support more natural human forms of communication (e.g. handwriting, speech, and gestures) are beginning to supplement or replace elements of the GUI interaction paradigm. Pervasive computing and wearable computers are evolving. Some systems are MIT's intelligent room project, intelligent coffee cup; Stanford's interactive workspaces project, etc. Systems using interactive gestures with radio frequency identification (RFID) and near field communication (NFC) are reported. According to Norman Weinstein, *Technology Review*, in Affective Computing, *MIT Press*, by Rosalind W. Picard, reported that the latest scientific findings indicate that emotions play an essential role in decision making, perception, learning, and more –that is, they influence the very mechanism of rational thinking and if we want computers to be genuinely intelligent and to interact naturally with us, we must give computers the ability to recognize, understand, even to have and express emotions.

As new uses for computers are envisioned, user interface is hindering computers in their quest towards seamless integration into society. While well-established interface methods, such as the mouse and the keyboard, have proven to be extremely successful, they have the disadvantage of being foreign and unnatural to novice computer users. Newer technologies such as speech recognition have shown great promise, but current implementations leave much to be desired. Handwriting, however, has prevailed as one of the most effective ways to communicate. This is because it has the dual benefit of being both natural to most people, as well as being relatively permanent. Another obstacle that computers have faced, in their quest for total societal integration, comes from the business world, where the presence of handwriting is readily apparent. Whether it is a quick note jotted down on a piece of scratch paper, a handwritten fax, or corrections scribbled on the front of a report, the use of handwriting continue to be extremely prevalent in business. The dominance of handwriting is further multiplied by the

massive amount of handwritten paper documents filed away in storerooms and filing cabinets of businesses across the country. Though the use of electronic communication is quickly becoming indispensable to the business world, computers will not gain recognition as a universal tool until they can effectively and efficiently deal with the aforementioned types of documents. Today we know mobile phones are having many advanced features such as on line handwriting recognition.

Machine simulation of human reading has been the subject of intensive research for the last three decades. However, the early investigations were limited by the memory and power of the computer available at that time. With the rapid advancement of information technology, there has been a dramatic increase of research in this field for many language scripts of developed and developing countries since the beginning of 1980s.

1.2 Off-line and on-line handwritten data

According to the way handwriting data is generated in the domain of handwriting recognition, we generally refer to two categories of handwritten data: off-line and on-line.

Here, off-line data refers to documents which are obtained by scanning or photographing some hard copies of documents with a digital camera or a scanner and which are generally stored in the form of images. It means transforming a language represented in its spatial form of graphical marks into its electronic representation. Documents used by this process are generally based on some physical support such as paper, in most of the time and can be historical documents, journals and magazines, music score, administrative forms, etc. Digitization process of off-line documents is shown in Figure 1.1(a). The resulting images may be in color, grayscale or binary according to the capturing device and the objective of digitization. An illustration of digitization of off-line historical document is shown in Figure 1.1(b).

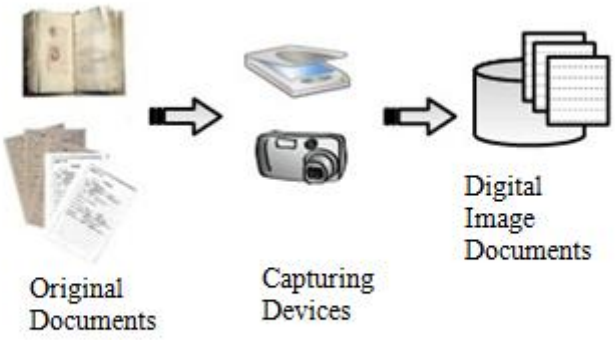


Figure 1.1(a): Digitization process of off-line documents.

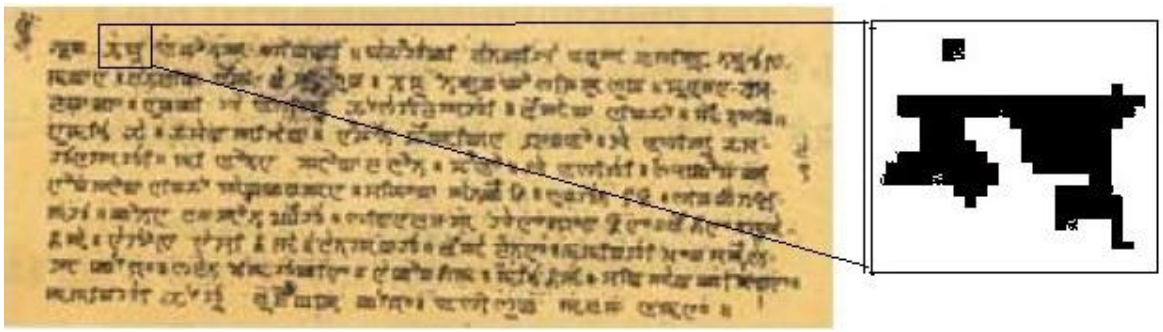


Figure 1.1(b): An example of a degraded document image

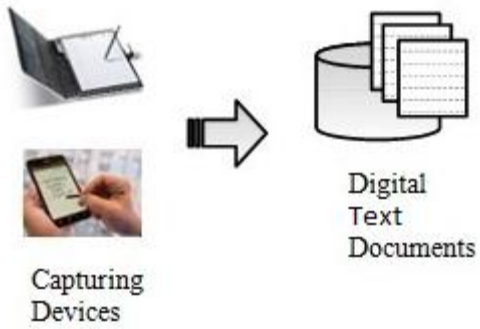


Figure 1.2: On-line handwriting capturing process

On the other hand, on-line handwriting data are captured during the writing process by a special pen on an electronic surface. On-line data refers to the data which is acquired through modern devices such as smartphones

or tablets, and for which the information is represented by a sequence of points describing the trajectories of handwriting, as illustrated in Figure 1.2.

The on-line data is generally known as on-line signal in the domain of handwriting recognition. The coordinates (x, y) of each point in the on-line signal is captured and it describes its spatial position. Some capturing devices may also provide the acquisition time and/or pen pressure and/or pen inclination. Some capturing devices only rely on spatial sampling (DPI, Dots Per Inch). In this case, only the coordinates of points are provided and the distance between two consecutive points is homogeneous. Some other capturing devices, rely on temporal sampling (Hz, number of samples per second). As a consequence, distances between two consecutive points are heterogeneous. In this case, the density of points during slow writing is more important than the density of points during rapid writing.

The pen-tip movement is detected along with pen-up/pen-down states during the collection of online data. When the pen touches the digitizer (writing pad), it is a pen-down state and a pen-up state is sensed when the pen is lifted off. A stroke is the set of points captured between successive pen-down to pen-up states. Additional information such as the speed of writing, stroke number and order can be utilized for recognizing online handwritten data.

In the literature, the recognition of off-line handwriting is more complex than the on-line case due to the presence of noise in the image acquisition process and the loss of dynamic or temporal information such as the writing sequence and the velocity. Therefore, many authors have focused on recovering on-line signal from off-line image, as illustrated in Figure 1.3(a) [13, 14, 31].

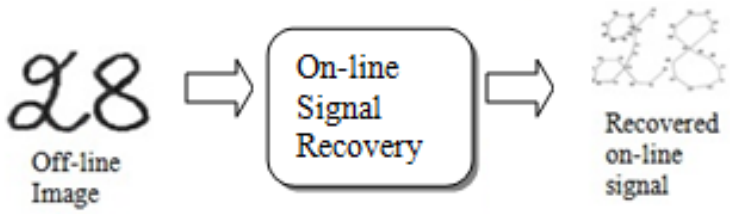


Figure 1.3(a): On-line signal recovery

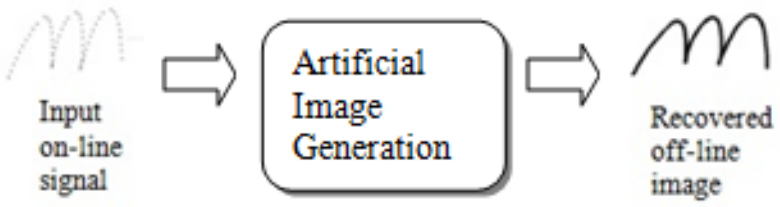


Figure 1.3(b): Off-line signal recovery

It is necessary to guess/reconstruct some temporal information in order to deducethe on-line signal based on the off-line image. The effectiveness of the off-line handwriting recognition can be improved from the recovered on-line signal. On the other hand, it is easier to create an off-line image from an on-line signal of handwriting. By connecting the sequence of points, an artificial image can be simply created as shown in Figure 1.3(b). In order to take profit of their complementarity, some authors try to use both on-line signal and its recovered off-line image.

1.3 Categories of Manipuri handwriting styles

The three categories of handwriting styles are:

1. ***Boxed discrete digits:***

Since the digits are written in pre-defined boxes, they are also known as isolated digits. Segmentation process is not required, so this category is the easiest one compared to other categories. The pixels (for off-line image) or strokes (for on-line signal) contained in each box represent one digit as illustrated in Figure 1.4(a). Hence, an ICR can be directly applied.

Figure 1.4(a) Boxed Digits

2. ***Spaced discrete digits and characters:***

In this case, there is no pre-defined box. However, there is a writing constraint that a blank space between characters is provided. In this case, in order to segment a handwritten word into a sequence of characters, a simple character segmentation method based on blank space detection or connected component extraction can be applied. Then, each segmented shape can be directly submitted to an ICR.

Figure 1.4(b) Spaced isolated digits

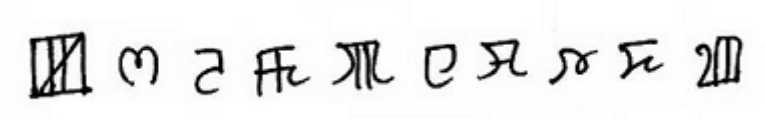


Figure 1.4(c) Spaced isolated characters

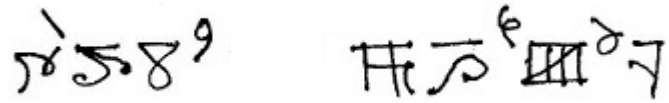


Figure 1.4(d) Spaced discrete characters of words

3. Mixed Cursive and Discrete:

It is unconstrained handwriting and is obviously much more difficult than the other categories. Without any writing constraint, writers can use their own writing style. Due to the huge variations of possible handwriting styles, the complexity of connections between characters is important. Therefore, segmenting an input handwritten word into a sequence of characters is much more complicated compared to the previous categories since segmentation points are generally unknown as illustrated in Figure 1.4(e). The words are written with touching characters.

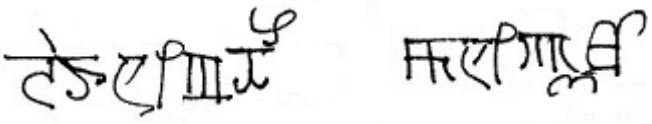


Figure 1.4(e) Examples of unconstrained handwritten Manipuri (Meetei/Meitei) words.

Apart from the above mentioned categories, off-line handwritten character recognition systems are classified as follows:

1. Writer dependent and independent systems:

Writer-dependent systems are trained to recognize handwriting of a single individual whereas the goal of a writer-independent off-line system is to recognize handwriting of a variety of writing styles. In case of writer-

independent systems, they are able to recognize handwriting that they may not have seen during training. Where as in writer-dependent systems, handwriting samples of a single individual is being trained and tested with the system. In general, better accuracy rate is presented by writer dependent systems as compared to writer independent scenarios. Obviously, constructing writer independent systems is more difficult than writer dependent systems. It is the fact that writer independent system is expected to handle much greater varieties of handwriting styles.

2. ***Lexicon based and lexicon free systems:***

In case of some word recognition systems, it employs a small or fixed lexicon such as postal address interpretation and bank check reading. By matching the recognized word against a word contained in the lexicon, the accuracy is increased. But the recognition accuracy depends upon the size of the lexicon. It is noted that the recognition accuracy reduces with increasing lexicon sizes. On the other hand, the recognition is performed without the aid of a dictionary in lexicon-free systems.

1.4 Terms Associated with Computerised Handwriting Readers

Three common terms that researchers use when dealing with automatic reading of handwriting are: Recognition, Interpretation and Identification [1]. The research in this thesis explores the *Recognition* of handwritten characters of Manipuri (Meetei/Meitei) script. It is the act of transforming the graphical marks associated with human handwritten script into symbols that are stored on a computer system in the form of 8-bit ASCII code or 16-bit Unicode.

Interpretation is associated with computerised readers that are used to determine the meaning of a section of handwritten text. The interpretation of a handwritten address on a piece of mail is a good example.

Lastly, Handwriting *Identification* refers to the task of identifying the author of a fragment of handwriting. An example application includes signature verification, i.e. the task of identifying whether a piece of

handwritten signature belongs to a particular writer. Another area where identification is necessary is in the field of forensics, where it may be required to identify the handwriting of a suspect. Identification stands out from the three forms of computerised handwriting analysis, the reason being that handwriting identification focuses on the unique nature of a fragment of handwriting sample to differentiate between two possible authors. Conversely, handwriting interpretation and recognition require that variations be eliminated from fragments of handwriting. This allows the handwriting to become uniform, which makes it easier to extract the message or meaning.

1.5 Problem Statement

One of the most impressive capabilities of our brain is the ability to recognize patterns in nature. The brain is considered to be one of the most complex systems developed throughout the evolution of life in this planet. Throughout the history of the human search for knowledge, recognizing patterns in nature and trying to understand how those patterns are related into a set of rules and laws and how systems emerges in general, has been one of the main rationale behind the vast range of theories and concepts that we have made of the natural world.

A brain contains over one hundred billion computing elements called neurons. Exceeding the stars in our Milky Way galaxy in number, these neurons communicate throughout the body by way of nerve fibers that make perhaps one hundred trillion connections called synapses. This network of neurons is responsible for all of the phenomena that we call thought, emotion and cognition, as well as for performing myriad sensorimotor and automatic functions [2].

What are patterns and what do we understand about them? A general definition of a pattern can be stated as a distinct set of characteristics or behaviors that can be differentiated and classified from a collection of information or data. This definition is very broad and subjective. In this way, any object, idea, rule or even a word in the dictionary can be considered to be a pattern. This makes patterns intrinsically dependable on the observer.

Even a physical or observable characteristic such as size, weight or form of an object is then a kind of a pattern. Furthermore, the laws of physics in itself can also be considered as concise way to describe the natural pattern of dynamical behavior. In this context, the human mind is a very elaborate and sophisticated system capable of identify, understand and discover pattern. Pattern recognition as a scientific discipline is related to machine intelligence systems built for decision making processes, whose purpose is the classification of objects into a number of categories or classes. The objects depend on the application: it can be images or signal waveforms or any type of measurements that need to be classified. The recent and fast developments of computer technology and resources made possible various practical applications of pattern recognition, which in turn contributed to the demands for further theoretical developments. Some of the areas of increased interest in pattern recognition are machine vision, character recognition, medical diagnosis and speech recognition.

In the coming years, the ubiquity of intelligent systems is certain to have a profound impact on the ways in which human-made systems are conceived, designed, manufactured, employed, and interacted with. This is the perspective in which the contents of Neuro-fuzzy and Soft Computing should be viewed.

Soft computing is an emerging approach to computing which parallels the remarkable ability of the human mind to reason and learn in an environment of uncertainty and imprecision (Lotfi A. Zadeh, 1992) [3].

The essence of soft computing is that unlike the traditional, hard computing, soft computing is aimed at an accommodation with the pervasive imprecision of the real world. Thus, the guiding principle of soft computing is to exploit the tolerance for imprecision, uncertainty, and partial truth to achieve tractability, robustness, low solution cost, and better rapport with reality. In the final analysis, the role model for soft computing is the human mind.

Soft computing is not a single methodology. Rather, it is a partnership. The principal partners at this juncture are fuzzy logic (FL), neurocomputing (NC), and probabilistic reasoning (PR), with the latter

subsuming genetic algorithms (GA), chaotic systems, belief networks, and parts of learning theory. The pivotal contribution of FL is a methodology for computing with words; that of NC is system identification, learning, and adaptation; that of PR is propagation of belief; and that of GA is systematized random search and optimization.

In the main, FL, NC, and PR are complementary rather than competitive. For this reason, it is frequently advantageous to use FL, NC, and PR in combination rather than exclusively, leading to so-called hybrid intelligent systems. At this juncture, the most visible systems of this type are neuro-fuzzy systems.

Many studies on the recognition of writings such as characters, words or strings of digits documents of non-Indian languages such as English, Chinese, and Japanese etc. can be found in the literature. Nowadays more number of literatures has been reported on recognition of Indian language scripts. Main reasons for this slow development compared with the non-Indian language documents could be attributed to the complexity of the shape of characters of these scripts and also the large set of different patterns that exist in these languages, as opposed to English.

Our visual system is capable of recognizing images in an impressively fast and accurate way. We recognize various objects in the daily environment without much effort. However even the task of handwritten character recognition that is regarded as simple for a human, to a machine it is still a problem which depends on various constraint and parameters and still without a definite solution. Although the HCR problem is relatively simple since it is defined in a two state grid without gray levels or colors. We address and propose new methods to deal with it. Also our interest on this problem is in understanding the difficulties involved on constructing pattern recognition systems in general framework.

In this thesis, we present our study of the off-line handwritten character recognition (HCR) of Manipuri (Meetei/Meitei) script.

1.6 Research Objectives and Scope of this Thesis

India is a multilingual country with 22 constitutionally recognized languages. Proliferation of Information Technology in the society requires availability of user-friendly tools and technologies that can enable using their languages on the computers deriving the maximum advantage of the Information & Communication Technology (ICT) revolution [35]. Development of Language Technology in India has been taken a quantum leap and in future, India is poised to become multilingual computing hub of the world. The software Localisation market is growing at a very fast rate and this expansion has created a demand for *specialized human resources* having existing programme unable to fill. With this objective it is proposed that necessary human resources need to be generated in order to meet the demand-gap in the Language Technology market in India.

There are several contributed works in the literature for Indic scripts such as Kannada, Bangla, Telugu, Devanagari, Gurmukhi, Malayalam and Tamil. In particular, Indian scripts comprise compound symbols resulting from vowel-consonant combinations and in many cases, consonant-consonant combinations. Moreover, the closeness between some of the characters requires for sophisticated algorithms.

Thus arises the need for developing off-line handwritten character recognition (HCR) of the Manipuri script.

The objective of our research is to propose off-line handwritten character recognition (HCR) of the script. In particular we focus on two important aspects for offline handwritten character recognition of Manipuri scripts: (1) Feature Extraction and (2) Selection of the architecture of Neural Networks Classifiers for the character sub-sets. Segmentation of lines of text and segmentation of isolated digits or characters from the line with connected component analysis is also adequately addressed. The 2nd category of Manipuri handwriting style that is the spaced discrete digits and characters are types of handwritten characters addressed for recognition in this research.

In this work, we propose a new feature vector and it is named as Hybrid features (HF). Hybrid features (HF) is the combination of two feature vectors namely, Probabilistic features (PF) and Fuzzy features (FF). Probabilistic features (PF) are extracted from the resized binary image object and these are rotation invariant and decent amount of scale invariant.

Results of the experiments conducted for recognition of random samples against the training set based on the Kullback-Lieibler divergence using the extracted probabilistic features for isolated sample digits as well as isolated sample characters are provided. Fuzzy features (FF) extraction technique based on zoning is also presented.

Experimental results on performances of Feature Selection methods are sufficiently addressed. The performance evaluations of the proposed HF have been made with respect to the other features. Also, the performance observations of Hybrid (HF) features and Gabor wavelets features are highlighted.

There are 56 character classes of Manipuri Script. It consists of 27 consonants (27 alphabets of Iyek Ipee), 8 final consonants (Lonsum Iyek), 8 vowels (Cheitap Iyek letters), 3 khudam Iyek (Cheikhei, Lum, and Apun) and 10 numerals.

The total 55 character data classes (without the Lum symbol) of Manipuri Script are divided into the following categories for training with 5 different Feed Forward Multilayer Perceptrons (MLPs) of backpropagation algorithm. Two type of training algorithm are investigated. One MLP with gradient Descent backpropogation with adaptive learning rate and the other with resilient backpropagation training algorithm. The 5 different MLPs used for training the character sub-sets are listed below:

MLP1 (A):

10 Numerals or Digits classification (Cheising Iyek):

10 classes (1, 2, 3…9, and 0)

MLP2 (B):

27 Characters (27 alphabets of Iyek Ipee) + 7 Lonsum Iyek:

34 classes (1,2,3….34)

(Ee Lonsum of Lonsum Iyek is dropped as it is equivalent to Ee (I,E) of 27 alphabets Iyek Ipee.)

MLP3 (B1):

Only 27 Characters (27 alphabets of Iyek Ipee) without 7 Lonsum Iyek characters:

27 classes (1,2,3,...27)

MLP4(C): 8 vowels (Cheitap Iyek letters) + 2 khudam Iyek (Cheikhei symbol for Fullstop + Apun symbol for Sign of Ligature):

10 classes (1, 2, 3...9 and 0)

MLP5 (D):

The total 44 character classes of Manipuri Script without the 10 numerals:

44 classes (1, 2, 3...44)

In this work, we take a step forward in the goal of developing a robust writer-independent, lexicon-free recognition system for Manipuri words with isolated characters.

1.7 Outline of the Thesis

This thesis consists of seven chapters (including the introduction and the conclusion) and appendix. The current chapter (Chapter 1 Introduction) outlines the problem statement, the research objectives and outline of this thesis.

In Chapter Literature Survey, a brief report on linguistic survey has been given and an introduction to the Manipuri script is described along with a brief historical background and revival movement of the Script. A comprehensive survey of handwritten character recognition research particularly feature extraction and selection, classification, recognition, and verification methods for handwritten characters and numerals is given. Literature survey on the HCR system related to document image analysis and HCR for English and a brief survey of HCR research in Indian scripts are presented.

In Chapter 3, Preprocessing and Segmentation, the importance of preprocessing is highlighted. Several preprocessing steps are discussed. Segmentation of lines of handwritten digits from the deskewed input file and segmentation of isolated digits from the extracted line are presented. And

also segmentation of lines of handwritten characters from the deskewed input file and segmentation of words from the segmented line and segmentation of isolated characters but overlapped in the segmented word are also presented.

In Chapter 4, Probabilistic features (PF) extraction technique based on rotation invariant and decent amount of scale invariant is presented. Results of the experiments conducted for recognition of random samples against the training set based on the Kullback-Lieibler divergence using the extracted probabilistic features for sample digits as well as sample characters are given. Fuzzy features (FF) extraction technique based on zoning is also presented. Then, Hybrid Feature (HF) Extraction Technique is proposed and presented.

In Chapter 5, Performance analysis of the proposed hybrid features of the Recognition System based on the experimental results is presented. The recognition results of digits as well as characters based on proposed hybrid features are significantly promising.

The overall steps required for the recognition of digits and characters from an input image page are discussed. The investigations and performances of the proposed feature set namely Hybrid (PF+FF) is presented and lastly the performance observations of Hybrid (PF+FF) features and Gabor wavelets features are highlighted.

In Chapter 6, the handwritten character recognition system of Manipuri Script (HCRMS) is presented. The recognition systems of an input image page for digits and characters using the trained ANNs classifiers and the recognition systems for a word consisting of isolated characters and word consisting of overlapped characters are presented in this chapter.

In Chapter 7, Conclusion and Future scope, conclusions are drawn based on the in-depth analysis and discussion from Chapter 6. A summary of thesis contributions is given and discussions on future scope are also addressed.

Chapter 2
Literature Survey

This chapter includes a literature survey of the Manipuri (Meetei) script as well as the literature review of the handwritten character and digit recognition.

2.1 Literature Survey of Manipuri Script

The history of Kangleipak (Imphal Valley) may be dated roughly 10,000 years B.C. and about 12,000 years ago. The Imphal valley of Kangleipak was under water for many thousand years. Before that the people of Kangleipak lived on the surrounding hills, the centre of dispersion of population being the Koubru (Koupalu in the puya) mountain to the north west of Kangleipak. Living on the mountain, the people of Kangleipak developed certain degree of civilization approaching very much to written history period [4].

The name of the Race, Meetei was developed only sometimes in 2000 B.C. when they began to settle some thousands of years in the valley of Kangleipak, now called Imphal valley. These thousand years were in the spoken period of History of Kangleipak, called proto-History generally, before they become the Meetei and writing scripts invented. The written history of Kangleipak began around 2000 B.C. and this is supported by clinching evidence of Kanglei Indigenous written literary evidence [4]. The legends and traditions of a people are very much a part of the history of the people and also of the country in which the particular people lived. The legends and traditions of the people take the forms of stories told to young generations for centuries without interruption. The beginnings of history of all people of the earth are all legends and traditions immediately before the written history, the real history in black and white after invention of the writing script. The beginnings of all written histories are all unwritten spoken or verbal histories. The period of these spoken or verbal forms of history is called the Proto-history period. For thousands of years before the

invention of writing script and before coming into existence of real written history of Kangleipak the indigenous people of Kangleipak sung the legendary and traditional song as follows

Awang Koubru Assuppa,
Leima-Lai khunta Ahanpa,
Nongthrei Ma-u Linglipa,
Eerik mapan Tharipa,
He Lainingthou !

This is the legendary mythical song sung by the indigenous people of Kangleipak for some thousand years regarding their origin and original settlement area in Kangleipak. The legendary and mythical songs have some words like “Nongthrei” “Eerik” which include “r” from 35 scripts of the Hindu. The Meetei, upto 18th century upto the advent of Hinduism to Kangleipak, used and wrote only in 18 alphabets[4].The original actual words are “Nongthaklei” which means “Heavenly flower meaning living creatures including Human being”, “Eelik which means “Blood drop”. The meaning of the song, in the mythical and legendary meanings, is that the universal Father God plants and creates living creatures first on Koubru mountain and therefore the created human beings pray Lainingthou, the deathless King of Gods for creating them and sustaining them [4].

2.1.1 Manipuri (Meeteilon)

Manipuri, also called **Meeteilon** [4, 5], Meiteiron and Meithei [6] in linguistic literature, is the official language of the State of Manipur. It is the mother tongue i.e., the first language of the ethnic group Meetei. However, apart from the Hindu Meiteis and the Meetei following the traditional religion of Sanamahi, Meitei Pangals i.e., Manipuri Muslims also speak Manipuri as their mother tongue.

Although Manipuri native speakers mostly reside in the state of Manipur, there are native speakers in the neighboring North-Eastern states of India, notably in Assam, Tripura, Nagaland and West Bengal. In India, the total number of people who speak Manipuri as their mother tongue numbers 1,270,216 out of which 1,110,134 speakers reside in Manipur (census of India, 1991).

It is a tonal language of Tibeto-Burman language family [4, 5]. The uniqueness of the language can be understood by having three different sentences but using the same phoneme "mapal" with three different tonal variations in which "p" has high or rise, fall or medium and heavy tonal variations. When the phoneme "mapal" is pronounced with high or rise of 'p', it means "flower". Example is "Lei mapal yamna satle."(There are a lot of flowers blooming"). When the phoneme "mapal" is pronounced with fall or medium tone of 'p', it means "plant". Example is "Mapal yamna phareda masi panbise" (This tree is fully grown). And when the phoneme "mapal" is pronounced with heavy tone, it means "nearby". Example is "Turel mapal amadi pukhri mapal da chatkanu." (Do not go near the pond or river bank).

As a result of all the recommendations, in 16th April, 1980, the Governor of Manipur had been pleased to announce that Meetei script should be introduced in all schools right from the beginner's upto any higher level from this year (Govt. of Manipur, Gazette, Vide Order No.1/2/78-SS/E dated 16th April, 1980). However, this announcement was ineffective causing even many agitations. But the fruit for these agitations push the script a step forward toward maturation.

Many open discussions and seminars were held to bring out the collective ideas of resource persons. Many agitations have been launched by MEELAN (Meetei Erol Eyek Loinasillol Apunba Lup) during the Script movement. As a part of this movement, two days seminar to examine the feasibility of introducing Manipuri Script from Class I-V simultaneously in the schools of Manipur were conducted at Manipur University Recreation Hall on 28th-29th April, 2005 with recommendations for introducing Meetei Mayek from Class I to III. The Govt. of Manipur took decision on 14th May, 2005 for the implementation of Meetei Mayek in the state and it had been resolved to include Meetei Mayek as a compulsory subject in Class I and II from academic session 2006-07. A new elementary book "Sindam Lairik Ahanba" by Writers' Group, MEELAN (A-eba kanglup, MEELAN) had been published in Jun, 2007.

Mayek Expert Committee was formed by the then Chief Minister, Yangmaso Seiza in 16th Nov., 1978, -Manipur Gazette No.387 Imphal, Saturday, November 18,1978 (This script contains Iyek Ipee/Mapung Iyek, which have 27 alphabets (18 original plus 9 letters called Lom Iyek, derived from original 18 alphabets), Lonsum Iyek (8 letters), Cheitek Iyek (8 symbols), Khudam Iyek (3 symbols), Cheishing Iyek (10 numeral figures). In addition to these there are 6 vowel letters.

Many social scientists express their opinion of the need of four strong pillars of support for the use of the script by the people. The needed supports are – (i) Acceptance of the script to be used by the people. (ii) Support of the legalized script recommended by the experts (iii) Support of the State Govt. policies for the script and (iv) support of the of role of IT policies of the State.

Kartica 27,1900). 24 sessions of the Mayek Expert Committee had been held during the period 1st Dec., 1978 to 31st July, 1979 and last session of the committee had recommended the Scripts consisting of 27 alphabets and other allied supplements (uses of lonsum, cheitap, cheikhei, khudam and cheising, original numerical figures etc)[5]. The following are the documents for ready reference:-

(A) Govt. of Manipur, Secretariat: Education Department Order No. 1/38/SSE/78 Imphal, the 16th Nov., 1978.

(i) Orders by the Governor of Manipur No. 1/2/78-SS/E, Imphal, the 16th April, 1980, vide Manipur Gazette No. 33 Imphal, Tuesday, April 22,1980(Vaisakha 2, 1902) (i)& (ii) of Annexure to the Governor of Manipur order No. 1/2/78-SS/E. Dt.: 16th April, 1980.

(C) Govt. of Manipur Gazette No.40 Imphal, Friday, April 25, 1980 (Vaisakha 5, 1902)

Iyek Ipee

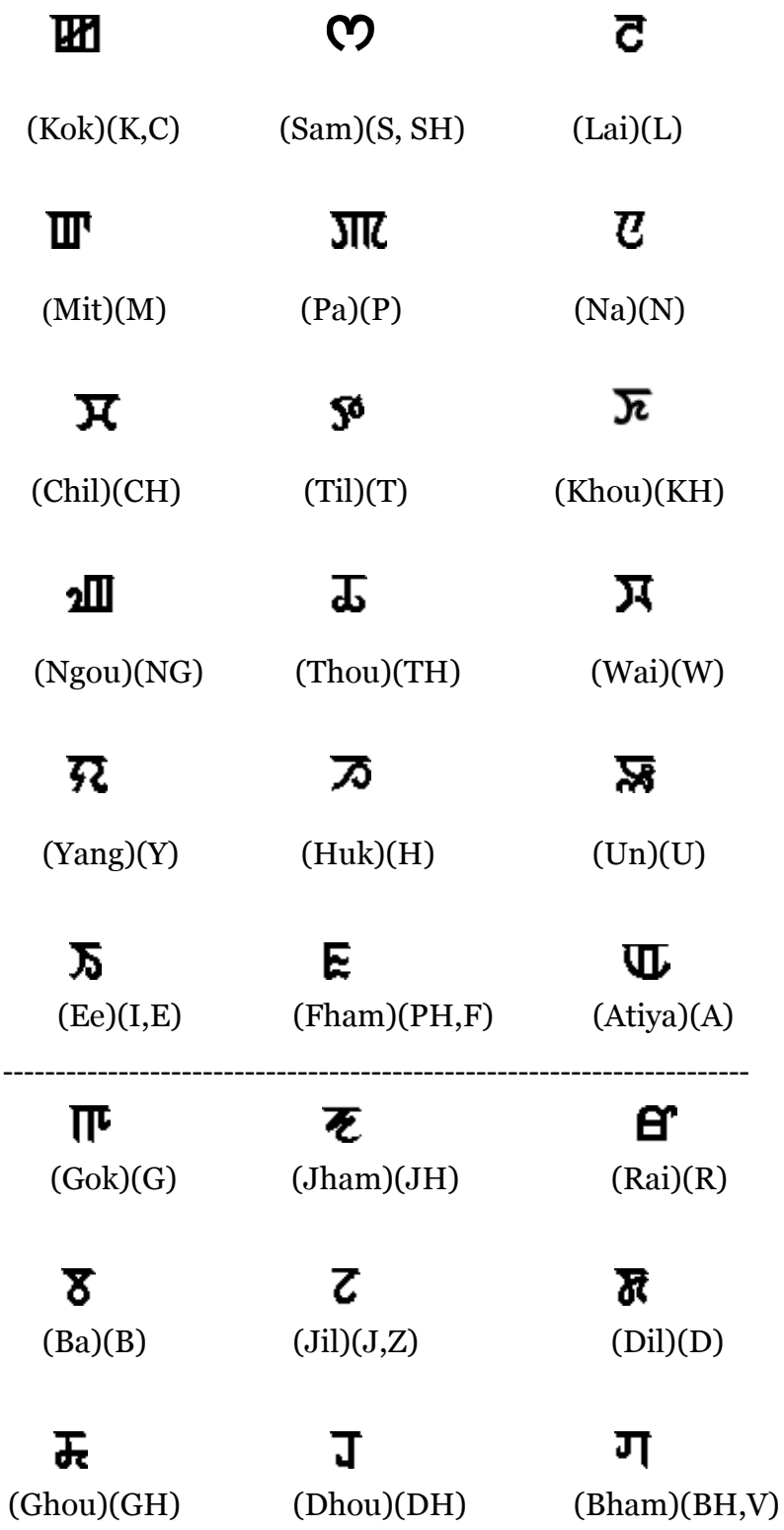


Figure 2.1: Twenty seven (27) letters (18 original + 9 evolved alphabets) of Meetei script (Iyek Ipee)

Lonsum Iyek

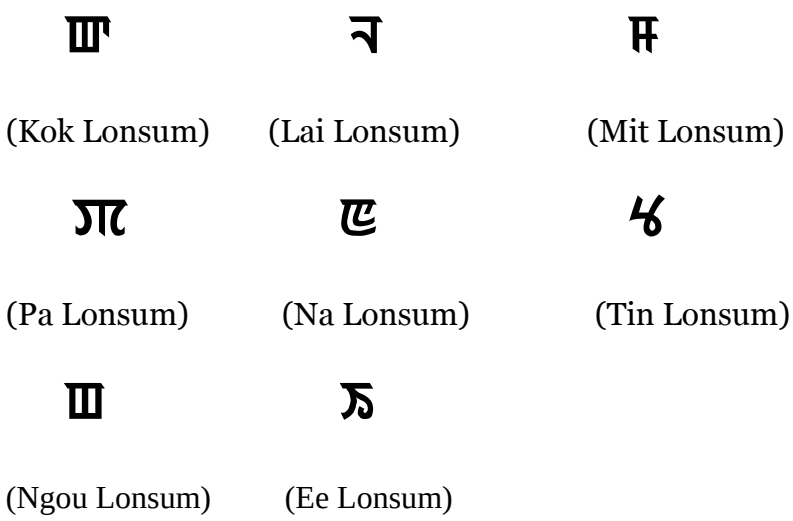


Figure 2.2: Eight Lonsum Iyek (8 letters)

Cheitap Iyek

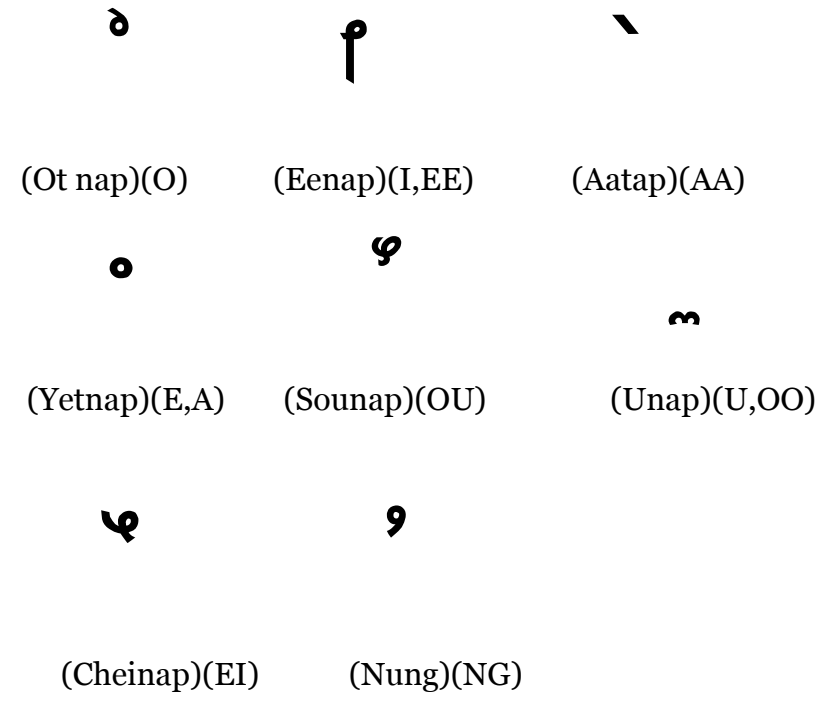


Figure 2.3: Eight Cheitap Iyek letters

Vowel Letters

ꯑꯥ (Aa) ꯑꯦ (Ae) ꯑꯤ (Ei)

ꯑꯣ (O) ꯑꯧ (Ou) ꯑꯪ (Ang)

Figure 2.4: Six vowel letters (Atiyaada Cheitap Taplaga Thoklakpa khonthok)

Source:

(i) Annexure to the Governor order No.1/2/78-SS/E Dated 16-4-80.

(ii) Annexure to the Governor order No. 1/2/78-SS/E Dated 16-4-80.

Khudam Iyek (Symbol)

꯫ (Cheikhei) (Fullstop)

꯭ (Lum Iyek) (Symbol for heavy tone)

꯱ (Apun Iyek) (Sign of Ligature)

Note: For other symbols international symbols be followed.

Figure 2.5: Three Khudam Iyek (3 Symbols)

Cheising Iyek(Numeral figures)

꯱	꯲
Ama (One)	Ani (Two)
꯳	꯴
Ahum (Three)	Mari (Four)
꯵	꯶
Manga (Five)	Taruk (Six)
꯷	꯸
Taret (Seven)	Nipal (Eight)
꯹	꯱꯰
Mapal (Nine)	Tara (Ten)

Figure 2.6: Ten Numeral Figures (Cheising Iyek)

ꯀꯣ	ꯀꯤ	ꯀꯦ	ꯀꯧ
(Ko)	(Ki)	(Ka)	(Kou)
ꯀꯨ	ꯀꯩ	ꯀꯪ	ꯀ꯫
(Ku)	(Kei)	(Ke)	(Kang)

Figure 2.7: Examples of Cheitap (semi vowels) when attached to ꯀ

Vowels	ꯥ aa	ꯪ e	ꯨ u	ꯤ i,ee	ꯩ ei	ꯣ o	ꯧ ou	꯫ ang
Modified shape	ꯥ	ꯪ	ꯨ	ꯤ	ꯩ	ꯣ	ꯧ	꯫
When attached to ꯀ	ꯀꯦ ka	ꯀꯪ ke	ꯀꯨ ku	ꯀꯤ ki	ꯀꯩ kei	ꯀꯣ ko	ꯀꯧ kou	ꯀ꯫ kang

Figure 2.8: Examples of vowel modifiers (Vowels and modified shape) when attached to ꯀ with English pronunciations.

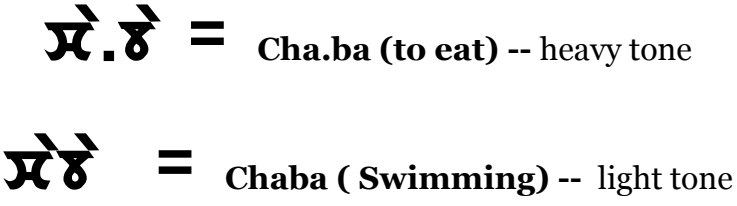


Figure 2.9: Examples of the use of Lum Iyek(Tonal symbol)

School can be written with Apun Iyek as

Figure 2.10: Example of writing a word "School" in kanglei script.

The report submitted by the Meetei Mayek Expert Committee constituted vide Government order of even number dated 16th November,1978 on the re-introduction of the study of Meitei Scripts numbering 27(Twenty seven) alphabets and its allied supplements had been approved by the Govt. of Manipur.-Manipur Gazette No.33. Imphal, Tuesday, April 22, 1980 (Vaisakha 2, 1902)

2.2 Handwritten Character Recognition Research

For the effective integration of computer into human society and for improving the human computer interaction, today handwritten character recognition (HCR) system has been developed for many languages in the world. HCR and optical character recognition (OCR) in a more general context are an integral part of pattern recognition. And we also review the research works related to different script. First of all, we review the historical progress of research for English script that has contributed greatly to the broader handwriting recognition community. After that, we review some advances in handwriting recognition research related to some Indian script in the course of the survey.

2.2.1 General Overview

One of the wider goals of the fields of artificial intelligence and machine learning is to enable computers to accomplish tasks which are natural to people, in accordance with the long-term goal of analyzing and emulating human intelligence or maybe even consciousness. As computer power increases over the years, their range of applications have also increased, one of the major goals of research has been to make computers easier to communicate with and thus to make their benefits available to a much greater number of people. Computers should be better able to interact with people and act within the world in a less constrained manner. These aims are reflected in the computer industry attempt to make computers more user friendly.

Optical character recognition (OCR) belongs to the family of techniques performing automatic identification. The traditional way of entering data into a computer is through the keyboard. However, this is not always the best nor the most efficient solution. In many cases automatic identification may be an alternative. Various technologies for automatic identification exist, and they cover needs for different areas of application. Some examples of these applications are speech recognition, vision systems and bar code. Handwriting is a natural means of communication which nearly everyone learns at an early age. In many situations, writing by hand is the fastest and most convenient way to communicate with another person. These facts have given raise to an enormous amount of research aimed at the automatic processing of handwritten data. HCR would provide an easy way of interacting with a computer requiring no training to use effectively, and a computer able to read handwritten characters would be able to process a host of data which at the moment is not accessible to computer manipulation. Even though various advances have been made in recent years, the recognition of handwritten characters still pose an interesting challenge in the field of pattern recognition.

Handwriting recognition has a variety of applications. Worldwide, the money spent for data entry from handwritten notes, forms and records runs into trillions of dollars. Looking at 2010 census of India, the data collection

by about 60,000 enumerators in handwritten forms took six months, whereas keying in the data into servers took over two years. In these and other applications mandating handwritten reports such as healthcare and industrial quality control and testing, an instant digital conversion to text will reduce huge costs as well increase productivity [10].

2.2.2 Historical Review

The prelude to OCR research is said to have commenced with an objective to develop reading machines for the blind. The original research attempted to develop devices capable of duplicating and transcribing various types of matter including printed, typed or handwritten characters. The origins of character recognition can actually be traced back in 1870. This was the year that C.R. Carey invented the retina scanner which was an image transmission system using a mosaic of photo-cells [152]. Two decades later the Polish P. Nipkow invented the sequential scanner which was a major breakthrough both for modern television and reading machines. However the first successful attempts at developing a device to aid the blind was made in 1900 by Tyurin [153] and next by Fournier d'Albe who demonstrated his optophone in 1912. The concept of OCR came soon after and was originally restricted to recognising printed characters. The first patent for this purpose was obtained in 1929 by Tausheck in Germany, followed closely by Handel in 1933. At that time certain people dreamt of a machine which could read characters and numerals. This remained a dream until the age of electronic computers arrived, in the 1950's.

A number of companies, including IBM, were conducting research into OCR throughout the early 60's, culminating in the first marketable commercial OCR system: IBM's 1418. Early systems were very constrained in the sense that they were bound to reading special, artificial fonts. These types of OCR systems are usually associated with the first generation of OCR. The second generation, on the other hand, was characterized by hand-printed character recognition capabilities. In the early stages, only numerals could be recognised by these pioneering machines. One such device was IBM's 1287 OCR system. This was the first of the second-generation machines, and one

of the most famous. It was originally exhibited at the World's Fair in 1965. Also, in this period Toshiba developed the first automatic letter sorting machine for postal code numbers and Hitachi made the first OCR machine for high performance and low cost.

Several companies, including IBM, Recognition Equipment, Inc., Farrington, Control Data, and Optical Scanning Corporation, marketed OCR systems by 1967. During the late 1960's, the technology underwent many dramatic developments, but OCR systems were considered exotic and futuristic, being used only by government agencies or large corporations. Systems that cost one million dollars were not uncommon.

For the third generation of OCR systems, appearing in the middle of the 1970's, the challenge was documents of poor quality and larger printed and hand-written character sets. Today, OCR systems are less expensive, faster, and more reliable. It is not uncommon to find PC-based OCR systems capable of recognizing several hundred characters per minute. More fonts can be recognized than ever with today's OCR technology and some systems claim to be omnifont - able to read any machine printed font. Less expensive electronic components and extensive research have paved the way for these new systems. Increased productivity by reducing human intervention and the ability to efficiently store text are the two major interest of research and development of OCR systems. The research areas include handwritten recognition and form reading. Reliable recognition of handwritten cursive script is now under intense investigation. In addition, research is being conducted in reading forms, that is, using all available information to formulate an interpretation of the document. For instance, the postal service research focuses on assigning ZIP codes to letter images which may not contain any ZIP code [15]. By understanding the various address fields such an assignment can be made. The use of contextual information in both handwritten recognition and form reading is essential.

The classification approaches generally used template matching in which the image of an unknown character is matched with a set of previously stored images. Matching techniques are based on the degree of similarity between two vectors in feature space. The employed techniques are: direct matching

[100], elastic and deformable matching [154] and matching by relaxation. Structural methods were also employed in character recognition along with statistical methods [155].

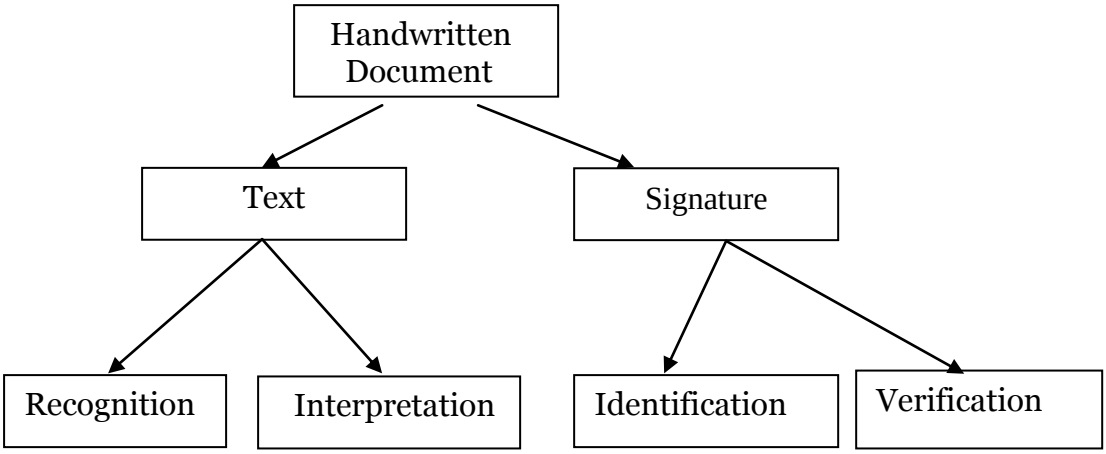


Figure 2.11: Terms associated with computerized handwriting readers.

It is generally accepted that the method of recognising on-line handwriting has achieved better results than it's off- line counterpart. This may be attributed to the fact that more information may be captured in the on-line case such as the direction, speed and the order of strokes of the handwriting. This information is not as easy to recover from handwritten words written on an analog medium such as paper. Although applications and techniques vary considerably, the general taxonomy of both on-line and off-line handwriting analysis is similar as shown in Figure 2.11.

Interpretation is associated with computerized readers that are used to determine the meaning of a section of handwritten text. The interpretation of a handwritten address on a piece of mail is a good example. *Identification* refers to the task of identifying the author of sample handwriting. An example application includes signature verification, i.e. the task of identifying whether a handwriting sample belongs to a particular writer, for example, signature verification of bank cheque. In both approaches, what can be of importance to be determined are the author of the writing, the words and meaning of what has been written or even both information. However, these two requirements demands for very different approaches, and the processing involved may not overlap. Techniques also differ

depending on whether the author is to be recognized from a signature or from a piece of text.

Another area where identification is necessary is in the field of forensics, where it may be required to identify the handwriting of a suspect. Identification stands out from the three forms of computerized handwriting analysis. The reason is that handwriting identification focuses on the unique nature of a fragment of handwriting to differentiate two authors. Conversely, handwriting interpretation and recognition require that variations be eliminated from fragments of handwriting. This allows the handwriting to become uniform, which makes it easier to extract the message or meaning.

The field of handwriting recognition is similar to the well-known subject of speech recognition, which is often classified along the lines of speaker dependence, vocabulary size and isolated or continuous word. Analogues for each of these constraints are also found in handwriting recognition. One of the main difficulty of a handwritten character recognition system lies in the enormous diversity of handwritten styles. It is much more difficult to devise a system to recognize many people's handwriting than one which need only recognize that of a single author. The HCR system has to be able to cope with the problem of multiple writers and be also able to adapt to new and different pattern. A robust recognition system is necessary to provide a general solution. Two other constraints to the problem are vocabulary size and segmentation of the characters. The task of recognizing words from a small lexicon is much easier than from a large lexicon. Thus an important criteria in assessing the system performance is the size of the lexicon used. In speech recognition the segmentation of continuous speech into its component words have been found to be very difficult. Similarly for handwriting recognition it is hard to distinguish the boundaries between letters. The task can be simplified by forcing the writer to use separated letters, to write in capitals or to write clearly separated capitals in pre-printed boxes [21].

2.2.3 Applications

There are several possible applications of handwriting recognition systems. Here, some of the most important uses of this technology are listed and described, even though several other applications are also possible.

Process Automation

Automatic address reading for postal mail sorting is a typical example of process automation. Off-line systems capable of recognizing isolated digits have already been created and installed in many post offices around the world, as part of automatic mail-sorting machines. A further enhancement of the same would be a system capable of reading the whole address. Reliable automatic recognition of handwritten words would replace the time consuming manual arrangement of letters. Automatic recognition of machine printed addresses has been successfully implemented. The recent advances in handwriting recognition are quite promising.

Banking Automation

An important commercial application for off-line cursive script recognition is the machine reading of bank cheques. Like in mail sorting machines, throughput requirements can be very strict. However, the solution of the recognition problem is facilitated by some constraints, e.g. limited character set or containing only isolated character. Given the number of checks passing through the banking system each day, such a system, even if only able to confidently verify half of the checks, would save much labor on a tedious and unpleasant job. Furthermore, security could be improved since signature verification system is nowadays under much investigation and could also be implemented.

Research on HMM-MLP hybrid system for segmenting and recognizing unconstrained handwritten dates written on Brazilian bank cheques had been carried out [27]. The system evolved by dealing with many sources of variability, such as heterogeneous data types and styles, variations present in the date field, and difficult cases of segmentation that make the recognizer task particular hard to do.

Signature Verification

The verification of signatures is a convenient and reliable method to check whether a specific person is authorized to conduct a certain transaction, to

access data, etc. The advantage of signatures over personal identification numbers (PIN) or passwords is that they are legally binding. Velocity and pressure are two important features which are measured by today's verification systems to uniquely characterize the signature of a person. Automatic signature verification has not yet achieved the required level of robustness. However it has an enormous potential for new applications.

Office automation

The main goal of office automation is to make typical process in an office more efficient. In almost every office the conventional media for information processing is text printed or written paper. Typical processes are retrieving and sorting. In order to automatize and speed up these processes with the help of computers, it is necessary to convert text data from its pictorial form to a representation that can be processed by a computer.

One such example is the use of Intelligent Character Recognition (ICR) for the examination processing system. Manipur University is currently using an ICR system for the examination processing of graduate courses. The ICR software system has been developed by German based company and has been supplied by the Exxon automation Pvt. Ltd. Mumbai.

Personal Digital Assistants (PDAs) and Mobile Phones

Personal Digital Assistants are small hand-held computers which combine agenda, address book and telecommunication facilities. There are a lot of applications for this machine. It can be used as a personal organizer, a notebook, a data acquisition device, inventory surveys, etc. PDA offers an interesting alternative to the use of paper because they avoid the drawbacks of paper but can store data in a format that can be processed by a computer. Entering data into a small hand-held PDA via small keyboard can be very inconvenient. It is far more easy to use handwriting to enter data into such a small device. Various PDA already use this kind of technology but still the user most of the times has to adapt to the machine than the other way around.

The advanced features of Mobile phones which are currently available have many applications. It has features such as Handwriting recognition with stylus pen for inputting data.

2.2.4 Document Image Analysis

The use of computers in the creation of documents has been of great benefit to society for the past few decades. Software tools such as word processors, computer aided design (CAD) packages, drawing programs, and mark-up languages assist us in the creation of these documents and allow for their storage in a format understood by a computer. In this format, a document can be easily edited and high quality hard copies can be created using a printer, or it may be quickly distributed electronically to others across world-wide networks.

Additionally, we may want to take advantage of other facilities available to us when the document is in a computer readable or understandable format. Some of these include keyword or pattern searching of what may be very lengthy documents, applying optimization algorithms or simulations on things such as electronic circuit designs or improving the visual quality of the pages of a book or a photograph by removing noise that could be the result of years of decay. Document analysis can be broken into a number of steps as shown in Figure 2.12.

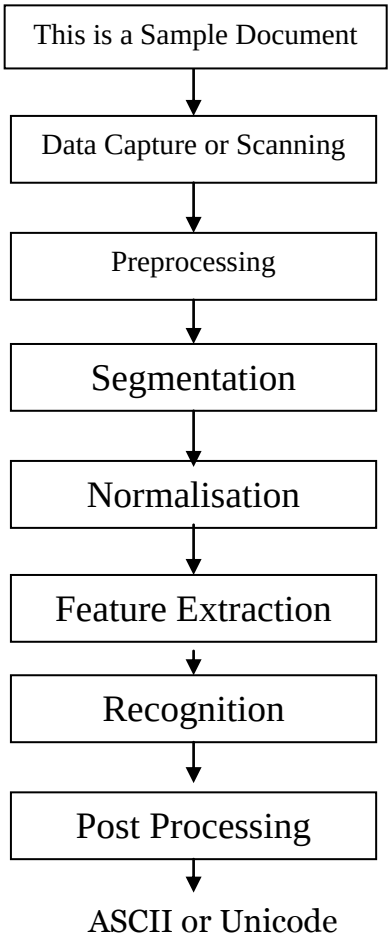


Figure 2.12: Major steps in Document Image Analysis

However, as the present form of the document is on paper, all these benefits are not possible. Although, we are now in the age of desktop publishing and most recently printed journals and books are originally produced in an electronic format, this is still not the case for the trillions of old documents, nor the handwritten notes, forms, or drawings, that are still in use by all of us even today. The information contained within these documents must first be extracted from the hard copy and stored in a computerized format if we wish to have the benefits described above. This is the motivation of document analysis. Optical Character Recognition (OCR) is that subfield of Document Analysis which is mainly concerned with the recognition of machine printed or handwritten words or characters (HCR) in a document.

In the case of handprint it is referred to as intelligent character recognition (ICR). The state of the art in the field of OCR has reached the point of practical use in recent years.

While production systems now exist that are capable of recognizing characters that can be reliably segmented, other more difficult to segment forms of writing remain an open problem. Following the trends in the development of speech recognition, much of recent research has turned away from attempting to presegment cursive characters, and instead has concentrated on segmentation free approaches.

2.2.5 Numerical/Character Recognition

Intensive research on the recognition of isolated digits in the past decade has led to recognition rates close to 99% (zero-rejection level). Many experiments have been conducted on the CENPARMI (Centre for Pattern Recognition and Machine Intelligence), CEDAR (Center of Excellence for Document Analysis and Recognition), and NIST (National Institute for Standards and Technology) databases, which are well-known databases used by researchers in this domain.

Different techniques are available in the literature that has been employed for recognizing handwritten numerals/characters. A system to recognize handwritten numerals/characters requires a number of standard components. [24,25,26] outline the typical components of such a system: (a) Data capture/Scanning/Digitisation, (b) Location and Segmentation (c)

Preprocessing, (d) Feature Extraction and (e) Recognition/Classification. Parts (a) and (b) refer to the acquisition of the numeral/character image. In this section, only the recognition process for isolated numerals/characters is examined (isolated characters are those that have already been separated from surrounding characters and words).

Data capture/Scanning/digitization in this case consists normally in a digitized image of the analog document using an optical scanner. A page of a document is first captured as an image by using a scanning device and stored as a collection of pixels in the form of a matrix, often with as many as 300 pixels per inch of the original document. The resulting image may be in black and white in which case the pixels will be stored as a single byte, 256 level grayscale in which they are stored as one byte where the value zero indicating a white pixel, whereas a black pixel is represented by the value 255, or color in which pixels may be stored using three components, red, green, and blue, with each of these components possibly stored as a single byte for a total distinction of over 16 million colors.

Preprocessing of the digitized image will be performed and after the regions containing the text are located, each symbol is extracted through a segmentation process. The extracted symbol is followed in most of the cases by a preprocessing, so to eliminate noise and to facilitate the extraction of features in the next step. Feature extraction is one of the most important step in the recognition system, since the feature select has to represent well the pattern in which one wishes to classify. In many case, the amount of data selected in the feature extraction is huge so a reduction of this data is necessary. The classification procedure is the core of the recognition system.

2.2.5.1 Preprocessing

Preprocessing of an image prepares the image for subsequent recognition by removing some of the irrelevant elements of the image while retaining the relevant information. Common forms of preprocessing include thresholding, noise filtering, smoothing, skew detection and correction, line segmentation, word segmentation, isolation of individual characters, typically those that are written discretely rather than cursively.

Various shades of gray are represented between these two values of 0 and 255 for a gray level image. Many researchers have decided to convert the initial gray-level images into a less storage intensive format i.e. a binary (0 and 1), black and white format. It is argued whether recognition performed on features directly extracted from gray-scale or from binary images produces the better result. The fact remains that many researchers have used to perform recognition on binary characters for simplicity and speed [27],[28].

The process of converting a gray-level image to a binary image is called thresholding or binarization. The task of thresholding is to extract the foreground (ink) from the background (paper). This is the operation of selecting which elements of a character image may be considered the background (white pixels) and which elements are to be considered the character itself (black pixels). Some threshold is usually used so that pixels with a luminance over the threshold are marked as being background pixels while pixels with a luminance under the threshold are considered to be part of the character image. Selecting an appropriate threshold has been the subject of active research for a number of years.

Broad categories of thresholding algorithms are: (1) Global, (2) Locally adaptive methods [29][157], (3) Hybrid. In the former, a single threshold is calculated for the entire image. Global thresholding picks one threshold value for the entire image, often based on an estimation of the background level from the intensity histogram of the image. Adaptive thresholding is a method used for images in which different regions of the image may require different threshold value. One way of accomplishing this is by computing a local threshold for each pixel based on a window consisting of its surrounding pixels.

A number of examples for calculating the threshold such as taking into account the cumulative gray-scale count of the entire image or by determining the minimum of the image's bi-modal histogram are available in the literature. A common method practiced is to use a histogram of the pixel values in the image, there should be a large peak indicating the general value of the background pixels and another smaller peak indicating the value of the

foreground pixels. A threshold can then be chosen in between the two peaks, this is known as valley-seeking. This method is successful; however images do not always contain well-differentiated foreground and background intensities due to poor contrast and noise. Perfect thresholding is a difficult task. An iterative thresholding presented by Ridler and Calvard is a switching mechanism which decides always between background and object, where the corners of the image are taken as initial values for the background [36]. Gonzalez and Woods describe similar iterative procedure where the midpoint between the minimum and maximum intensity values in the image is taken as initial estimate [37]. Another very popular thresholding algorithm is the Otsu's algorithm [38]. Otsu's algorithm, regards the histogram as probability values and defines the optimal threshold value as one that maximizes the between-class variance, where the distributions of the foreground and background points are regarded as two classes. Each value of the threshold is tried and one that maximizes the criterion is chosen. There are several improvements to this basic idea, such as handling textured backgrounds similar to those encountered on bank checks. Such method measures attributes of the resulting foreground objects to conform to standard document types. Solihin and Leedham [39], [40] describe a global multistage thresholding algorithm which is one that performs thresholding in n stages (n>1). In each stage k, it uses the threshold value(s) produced by stage k-1 and additional information from the image, to produce more accurate threshold value(s) for stage k+1. This process continues and ends at stage n where the one value produced by this stage is the final threshold value. One example of multistage thresholding technique is the quadratic integral ratio (QIR) technique. It was found superior in thresholding handwriting images, especially under tight requirements where the details of the handwriting are to be retained, the papers used may contain strong coloured or patterned background and the handwriting may be written by a wide variety of writing media such as a fountain pen, ball-point pen, or pencil. It was reported that the performance of automatic QIR technique with a fixed value of the threshold for ball-point pen is better than that of Otsu's. Pal and Pal [41] reviewed various methods for gray level image

segmentation. Thresholding is undoubtedly one of the most popular segmentation approaches because of its simplicity. However, the automatic selection of a robust, optimum threshold has remained a challenge in image segmentation. Pal and Pal [42] developed another entropy-based method by considering the joint probability distribution of the neighboring pixels which they further modified [43] with a new definition of entropy. An optimum image thresholding method introduced by P.K. Saha and J.K. Udupa [44] reported that the method accounts for both intensity-based class uncertainty-a histogram-based property-and region homogeneity-an image morphology-based property. Sahoo et al. [45] evaluated more than twenty global thresholding methods via uniformity and shape measures. They concluded that Otsu's class separability method [38], Tsai's moment preserving method [46] and Kapur et al.'s entropy method [47] are satisfactory. Lee and Chung [48] compared five global thresholding methods employing the error probability, shape uniformity measures as the criteria. They concluded that methods of Otsu and Kittler and Illingworth [49], [50] yielded relatively acceptable results. Otsu's method [38] assumed that the thresholded images have two normal distributions with similar variances. The threshold is selected by partitioning the image pixels into two classes at the gray level that maximizes the between-class scatter measurement. Kittler and Illingworth's methods assumed that the thresholded images have two normal distributions with distinct variances. There are many shortcomings for global methods such as (1) When images have highly unequal population sizes, global methods tend to split the larger mode into two halves [49],[50]. (2) If the gray levels of foreground and background are inseparable, global methods cannot find acceptable thresholds.(3) Global methods cannot handle images with gradually decaying background or with texture. In short, global methods neglect the spatial relationships among pixels. Chun-Ming Tsai and Hsai-jian Lee [51] presented binarization algorithm for color document images by solving the first two problems using a decision tree based binarization method, which selects various color features to binarize color document images. Because of the advancement of color printing technology, color document images are employed increasingly. Documents

can be digitized and then recognized via optical character recognition (OCR) techniques. For most OCR engines, character features are extracted and trained from binary character images. It is relatively difficult to obtain satisfactory binarized images from various kinds of document images. Locally adaptive techniques calculate the threshold of each pixel in an image based on the information contained within its neighbourhood [32]. Trier and Jain [32] [33], evaluated 11 popular local methods for map images. Their experimental results demonstrated that when using the OCR engine of Trier and Jain, the methods of Niblack [52] and Bernsen [53] produced better and faster OCR results. Based upon local means and standard deviations, Niblack's method defined varying threshold values upon the image. Bernsen's method detected the lowest and the highest gray level, G_{low} and G_{high} , in a w x w square window centered at (x,y). The threshold of pixel (x,y) is defined as $T(x,y) = (G_{low} + G_{high})/2$. In summary, local methods often yield better experimental results, but are slower than global methods. Locally adaptive techniques outperformed the global techniques on map images that were low in contrast and had variable background intensity and noise. It might be noted that in the above-mentioned study, the locally adaptive method with the best performance also took 10 times longer to execute than the best global method (3 seconds and 0.3 seconds respectively). Another performance evaluation by Abak *et al.* [34] confirms this by mentioning that three of the locally adaptive methods tested were amongst the slowest algorithms for document binarisation. They also mentioned that based on such criteria as background contamination and chipping away from the foreground, certain global methods ranked quite well against some of the locally adaptive methods. In their evaluation, Trier and Jain found that no one method outperformed the other methods on the entire map images tested. However, they also concluded that given the images, some of the best locally adaptive techniques still did not give high enough binarisation accuracy for use in further automatic processing. This is an observation that may explain why certain researchers still prefer to extract features directly from grey-level images.

Hybrid thresholding methods combine both global and local methods. Liu and Li [54] proposed a 2-D Otsu thresholding method, which performs better than the 1-D Otsu method does, when images are corrupted by noise. Their method calculates the local average gray level within a limited window. They constructed a 2-D histogram, in which the x-axis and the y-axis are the gray value and the local average gray level, respectively. The optimal threshold vector(s,t) is selected at the maximum between-class variance. Gong et al.[55] proposed a fast 2-D Otsu method to accelerate Liu and Li's method. Via this method, the computation complexity can be decreased from $O(G^4)$ to $O(G^2)$, G is the number of gray levels. Both the computation time and the memory space are reduced greatly, while the segmentation quality is maintained. In addition, Tseng and Lee [56] proposed a document image binarisation method, which is a two-layer block extraction method. In the first layer, dark backgrounds are extracted by locating connected-components. In the second layer, background intensities are determined and removed from each component. In comparison, hybrid methods have better experimental results, but are also slower than global methods.

The next step in processing that shall be discussed relates to the *normalization* of character images. Hand-printed characters may vary in the following ways: (1) Size (height and width), (2) Slant (and rotation), (3) Line Thickness, (4) Style (such as open and closed-top 4's, looped and non-looped 2's, etc.), (5) Ornamentation, (6) Stroke Proportions (relative widths of strokes that constitute the character) and (7) Stroke Regularity (smoothness and well-formedness of lines and curves). It is mentioned that the above list is non-exhaustive. As can be seen, the variability of handwritten patterns poses a very big problem for machine recognition of characters. It is for this precise reason that researchers have developed *Normalization* techniques that form part of the preprocessing phase to be used prior to feature extraction and classification.

Normalization refers to such operations as: the estimation and correction of a character's slant, scaling the character to a *uniform size* and also possibly reducing the character to a skeleton so that the line width is a uniform, one pixel wide. In the case of certain classifiers (such as Artificial

Neural Networks), a uniform feature vector is required for training. In such cases it may be necessary to scale all characters up or down to obtain character matrices of uniform size. Researchers have used various types of size normalization techniques [59]. It is necessary to modify the Cartesian coordinates of the input image by multiplying each via scaling constants. To determine how the coordinates shall be filled in the destination image, reverse mapping is performed. It is the process whereby the destination image is traversed and calculations via an inverse transformation decide which pixels in the source image shall be used to produce the destination pixel. In some cases the mapping function may calculate a fractional pixel address. To solve this problem when mapping, a technique called interpolation may be used. Interpolation estimates the new pixel in the destination image by obtaining a value from some function of the neighbors of the corresponding source address. Many interpolation methods are available such as nearest neighbor, bi-linear interpolation, *bi-cubic interpolation* etc.

The next type of normalization that shall be examined is that of slant estimation and correction or skew detection which is an additional form of preprocessing required for text documents. Due to inaccuracies in the scanning process, the document may be slightly tilted within the image. This can hurt the effectiveness of later algorithms and therefore should be corrected. Methods of detecting the amount of skew include using the projection profile of the image, using the Hough transform, Fourier transform and a form of nearest-neighbor clustering of connected components. The projection profile method creates a profile histogram by projecting lines at angles around and including the horizontal direction. The angle for which the histogram shows the largest maximum variation between peaks and valleys is taken as the correct angle.

The Hough transform is a method of line detection that maps(x,y) space to an accumulator space(r,θ), where θ is the angle of a line found in the image and r is the distance of the line from the origin. This can then be used to find the dominant angle for which the accumulator has high counts, and we can set this as the skew angle. Proper tuning of the resolution of r must be

done so as to compute the skew from objects of the desired size (e.g., from columns of characters), thus forcing the size of these objects to be known ahead of time. Techniques using run-length encoding have been proposed to overcome this problem.

Nearest-neighbor methods involve segmenting the image into components and computing the nearest neighbors for the center of gravity of each of these components. This has the effect of connecting characters within words since they are closer together than cross-word character distances. An angle histogram of the direction vectors connecting these characters can be created and the peak of the histogram taken as the skew angle.

In some character recognition systems, the slant correction phase is not required as spurious slants are detected in the feature extraction phase [59]. Other researchers have chosen to incorporate slant estimation and correction as a discrete step into the overall process of character recognition [57]. It is emphasized that preprocessing is important for removing *noise and irregularities* that may have been introduced when the numeral images were scanned from their source [57]. Some anomalies are identified as a result of the acquisition process i.e. touching line segments and smeared images.

An important function of the preprocessor is to remove isolated pixels and bumps while also filling holes. This process is referred as smoothing. The smoothing process is not limited to removing irregularities in the numeral's contour.

To remove small objects from binary image of the scanned page of isolated characters or digits, the basic steps of the algorithms is as follows: (1) Determine the connected components (2) Compute the area of each component (3) Remove small objects. The small objects has the user specified maximum number of pixels in objects, specified as a nonnegative, integer-valued, numeric scalar. The default connectivity is 8 for two dimensions and 26 for three dimensions. The algorithm
removes all connected components (objects) that have fewer than P pixels from the binary image BW, producing another binary image, BW2.[182]

To reduce noise in document processing, the *k*Fill filter has been extensively used [158]. This is a more general filter designed to reduce isolated noise and noise on contours upto a selected limit in size. The size adjustment parameter is the *k* of *k*Fill.

2.2.5.2 Segmentation

The image is segmented into separate components, such as graphics, pictures, and text, and each segment can be further processed using different techniques tuned to the type of data contained in that component. At a finer level, the field of character recognition further addresses the problem of word-level segmentation and character level segmentation. This section discusses some of the issues of segmentation, first at a higher level, which will be referred to as region segmentation and then the finer levels are the word segmentation and character segmentation.

Region segmentation can be categorized as page layout analysis and forms processing. The categorization is based on the amount of prior knowledge that exists about the structure of the document. Page layout analysis is concerned with the segmentation of different regions of a page of text such as paragraphs, titles, rows, etc., with little or no prior knowledge of the layout and contents of the document and relies on top-down and bottom–up approaches to reconstruct the structure of the image before individually processing these components. The field of forms processing works with documents for which layout information is available ahead of time and the job of defining individual components of the image is reduced to finding some registration points on which the structural definition can be aligned. In the *Page Layout Analysis*, two stages have been defined in the literature[71].The first is the *structure analysis* which is concerned with the segmentation of the image into blocks of document components such as paragraphs, rows, words, characters, etc.. The second is the *functional analysis* which uses location, size, and other layout rules to label the functional content of document components such as title, abstract, etc. Structural analysis may be performed in one of the following ways: a top-down approach in which the image is broken into smaller and smaller

subgroups, a bottom-up approach in which first the smallest components are segmented and then grouped into larger and larger components, or some combination of the two approaches. The top-down approach is often accomplished by a run-length smoothing algorithm which smears together components that are within a predefined distance of each other [72]. This distance is set depending on the size of the components contained in the regional blocks for segmentation. Projection profiles can then be used to find and segment the blocks containing these components. Many bottom-up approaches also use this run-length smoothing-projection profile technique to divide the image into regions in which c*onnected-component analysis* can be performed. Larger and larger document components are grouped by using statistics on these connected-components.

2.2.5.3 Word and Character Level Segmentation

In most texts, large amount of space are found separating the words and segmentation of words can usually be performed by looking for valleys in the vertical projection profile. Connected component analysis is another method where word boundaries can be defined based on the distance between two connected components. A survey of character segmentation techniques is reviewed by Casey et al. [29]. The difficulty in character segmentation ranges from that of machine printed discrete characters to the more difficult problem of recognizing words and symbols that have been handwritten by a human.

Five different types of handwritten word defined in the literature are (1) Boxed discrete characters (2) Spaced discrete characters (3) Run-on discretely written characters (4) Pure cursive script writing (5) Mixed cursive, discrete, and run-on discrete.

In the boxed discrete characters, it requires the writer to place each character within its own box on a form. The boxes themselves can be easily found and dropped out of the image, or can be printed on the form in a special color ink that will not be picked up during scanning, thus eliminating the segmentation problem entirely. Spaced discrete characters can be segmented reliably by means of horizontal projections, creating a histogram of grey values in the image over all the columns, and picking the valleys of

the histogram as the points of segmentation. This has the same level of segmentation difficulty as is usually found with clean machine printed characters. Characters of run-on discretely written characters and (1) and (2) writing types are usually discretely written, however in (3) they may be touching, therefore making the points of segmentation less obvious. This level of difficulty may be found in the degraded machine printed characters. These require more sophisticated methods to find points where characters are connected and break them apart. Even more difficult are the problems of cursive handwriting and mixed cursive and discrete writing. Incorrect segmentation such as splitting a character or grouping two characters leads to incorrect classification of the characters during the recognition phase. Post processing may attempt to remedy this, but the system has the weakness during segmentation. Research results on concurrent segmentation and recognition are found in the literature. Character segmentation is defined as: "...an operation that seeks to decompose an image of a sequence of characters into sub-images of individual symbols." [29]. An important point that must be understood is that before characters in a word may be segmented, it is necessary to first ascertain what a character is. It seems that the character segmentation process requires that the properties of a character be known; this information may be obtained through recognition. Unfortunately, to obtain knowledge of a character's appearance, segmentation is required. This means that one stage seems to be dependent on the other, whereby knowledge of character symbol structure in a word may be required to aid in segmentation. Casey and Lecolinet go further in saying that segmentation points should not solely be created on the basis of local information. Rather, previous and future segmentation decisions should also be made on the basis of contextual information. In other words if segmentation is made between two primitives in a word and the resulting letters do not fit with any letter permutations in a word lexicon, such segmentation may be deemed incorrect and may influence previous and future segmentations. Therefore Casey and Lecolinet summarize the last forty years of character segmentation research as being dependent on local topological information as well as global contextual information. It is with

cursive handwriting in particular that the idea of *segmenting during recognition* has become popular. This is often accomplished by use of a sliding window. A window of some predefined size is slid from left to right across the image over (pseudo) time using some predefined step size. At any point in time, recognition is performed on the contents of the window. In this way, a dynamic signal is simulated from a static image.

In the case of touching numeral segmentation and word segmentation, the sub-sections are (1) *Dissection Techniques:* These refer to techniques for segmentation that are based on the concept of segmenting numeral images into sub-components utilizing general features. No classification, contextual knowledge or character shape discrimination steps are present in these segmentation techniques. The dissection itself describes the entire technique. (2) *Recognition-based Segmentation:* These techniques do not employ specific dissection strategies. Rather the image is partitioned into overlapping sections and a classifier is used to perform "segmentation" by verifying whether a particular section consists of a character. Casey and Lecolinet add that this type of technique is referred to as "recognition-based" because the character segmentation is a by-product of recognition. (3) *Hybrid Strategies: Over-segmentation:* These strategies tend to be a combination of the first two that were mentioned. Dissection is used to over-segment the word or connected numeral component into a sufficient number of components as to encompass all segmentation boundaries present. In the next step, classification is employed to determine the optimum segmentations from a set of possible segmentation hypotheses. (4) *Holistic Strategies:* Finally, these strategies intend to recognize words as entire units rather than attempting to extract individual characters.

2.2.5.4 Hand-printed

Segmentation techniques for hand-printed words as opposed to handwritten cursive words will be discussed in this section. In the former, the characters are not explicitly joined via "ligatures". Ligatures are mainly present in cursive words, they are defined as being: the line segments that connect one character to another. Although ligatures are absent from hand-printed text, other anomalies may be present such as slanted characters

which results in overlapping. Another problem includes characters that are very close to each other and hence have touching components.

The technique proposed by Leedham and Friday [70] first separated the isolated characters in a word image by locating the connected components. Then to further split the connected characters, instead of using a vertical projection, the authors analyzed the projections at angles between -16^{o} and +16^{o} at 2^{o} increments. The technique was able to detect near vertical lines and gradual curves in the presence of slanted characters. A projection chosen at the correct angle would display a sharp rise from left to right. This is identified by a maximum in the derivative of the projection. The dissection then took place along the angle of appropriate projection. Other rules were implemented to remove segmentation boundaries that split more than one connection point, and merged small components that had been removed from a character's body.

Cesar describes a segmentation system of Canadian handwritten postal codes [71]. This research involved the location of connected components followed by bounding box analysis. Connected component location involves searching the input image for connected foreground (black) regions. Upon location of these regions it is possible to decide whether the connected components should be merged or split. Bounding box analysis is often used following the location of connected components. The "bounding box" simply refers to the location and dimensions of a connected component. Bounding box analysis can be very helpful; it provides information such as the proximity of connected components. This information may be used to decide whether two adjacent connected components should be joined. The size of the bounding boxes may be used to decide whether connected components should be split. In Cesar's research, connected components were found by performing a contour-tracing algorithm. While being executed, the process recorded the location of the upper-most and bottom-most rows as well as the location of the left-most and right-most columns. These values provided the bounding box locations for each connected component. The connected components were then split or merged based on rules pertaining to the height and width of the bounding boxes. Prior to

splitting and merging, any small fragments were eliminated if their size (given by bounding box information) was smaller than some threshold.

2.2.5.5 Feature extraction- An overview

Feature extraction can be defined as the process of extracting distinctive information from the matrices of digitized characters. Feature extraction is a phase of the recognition process in which the objects are measured. A measurement is the value of some quantifiable property of an object.

A feature is a function of one or more measurements, computed so that it quantifies some significant characteristic of the object. This process produces a set of features that, taken together, comprise the feature vector. Features are the fundamental component of characters. To find out a group of the most effective features for classification is the basic task of feature extraction and selection. It means that compressing from high-dimensional feature space to low-dimensional feature space so as to design classifier effectively [87].

Given an input set of features, two different ways for achieving dimensionality reduction are: (1) The first is to select the hopefully best subset of features of the input feature set. This process is termed feature selection (FS). (2)The second approach that creates new features based on transformation from the original features to a lower dimensional space is termed feature extraction (FE). This transformation may be a linear or nonlinear combination of the original features. The choice between FS and FE depends on the application domain and the specific training data which are available. Since some of the features are discarded, FS leads to savings in measurements cost and the selected features retain their original physical interpretation. In addition, the retained features may be important for understanding the physical process that generates the patterns. On the other hand, transformed features generated by FE may provide a better discriminative ability than the best subset of given features, but these new features may not have a clear physical meaning [180].

Due to the fact that the size, the location, and the orientation of an object may change when presented in various pictures, features which are invariant to scaling, translation, and rotation are the most desirable for object recognition. In this research, we take the shape information of the object as the fundamental feature since it possesses the desired invariant characteristics. SIFT descriptor has been also used in some recent research for off-line digit recognition [160].

The main benefits of feature selection are follows: (i) reducing the measurement cost and storage requirements, (ii) coping with the degradation of the classification performance due to the finiteness of training sample sets, (iii) reducing training and utilization time and, (iv) facilitating data visualization and data understanding. Generally, features are characterized as: (i) Relevant: features which have an influence on the output and their role cannot be assumed by the rest, (ii) Irrelevant: features not having any influence on the output, (iii) Redundant: a feature can take the role of another.

Advantages of filter methods are that they are fast and easy to interpret. The characteristics of filter methods are as follows: (i) Features are considered independently, (ii) Redundant features may be included, (iii) Some features which as a group have strong discriminatory power but are weak as individual features will be ignored, and (iv) The filtering procedure is independent of the classifying method.

The characteristics of wrapper methods are listed below: (i) Computationally expensive for each feature subset considered, since the classifier is built and evaluated,(ii)As exhaustive searching is impossible, only greedy search is applied. The advantage of greedy search is simple and quickly to find solutions, but its disadvantage is not optimal, and susceptible to false starts, (iii) it is often easy to overfit in these methods.

There are various ways to generate features from the raw data set. A number of transformations can be used to generate features. The basic idea is to transform a given set of measurements to a new set of features. Transformation of features can lead to a strong reduction of information as compared with the original input data. So for most of the classification a

relative small number of features are sufficient for correct recognition. Obviously feature reduction is a sensitive procedure since if the reduction is done incorrectly the whole recognition system may fail or not present the desired results [87]. Examples of such transformations are the Fourier transform, the Karhunen-Loeve transform, and the Haar transform. However feature generation via a linear transformation technique is just one of the many possibilities. There is a number of alternatives which are very much application dependent. Examples of such features are moment-based features, chain codes, and parametric models. It is generally agreed that feature extraction plays an important role in the successful recognition of machine-printed and handwritten characters.

In OCR applications it is important to extract those features that will enable the system to discriminate between all the character classes that exist. Many different types of features have been identified in the literature that may be used for numeral and character recognition. Two main categories of features are: *Global* and *Structural*.

The features that are extracted from every point of a character matrix are known as Global features. Originally some of the global techniques were designed to recognize machine-printed characters. Global features may be more easily detected and are not as sensitive to local noise or distortions as are topological features. However, in some cases small amounts of noise may have an effect on the actual alignment of the character matrix, hence displacing features. This may have serious repercussions for the recognition of characters affected by these distortions. Global features themselves may be further divided into a number of categories. The first and most simple feature is the state of all the points in a character matrix. In a binary image there are only black or white pixels, the state therefore refers to whether a pixel is black or white.

This simple "feature extraction" is applied to Template Matching, which is a technique that measures similarities in a given character pattern with the repository of stored patterns or "Templates"[12]. The objective is to calculate the distance between the input and the templates. The best match is decided based on a minimum-distance criterion. Variations on template

matching exist where additional logical rules are used to complement the matching process to identify particular characters. The use of all points in a character matrix produces problems in dimensionality. Namely, the dimensions of a feature vector to store all points of a binary character matrix are far too large. To deal with this problem, one strategy that has been adapted involves the extraction of features based on the *statistical distribution of points* [73]. Five methods that have been employed in the literature, based on the distribution of points, are briefly outlined here. (1) Moments: A number of methods in this category utilize the moments of pixels in an image as features. Tucker and Evans [74] calculated raw moments that were a function of the coordinates of each point in the image. Another type: central moments, are calculated by taking into account the distance of points from the centroid (centre of gravity) of the character [73]. In this instance, central moments are preferred to raw moments as they produce higher recognition rates and are invariant to the translation of the image. (2) Zoning: This method divides the character matrix into small windows or zones. The densities of points in each window are calculated and used as features to the chosen classifier [74]. (3) n-tuples: This method simply uses as features the occurrence of black or white pixels in a character image [76]. (4) Characteristic Loci: In this method, vertical and horizontal vectors are generated for each white background pixel in an image. Features are generated by counting the number of times a line segment is crossed in the vertical and horizontal direction [73]. (5) Crossings and Distances: Lastly, researchers have obtained features by analyzing the number of times the character image is crossed by vectors in certain directions or angles i.e. 0^o, 45^o, 90^o etc. [73].

Researchers have explored ways in which to decrease the size of feature vectors whilst also rendering the features immune to rotation and translation. This leads to the discussion of the third global feature category: *Transformations and Series expansions*. Transformations and Series Expansions have proven to be invariant to scaling, rotation and translation, whilst at the same time reducing dimensionality of the feature vector. Exhaustive list of transformations and series expansions that have been

employed for the task of character recognition are presented in [30], [73] and [78]. Some of the more noted types are the Fourier [79]-[84], Karhunen-Loeve [85] series expansions and the Hough Transform [86]. Walsh, Haar and Hadamard series expansions have also been explored for feature extraction [73], [78]. Granlund [79] along with Persoon and Fu [82] were among the first to introduce these Fourier descriptors. Granlund developed contour descriptors that were calculated by first denoting points on the contour of a character using complex numbers [87]. The contour was then expressed as a Fourier series of complex coefficients. Features independent of scale and rotation were then extracted from the ordinary coefficients. Many improvements for extracting meaningful features from Fourier coefficients have been described in the literature [83], [84].

Some researchers have combined Fourier descriptors with other topological features for the task of character recognition to improve performance [81]. Other researchers have used Fourier descriptors to represent the skeleton of characters rather than the boundary [87]. Although Fourier Descriptors have many advantages, they also have a number of drawbacks. One major disadvantage has to do with the detection of small spurs on the boundaries of characters. As an example, it is difficult to distinguish an O from a Q [30]. However, it must also be remembered that this supposed drawback might also be considered an advantage for filtering noise on the boundary.

Geometrical and topological features are also among the most popular features [73], [78]. In this category, researchers have extracted features based on the geometry, topology and structure of the character image. Such features may represent global and local characteristics of the character. This includes: strokes and bays in various directions, end points, intersections of line segments, loops, stroke relations, angular properties and sharp protrusions [78]. Feature extraction techniques by Trier *et al.* [87] mention features that are extracted from the contour of a character and from its vector representation. Many techniques based for the extraction of contour features exist [88][89][90]. One method suggested by Kimura and Shridhar [91] used contour profiles. The contour is divided in two (left and right), and

each half of the contour is approximated by a discrete function. The features may then be extracted from these functions. Trier *et al.* mention that it is possible to use vertical or horizontal profiles from the inner or outer profiles. They also add that it is possible to use the profiles themselves as features, as well as the width between values located at the right and left of the profiles. Other features include: the ratio of the vertical height of the character by the maximum of the width function, location of maxima and minima in the profiles and many more. Kimura and Shridhar [91] utilised zoning on the contour curves. In each zone they calculated a local histogram of the chain codes of the normalised contour. The feature vector for each character had 64 components, totalling 16 zones. Therefore in each zone a tally of the 4 possible directions (0^o "-", 45^o "/", 90^o "|", or 135^o "\") was kept.

Srikantan, Lam, Srihari [88] used a Sobel operator to calculate the contour gradients of localised contour variations. The authors define the gradient as the magnitude and direction of the greatest change in intensity in a small neighbourhood of each pixel in the image. Another example is from that of Takahashi [89] who used orientation histograms of zones in a character image by means of vertical, horizontal and diagonal slices. These orientations were extracted from the inner and outer contours of the character. Cai and Liu [90] extracted chain code features from the outer contours of handwritten numerals. They extracted location, orientation and curvature information from the contour. The features were extracted in sequence and could therefore be applied to a Hidden Markov Model-based approach for recognition. High accuracy recognition was recorded employing numerals from the CEDAR data set. It may be noted that the research presented in this section is a non-exhaustive list of contour-based feature extraction techniques; many others are described in the literature.

Finally, features extracted from vector representations of character contours are reported in the literature. In this type of feature extraction, the character image is thinned and a graph is extracted from the skeleton. Straight line segments, junction points and arcs may be approximated from this thinned representation [87]. Pavlidis [92] extracted approximate strokes from skeletons. Kahan *et al.* [93] added extra features to improve recognition

performance. Lam *et al.* [94] extracted line segments as well as convex polygons from the skeletons of preprocessed numerals. As with features extracted from the contour, the number of papers detailing the extraction of features from skeleton representations is quite great and shall not be covered further here. The keen reader may explore Suen's numeral recognition surveys for further examples [95], [96].

Oliveira et al.[123] proposed a specific concavity, contour-based feature sets for the recognition and verification of handwritten numeral strings. The OCR system could process either isolated digits or handwritten numeral strings. Britto et al.[97] combine foreground and background information extracted from columns and rows of character images to contemplate isolated digit and numeral string recognition. These complementary features are combined in a two-stage HMM-based method.

Summary of Combining feature extraction methods: (Hybrid Feature Extraction Method)

(1) Vamvakas et al. (2010) [161] --- Zoning based features/upper and lower character profile projection features/left and right character profile projection features/ distance based features.

(2) Chacko et al. (2011) [162] ----- Wavelet features/chain code features

(3) Wang & Sajjahar (2011) [163] --- Polar transformed images/ Zone based feature extraction.

(4) Yang et al. (2011)[164] --- Structural features/Statistical features

(5) Chel et al. (2011)[165] --- Transition Feature/ Sliding Window Amplitude Feature/Contour Feature

(6) Al-Khateeb et al.(2011) [166] --- Structural features/Statistical features

(7) Rajput & Horakeri (2011) [167] ---Boundary-based descriptors/namely/ crack codes /Fourier descriptors

(8) Sharma & Jhajj(2011) [168]---Zoning/ Directional Distance Distribution (DDD) /Gabor methods

(9) Choudhary et. al(2012) [169]---Vertical/ Horizontal/ Left Diagonal and Right Diagonal directions

(10) Li et al. (2012)[170]---Direction string / nearest neighbor matching

(11) Nemouchi et al.(2012) [171]---Structural(like strokes, concavities, end points, intersections of line segments, loops, stroke relations) /statistic (zoning, invariants moments, Fourier descriptors, Freeman chain code) features

(12) Ahmed et al.(2012) [172]--- Multi Zoning of the character array (i.e., dividing it into over- lapping or non-overlapping regions, computing the moments of the black pixels of the character, the n-tuples of black or white or joint occurrence, the characteristic loci, and crossing distances)

(13) Likforman-Sulem et al. (2012) [173] ---Structural and statistic features

(14) Kessentini et al. (2012) [174] ---Directional density / (black) pixel densities features

(15) Bhattacharya et al. (2012) [175] ---Chain code computation/ gradient feature / pixel count feature generation

(16) Reddy (2012)[176]---Vertical and horizontal projection profiles (VPP-HPP)/zonal discrete cosine transform (DCT)/chain-code histograms (CCH) / pixel level values

(17) Muhammad et al.(2012) [177]---Correlation based function features/ structural/statistical

(18) Vidya V et al. (2013) [178] ---Cross feature/ fuzzy depth/distance/ Zernike moment

(19) Primekumar et al. (2013) [179] ---Structural Feature/Directional

Generally, feature selection is finding a subset of features which improve the recognition accuracy. This process has two main phases. First phase includes a search strategy to select one feature subset among all possible, the second phase

includes a method for evaluating selected subsets with assigning a fitness value to them generally divided in two: filter methods and wrapper methods. (i) Filter methods, which evaluate a feature subset independently of the classifier and are usually based on some statistical measures of distance between the samples belonging to different classes. (ii) wrapper methods, which are based on the classification results achieved by a given classifier. Filter methods are usually faster than wrapper ones, as these latter require a new training of the used classifier at each evaluation. Moreover, filter–based evaluations are more general, as they exploit statistical information about the data, while wrapper methods are dependent on the classifier used.

The feature selection algorithm can be classified into two namely heuristic

and metaheuristic approaches. Many heuristic algorithms have been proposed in the literature for finding near–optimal solutions [76-77].GA is a one of metaheuristic approach and have been widely used to solve feature selection problems [82-87]

Summary of Feature Selection Approach:

Meta heuristic:

(1) Nasien D. et al.(2010) [183] --- Genetic Algorithm (GA),and Ant Colony, Optimization (ACO)

(2) Reza A. et al.(2010) [184] ------ Hybrid Genetic Algorithm, (GA) + Simulated Annealing (SA)

(3) Das N. et al. (2012) [185] -------Genetic Algorithm based Region Sampling

(4) Roy A. et al.(2012) [186] ------- Artificial Bee Colony (ABC)

(5) Li L. et al. (2012)[187] --------- Nearest Neighbor (NN)

(6) Nagasundara K.B. et al. (2012)[188] --- Multi cluster feature selection (MCFS)

(7) Stefano et al.(2014) [189] ---- Genetic Algorithm (GA)

(8) Roy A. et al.(2014) [190] ------ Axiomatic Fuzzy Set(AFS)

(9) Ghareh Mohammadi F.et al. (2014) [191] --- Artificial Bee Colony (ABC)

Heuristic:

(1) A. Marcano-Cedeno et al.(2010) [192] --- Sequential Forward Selection

(2) Nasien D. et al.(2011) [193] --------Randomized and Enumeration

based Algorithms

(3) Abandah G. et al.(2011) [194] --- Scatter criterion Symmetric uncertainty Fast correlation-based filter (FCBF) Minimal-redundancy-maximal-relevance (mRMR) Non-dominated sorting genetic algorithm (NSGA)

2.2.5.6 Classification

Multilayer Perceptron (MLP) [2]:

Multilayer perceptrons (MLPs) are artificial neural networks, learning models inspired by biology. As opposed to logistic regression, which is only a linear classifier on its own, the multilayer perceptron learning model, can also distinguish data that are not linearly separable.

An outlined of the architecture of an MLP is shown in Figure 2.13.

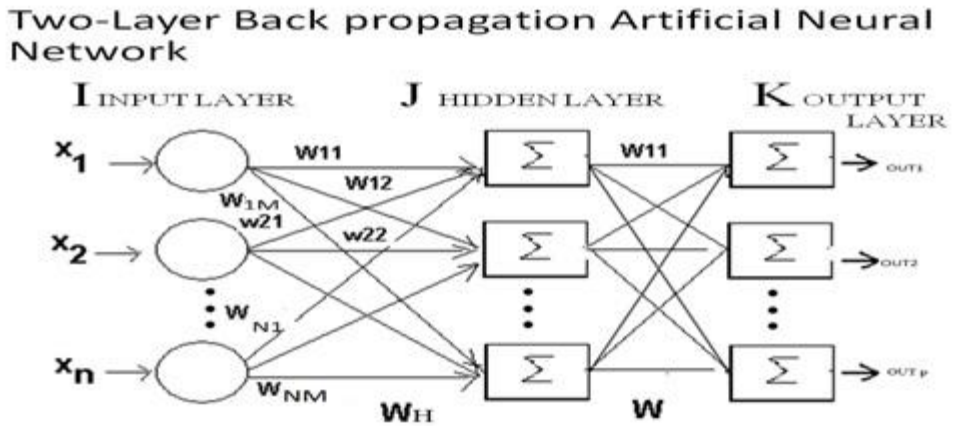


Figure 2.13 Basic view of the multilayer perceptron architecture. Layers consist of neurons; each layer is fully connected to the next one.

In order to calculate the class prediction, one must perform feed forward propagation. Input data are fed into the input layer and propagated further, passing through weighted connections into hidden layers, using an activation function. Hence, the node's activation (output value at the node) is a function of the weighted sum of the connected nodes at a previous layer. This process continues until the output layer is reached. First, the cost function is measured on the output layer, propagating back to the connections between the input and the first hidden layer afterwards,

updating unit weights. MLPs can perform multi-class classification as well, without any modifications. We simply set the output layer size to the number of classes we want to recognize. After the hypothesis is calculated, we pick the one with the maximum value. A nonlinear activation function is required for the network to be able to separate non- linearly separable data instances.

Learning: Backpropagation[2]:

A set of inputs is applied either from the outside or from a previous layer. Each of these is multiplied by a weight, and the products are summed. This summation of products is termed NET and for each neuron NET is calculated. After the Net is calculated, an activation function F is applied to modify it, thereby producing the signal OUT.

Sigmoidal activation function is given as

$OUT = F(NET) = 1/(1+e^{-NET})$

The derivative of F, $F' = OUT(1-OUT)$

Learning: Gradient descent backpropagation [2]

Forward Pass:

1. Select the next training pair from the training set;
   Apply the input vector to the network input
2. Calculate the output of the network
   O=F(XW) in vector notation

Backward Pass:

3. Calculate the error between the network output and the desired output (the target vector from the training pair).
4. Adjust the weights of the network in a way that minimizes the error.
5. Repeat steps 1 through 4 for each vector in the training set until the error for the entire set is acceptably low.

Adjusting weights of the output layer in the reverse pass is accomplished using modification of the delta rule; the following equations 2.1, 2.2 and 2.3, illustrate the weight adjustment from neuron p in hidden layer j to neuron q in the output layer k. The output of a neuron in layer k is subtracted from its target value to produce an error signal. This is multiplied by the derivative of the squashing function [OUT (1-OUT)] calculated for that layer's neuron k, thereby producing δ value.

$$\delta = OUT(1-OUT)(Target - OUT) \quad \text{---------- 2.1}$$

$$\Delta W_{p,q,k} = \eta\, \delta_{q,k}\, OUT_{p,j} \quad \text{---------- 2.2}$$

$$W_{p,q,k}(n+1) = W_{p,q,k}(n) + \Delta W_{p,q,k} \quad \text{---------- 2.3}$$

Adjusting weights of the hidden layers: Backpropogation trains the hidden layers by propagating the output error back through the network layer by layer adjusting weights at each layer. The value of δ needed for the hidden layer neuron is produced by summing all such products and multiplying by the derivative of the squashing function:

$$\delta_{p,j} = OUT_{p,j}(1-OUT_{p,j})(\sum(\delta_{q,k}\, W_{p,q,k})$$

Momentum method for improving the training time of backpropagation algorithm involves adding a term to the weight adjustment that is proportional to the amount of the previous weight change. The adjustment equations are modified to the following:

$$\Delta W_{p,q,k}(n+1) = \eta\,(\delta_{q,k}\, OUT_{p,j}) + \alpha[W_{p,q,k}(n)] \quad \text{----------2.4}$$

$$W_{p,q,k}(n+1) = W_{p,q,k}(n) + \Delta W_{p,q,k}(n+1) \quad \text{---------- 2.5}$$

Where α is the momentum coefficient, usually set to around 0.9.

Wasserman [2] describes an adaptive step size algorithm intended to adjust step size automatically as the training process proceeds.

Learning: Resilient Backpropagation [196]*:*

Resilient backpropagation (RPROP) is an efficient optimization algorithm proposed by Riedmiller and Braun in 1993 [196]. It is based on the

principle of gradient descent used with pure backpropagation. Instead of updating weights of a network with a fixed learning rate that is constant for all weight connections, it performs a direct adaptation of the weight step using only the sign of the partial derivative, not its magnitude. As such, it overcomes the difficulty of setting the right learning rate value.
For each weight, an individual weight step size is introduced: $\Delta_{i,j,}$ which solely determines the size of the weight-update.

This adaptive update-value evolves during the learning process based on its local sight on the error function E, according to the following learning-rule:

$$\Delta_{i,j}^{(t)} = \begin{cases} \eta^{+} * \Delta_{i,j}^{(t-1)}, & \text{if } \frac{\partial E^{(t-1)}}{\partial w_{i,j}} * \frac{\partial E^{(t)}}{\partial w_{i,j}} > 0 \\ \eta^{-} * \Delta_{i,j}^{(t-1)}, & \text{if } \frac{\partial E^{(t-1)}}{\partial w_{i,j}} * \frac{\partial E^{(t)}}{\partial w_{i,j}} < 0 \\ \Delta_{i,j}^{(t-1)}, & \text{else} \end{cases} \quad \text{------- 2.6}$$

Where $0 < \eta^{-} < 1 < \eta^{+}$

The adaptation-rule works as follows: Every time the partial derivative of the corresponding weight $w_{i,j}$ changes its sign, which indicates that the last update was too big and the algorithm has jumped over a local minimum, the update-value $\Delta_{i,j}$ is decreased by the factor η^{-}. If the derivative retains its sign, the update-value is slightly increased in order to accelerate convergence in shallow regions.
Once the update-value for each weight is adapted, the weight-update itself follows a very simple rule: if the derivative is positive (increasing error), the weight is decreased by its update-value, if the derivative is negative, the update-value is added:

$$\Delta w_{i,j}^{(t)} = \begin{cases} -\Delta_{i,j}^{(t)}, & \text{if } \frac{\partial E^{(t)}}{\partial w_{i,j}} > 0 \\ +\Delta_{i,j}^{(t)}, & \text{if } \frac{\partial E^{(t)}}{\partial w_{i,j}} < 0 \\ 0, & \text{else} \end{cases} \quad \text{---------- 2.7}$$

$$w_{i,j}^{(t+1)} = w_{i,j}^{(t)} + \Delta w_{i,j}^{(t)} \quad \text{----------2.8}$$

If the partial derivative changes sign, i.e. the previous step was too large and the minimum was missed, the previous weight-update is reverted:

$$\Delta w_{i,j}^{(t)} = -\Delta w_{i,j}^{(t-1)} \quad , \text{ if } \frac{\partial E^{(t-1)}}{\partial w i,j} * \frac{\partial E^{(t)}}{\partial w i,j} < 0 \quad \text{---------- 2.9}$$

Due to that 'backtracking' weight-step, the derivative is supposed to change its sign once again in the following step. In order to avoid a double punishment of the update-value, there should be no adaptation of the update-value in the succeeding step. In practice this can be done by setting $\Delta w_{i,j}^{(t-1)} = 0$ in the $\Delta w_{i,j}^{(t)}$ adaptation rule in equation 2.9. During the learning by epoch, the update-values and the weights are changed every time the whole pattern set has been presented once to the network. Zhang [116] reviews classification using ANNs.

G. Vamvakas et al. [200] proposed a methodology for off-line handwritten character/digit recognition. A new feature extraction technique based on recursive subdivisions of the image as well as on calculation of the centre of masses of each sub-image is presented. A hierarchical classification scheme based on the level of granularity of the feature extraction method is employed. Pairs of classes with high values in the confusion matrix are merged at a certain level and higher level granularity features are employed for distinguishing them. The hierarchical classification scheme was performed using SVM with radial Basis Function (RBF) kernel. The overall recognition rate is 93.21% for characters from CIL Database and is 98.66% for digits from MNIST Database.

Georgios Vamvakas et al.[161] proposed a new feature extraction technique based on recursive subdivisions of the character image so that the resulting sub-images at each iteration have balanced (approximately equal) numbers of foreground pixels, as far as this is possible. Classification step was performed using Support Vector Machine (SVM) with Radial Basis Function (RBF). Two-stage classification scheme based on the level of granularity of the feature extraction method is employed. Classes with high values in the confusion matrix are merged at a certain level and granularity

features from the level that best distinguishes them are employed for each group of merged classes. The recognition result for handwritten characters for CEDAR Character Database is 94.73% and for the handwritten digit using MNIST Database is 99.03%.

John Thornton et al. [197] demonstrated the superiority of SVM-based approaches for offline cursive character recognition. In particular, Camastra's 2007 study showed SVM to be better than alternative LVQ and MLP approaches on the large C-Cube data set. Te Camastra's SVM study was revisited in order to explore the effects of using an alternative modified direction feature (MDF) vector representation, and to compare the performance of a RBF-based approach against both SVM and HVQ. The results show that SVMs still have the better performance, but that much depends on the feature sets employed. The use of more sophisticated MDF feature vectors produced the poorest results on this data set despite their success on signature verification problems.

John Thornton et al. [198] presented a particular model of the neocortex developed by Hawkins, known as hierarchical temporal memory (HTM). The aim is to evaluate its ability to represent temporal sequences of input within a hierarchically structured vector quantization algorithm. These temporal pooling features of HTM on a benchmark of cursive handwriting recognition problems is tested and compare it to a current state-of-the-art support vector machine implementation. The results show that a relatively simple temporal pooling approach can produce recognition rates that approach the current state-of-the-art without the need for extensive tuning of parameters. The results show that temporal pooling performance is surprisingly unaffected by the use of preprocessing techniques. Comparing the HTM's performance with Camastra's results shows that the HTM was better than both the LVQ and MLP implementations and achieve 85% recognition using default settings.

Chunpeng Wu et al. [199] proposed a handwriting recognition method based on relaxation convolutional neural network (R-CNN) and alternately trained relaxation convolutional neural network (ATR-CNN). The relaxation convolution layer adopted in R-CNN, unlike traditional convolutional layer,

does not require neurons within a feature map to share the same convolutional kernel, endowing the neural network with more expressive power. As relaxation convolution sharply increases the total number of parameters, alternate training in ATR-CNN to regularize the neural network during training procedure is adopted. The proposed network [199] achieved the state-of-the-art accuracy both on handwritten digit dataset MNIST and ICDAR'13 Competition Dataset with an error rate of 3.94%, further narrowing the gap between machine and human observers (3.87%).

Abdeljalil Gattal et al. [201] demonstrates how the combination of oriented Basic Image Features (oBIFs) with the background concavity features can be effectively employed to enhance the performance of isolated digit recognition systems. By applying a uniform grid sampling to the image, the features are extracted without any size normalization from the complete image as well as from different regions of the image. One-against-all support vector machine (SVM) is used for classification. The experimental study is conducted on the standard CVL single digit database. A recall of 93.63% and a precision of 93.70% are reported using the oriented Basic Image Features (oBIFs22) extracted from the four regions of the digit image. The performance of the combinations of BF1, BF22, oBIFs22 and oBIFs3 (or oBIFs4) making a feature vector of dimension 277 is recall of 95.19% with a precision of 95.21%.

2.3 A brief survey of HCR research in Indian scripts.

Ministry of Communication and Information Technology, Government of India, has initiated Technology Development for Indian Languages (TDIL) programme and thirteen Resource Centres for Indian Language Technology Solutions (RCILTS) have been established under this project. Initiatives have been taken for long term research for development of Machine Translation System, Optical Character Recognition, On-line Handwriting Recognition System, Cross-lingual Information Access and Speech Processing in Indian languages by this programme. Commercial systems for machine printed characters are developed for some Indian

scripts namely Assamese, Bangla, Devnagari, Malayalam, Oriya, Tamil and Telugu.

Active work on recognition of handwriting in Indic scripts is recent. Researchers in India mainly focused on English and working on Indic languages was not considered fashionable until Technology Development in Indian Languages (TDIL), Department of Information Technology, under the Ministry of Communication and Information Technology started funding research consortia on several language technologies in the country. Under the aegis of TDIL, there is a consortium for Online Handwriting Recognition in Indian languages and currently researchers have developed word-level recognition engines with various levels of accuracy for Tamil, Kannada, Hindi, Malayalam, Telugu and Bangla, and character-level engines for Assamese and Punjabi. Currently, the recognition engine for Tamil is fairly mature [10].

Siddarth et al. [205] using statistical and background directional distribution features of gurumukhi Basic Characters with SVM classifier achieved accuracy of 95.04%.Singh et al. [206] developed a system using Gabor Filters for Recognition of Handwritten Gurmukhi Basic Characters with SVM with 94.29% accuracy.

Pathan et al. [207] developed offline handwritten isolated urdu 46 Basic characters recognition system using Invariant Moments and SVM and accuracy achieved is 93.59%. Soman et al. [208] developed a system with accuracy 92.26% for Telugu Consonants and 92.0% for Vowel Modifiers using Multi-classifier.

Rajput et al. [209] presented a system for recognition of Kannada Numerals using Fourier and Chain Code with SVM and accuracy is 98.45%. Jangid [210] presented a system for recognition of Devnagari Basic Characters using Statistical features with SVM and accuracy is 94.89%.

Shelke and Apte [211] presented a system for recognition of handwritten Marathi compound character recognition scheme using neural networks and wavelet features. Accuracy obtained is 96.23%

The recognition accuracy of 91.54% using SVM Classifier with structural features for degraded printed Gurumukhi script was reported

[132]. The recognition accuracy of the Gurumukhi OCR without post processing was 94.35%, which was increased to 97.34% on applying the shape based post processor to the recognized text[133]. A recognition system is proposed to recognize handwritten numerals in both Devnagri (Hindi) and English in the paper by G.S.Lehal and Nivedan Bhatt [145]. Off-line handwritten character recognition of Devnagari, the most popular script in India developed by U.Pal et al.[146] reported 94.24% recognition accuracy. The features used for recognition purpose are mainly based on directional information obtained from the arc tangent of the gradient. A modified quadratic classifier is applied on these features for recognition. Bhattacharya et al. [147] presented a multistage cascaded recognition scheme using wavelet based multiresolution representations and multilayer perceptron classifiers for the recognition of mixed handwritten numerals of three Indian scripts Devanagari, Bangla and English. In the paper by Umapada Pal et al.[148],a tri-lingual (English, Hindi and Bangla) 6-digit full pin-code string recognition is proposed andaccuracy obtained is 99.01% reliability from the proposed system when error and rejection rates are 0.83% and 15.27%, respectively. M.K.Jindal et al.[149] proposed two algorithms to segment touching characters, and one algorithm to segment overlapping lines in degraded printed Gurmukhi document with Various categories of touching characters in different zones, along with their solutions.

In the paper by U. Pal et al.[150], a comparative study of Devnagari handwritten character recognition using twelve different classifiers and four sets of feature is presented in order to get idea of the recognition results of different classifiers and to provide new benchmark for future research.

Jomy John et al. [204] proposed a two-stage approach for handwritten Malayalam character recognition. The first stage is a group classifier, where a group consists of similar characters and those that misclassify among themselves. A character assigned to a group in the first stage is classified to a particular character class in the second stage. The overall classification accuracy obtained is 97.72%.

U. Pal et al. [203] proposed a system for bangla handwritten numeral Recognition. Features are obtained from the concept of water overflow from

the reservoir as well as topological and structural features of the numerals. The overall recognition accuracy of the proposed scheme is about 92.8% from 12000 data.

Jomy John et al. [202] proposed a handwritten character recognition system for Malayalam language. The features are gradient and curvature. Directional information from the arc tangent of gradient is used as gradient feature. Strength of gradient in curvature direction is used as the curvature feature. The proposed system uses a combination of gradient and curvature feature in reduced dimension using Principal Component Analysis as the feature vector. The accuracies in two different datasets using SVM with Radial Basis Function (RBF) kernel are 96.28% and 97.96%.

K. Roy et al. [16, 17] proposed a System for Indian Postal Automation. Aspect Ratio Adaptive Normalization (ARAN) technique is performed instead of normalization of the character image by linearly mapping onto a standard plane by interpolation/extrapolation and without computing any feature from the image. The raw images are normalized into 28x28 and pixel size are used for classification using MLP giving the result of classifier on English numerals of Indian pin code is only 93.0%.

2.4 Conclusion

This chapter reviewed literature survey of Manipuri script along with a brief historical background and also reviewed many aspects of handwritten digits and character research for English as well as some Indian scripts. The next chapter deals with the preprocessing and segmentation of digits, lines, words and non-touching characters.

Chapter 3
Preprocessing and Segmentation

This chapter consists of several preprocessing steps which aim at producing images that are easy for the recognition process to operate more accurately and then presents segmentation of lines, segmentation of digits or characters, words from segmented line and characters are presented.

3.1 Preprocessing

The goal of preprocessing in handwriting recognition systems is to reduce irrelevant information such as noise that increases the task complexity in a writer-independent recognizer. The recognition of handwritten characters of the Manipuri script in this thesis is developed to deal with unconstrained handwritten samples written by multiple-writers.

The handwritten samples of digits and characters are acquired from 14 students and 2 faculty members of the Department and are sampled on A4 sized paper.

All 16 writers can speak the Manipuri language and write the Manipuri script although mother tongues are different for some writers. These handwritten samples are scanned with 300 dpi using a flatbed HP scanner and laptop computer. Some pages are in black and white or gray scale format and some pages are in color (RGB) format. As previously scanned some color format pages are available and for the curiosity to observe the results after classification, both the file formats are used in training and testing process.

There are 54 character classes (without the Lum symbol) of Manipuri Script. Each person has written the same 50 samples of the said character symbol in a page. These pages are scanned with 300 dpi. The two pages written by two persons having 50 samples each of the same character symbol are put in a single page. The data set has 5400 characters for 10 pages having 100 samples each corresponding to 54 class characters of the script. In addition, there are 16 pages of 50 samples for digits (800 samples) and also 16 pages of 50 samples for 10 class (mapum mayek) characters (800

samples). So, there are 7000 samples for digits and characters in total for the 54 classes.

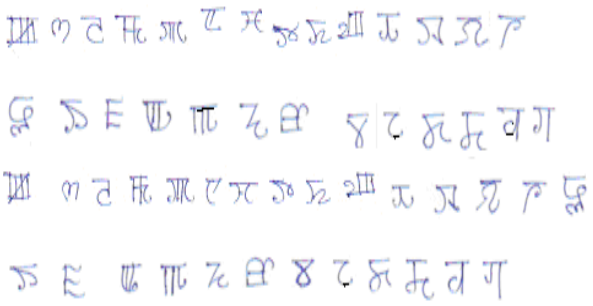


Figure 3.1: Some handwriting samples of 27 alphabets of Manipuri script

Figure 3.2: Some handwriting samples of 18 characters of Manipuri script (8 Lonsums + 8 Cheitaps + 2 special letters)

Figure 3.3: Some handwritten sample text lines of Manipuri script

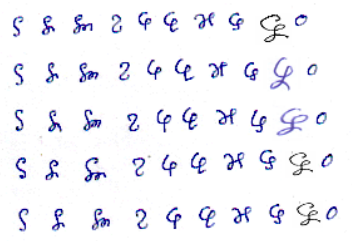


Figure 3.4: Some handwritten sample digits of Manipuri script

The scanned images were saved in windows JPG, Bitmap and Tif/Tiff format. Every scanned image is associated with background noise and other noise which creeps in due to errors in scanning. Some samples of 27 alphabets image are shown in Figure 3.1. Some samples of 18 characters (8 Lonsums + 8 Cheitaps + 2 special letters) are shown in Figure 3.2. Some samples of text lines are shown in Figure 3.3. And some samples of digits are shown in Figure 3.4.

Various types of skew angle detection and correction are used by many researchers in the literature. A technique is described in [127] as the skew angle may be determined by calculating horizontal and vertical projections at different angles at fixed interval in the range [0^0 to 90^0]. The angle, at which the difference of the sum of heights of peaks and valleys is maximum, is identified as the skew angle. As the algorithm for skew angle detection and correction was not implemented in this thesis, instead, using the Adobe Image Ready and Adobe photoshop (cs2) software , skew detection and correction of the sacked input file are performed. The process detects whether the handwritten line has been written on a slope, and then rotating the line if the slope's angle is too high so that the headline of the word is horizontal.

The RGB color image of a page converted into the grayscale intensity image by eliminating the hue and saturation information while retaining the luminance. The input RGB image or color map file containing the deskewed text lines are converted to grayscale. Grayscale format means that the intensity of each pixel in the image may vary between a value of 0 and 255. The value zero indicates a black pixel, whereas a white pixel is represented by the value 255.

In Matlab, RGB2GRAY function converts RGB image or color map to grayscale by eliminating the hue and saturation information while retaining the luminance. I = RGB2GRAY(RGB) converts the truecolor image RGB to the grayscale intensity image I. The image now has 256-gray levels.

3.1.1 Thresholding

Every scanned image is associated with background noise and other noise which creeps into due to errors in scanning procedure. It is useful to be able to separate out the regions of the image corresponding to components in which we are interested from the regions of the image that correspond to background. Various shades of gray are represented between these two values. Many researchers have decided to convert the initial gray-level images into a less storage intensive format i.e. a binary (0 and 1), black and white format.

It is argued whether recognition performed on features directly extracted from gray-scale or from binary images produces the better result. The fact remains that many researchers have used to perform recognition on binary characters for simplicity and computational speed.

The process of converting a gray-level image to a binary image is called thresholding or binarisation. This is the operation of selecting which elements of a character image may be considered the background (white pixels) and which elements are to be considered the character itself (black pixels). Some threshold is usually used so that pixels with a luminance over the threshold are marked as being background pixels while pixels with a luminance under the threshold are considered to be part of the character body. Selecting an appropriate threshold has been the subject of active research for a number of years.

Binarisation (thresholding) is often desirable to represent gray-scale or color images as binary images for reducing the data storage and increase processing speed. Transformation of an input image f to a binary image g such that:

$$g(i,j) = \begin{cases} \mathbf{1} \;\; for\; f(i,j) \geq T \\ \mathbf{0} \;\; for\; f(i,j) < T \end{cases} \quad \ldots\ldots (3.1)$$

A number of techniques exist for the automatic detection of thresholds in images. There are many advantages to storing word images in this format. Firstly, it is easier to manipulate an image with only two levels of colour. Also, further processing will be faster, computationally less expensive and will allow for more compact storage. There are of course a number of

disadvantages such as loss of information from the original image or the introduction of anomalies or noise.

In the histogram processing, a threshold can then be chosen in between the two peaks, this is known as valley-seeking. This method is successful; however images do not always contain well-differentiated foreground and background intensities due to poor contrast and noise. Perfect thresholding is a difficult task. An iterative thresholding presented by Ridler and Calvard[36] is a switching mechanism which decides always between background and object, where the corners of the image are taken as initial values for the background [31].

A well known thresholding algorithm is the Otsu's algorithm[33]. Otsu's algorithm, regards the histogram as probability values and defines the optimal threshold value as one that maximizes the between-class variance, where the distributions of the foreground and background points are regarded as two classes. Each value of the threshold is tried and one that maximizes the criterion is chosen. To examine the formulation of this histogram based method, the normalized histogram is treated as a discrete probability density function, as in

$$P_r(r_q) = n_q / n \qquad q=0,1,2,\ldots,L-1$$

Where n is the total number of pixels in the image, n_q is the number of pixels that have intensity level r_q and L is the total number of possible intensity levels in the image. Now suppose that a threshold k is chosen such that C_0 is the set of pixels with levels [0,1,.....k-1] and C_1 is the set of pixels with levels [k,k+1,....,L-1]. Otsu's method chooses the threshold value k that maximizes the between-class variance which is defined as

$$\sigma^2_B = w_o(\mu_o - \mu_T)^2 + w_1(\mu_1 - \mu_T)^2 \qquad .....(3.2)$$

Where

$$w_o = \sum_{q=0}^{k-1} P_q(r_q) \qquad w_1 = \sum_{q=k}^{L-1} P_q(r_q)$$

$$\mu_o = \sum_{q=0}^{k-1} qP_q(r_q)/w_o \qquad \mu_1 = \sum_{q=k}^{L-1} qP_q(r_q)/w_1$$

$$\mu_T = \sum_{q=0}^{L-1} qP_q(r_q)$$

In other words, this method of thresholding involves in choosing the threshold to minimize the intraclass variance of the thresholded black and white pixels.

The function *graythresh* of Matlab takes an image, computes its histogram, and then finds the threshold value that maximizes σ^2_B. Otsu's method [33] of thresholding is performed as a preprocessing step. The grayscale intensity image is then thresholded using Otsu's method using the function *graythresh*. This function returns the threshold value and this value is used to convert the grayscale image to binary image using the function *im2bw*.

After thresholding, the image is converted to a 1 bit binary image, thus making it simpler to carry out further operations as the image is reduced to a 2-d matrix of 1 s and 0s. In Figure 3.5, some handwritten sample digits, alphabets and a word has been shown.

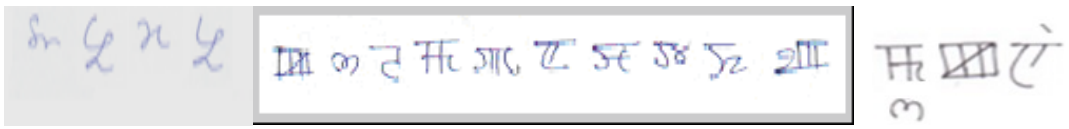


Figure 3.5: Sample handwritten digits, alphabets and a word

Figure 3.6: Binarised (Thresholded) digits, alphabets and a word

Figure 3.7: (a) A word (Laininghou means GOD) from an old manuscript (b) Thresholded word.

The corresponding binarised (thresholded) image is shown in Figure 3.6. Figure 3.7(a) shows a word (Laininghou means GOD) from an old manuscript and the corresponding thresholded image is shown in Figure 3.7(b).

3.1.2 Noise Removal

To detect and eliminate noise elements that were either prominent in the original word images or that were introduced at the thresholding stage is important for further processing.

To reduce noise in document processing, the *k*Fill filter has been extensively used[158]. This is a more general filter designed to reduce isolated noise and noise on contours upto a selected limit in size. The size adjustment parameters is the *k* of *k*Fill. The parameters of *k*Fill may be set accordingly in order to retain corners of 90^{o} or greater. In raster scan order, at each image pixel, filling operation are performed within a *k* x *k* window. The core or the interior region of the window is (*k*-2) x (*k*-2). The perimeter or the neighborhood is 4(*k*-1)

Depending on the pixel values in the neighborhood, filling operations of the core either to OFF or ON are decided. Two sub iterations, one for ON fills and other OFF fills are performed for each iterations. The process stops automatically, when no filling occurs on two consecutive sub iterations. If the core pixels are OFF(ON), then the decision of filling operation is ON(OFF) and it depends furthermore on three variables, n, c and r. These values are determined from the neighborhood pixels. If the filling operation is ON(OFF), *n* equals the number of ON(OFF) pixels in the neighborhood. The number of connected groups of ON pixels in the neighborhood is denoted by *c*. The number of corner pixels that are ON (OFF) is represented by *r*. From the window size *k,* the values of *n* and *r* are derived. Filling operations are performed if the following conditions are met:

(c=1) AND [(n>3k-4) OR (n=3k-4) AND r=2].

The significance of these conditions is as follows:

i) n>3k-4: The degree of smoothing is controlled by this term. Enhanced smoothing is obtained if the threshold value of *n* is reduced.

ii) n=3k-4 AND r=2: This ensures that corners of ≤ 900 are not rounded. Greater noise reduction would occur if this condition is absent but corners may be rounded.

iii) c=1: The connectivity of two regions either joining the two regions or separating two parts of the same connected region is ensured. If

there is no constraint on c , it means connectivity are allowed to change. Then greater smoothing would occur, but without the assurance that the number of distinct regions would remain constant.

At the expense of foregoing greater noise reduction, erring on the side of retaining image features are the conservative conditions. A procedure for establishing conditions in a particular application is to start experimenting with the default conditions and relaxing them until a middle ground is found with good noise reduction and a little unwanted alteration of the image[158].

To remove as much noise as possible while still retaining dots, periods, and serif, the variable n is set relative to the text size. The kFill filter has been used in preprocessing step as an improvement in subsequent task such as segmentation of lines, digits or characters and in analyzing the features.

After thresholding the deskewed input image page of characters, as shown in Figure 3.8(a), using Otsu's algorithm and from the binary image page, all connected components (objects) that have fewer than 10 pixels as salt and pepper noise are removed using *bwareaopen* procedure in MATLAB thereby producing another binary image as shown in Figure 3.8: (a) A word with noise (b) word without noise.

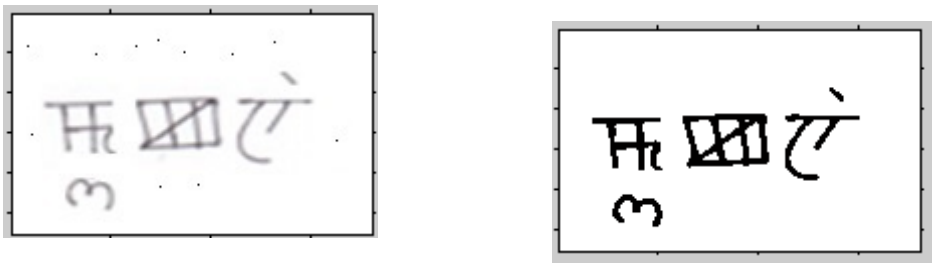


Figure 3.8: (a) Sample word with noise **(b) word without noise**.

3.2 Segmentation of Lines

From thresholded image, first line is segmented using a simple line extraction algorithm. The line extraction algorithm segments line by line from the thresholded image are shown in Figure 3.9.

The row coordinates of the first line are found by scanning rows of the deskewed negative binary input page, where integer value 0 represents

background and 1 represents foreground, for non zero sums of row as beginning of line and zero sums as the end of the line and the line is extracted and stored in matrix L1 as shown in Figure 3.11. Similarly, all the remaining lines of the page are extracted.

Figure 3.9: Binary image page of characters

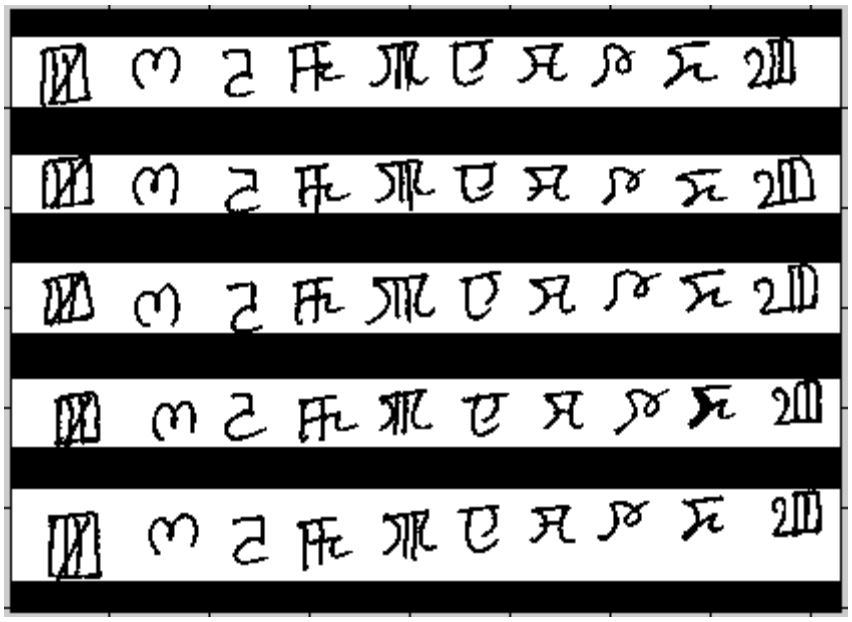


Figure 3.10: Binary image page with black pixels between lines

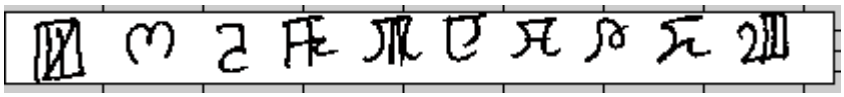


Figure 3.11: Segmented first line

3.3 Segmentation of digits

From the segmented line of digits, the digits are extracted using the following extraction technique.

After thresholding the input image, edges of the digits are found using Sobel method using structuring element (SE) of size 3. Edge detection is by far the most common approach for detecting meaningful discontinuities in gray level. An edge is a set of connected pixels that lie on the boundary between two regions. It is already observed that the magnitude of the first derivative can be used to detect the presence of an edge at a point in an image. It is understood that, to be classified as a meaningful edge point, the transition in gray level associated with that point has to be significantly stronger than the background at that point. Figure 3.12 shows edge detected image for digits. The algorithm for edge detection can be detailed as follows. Since we are dealing with local computations, the method of choice to determine whether a value is significant or not is to use a threshold. Thus we define a point in an image as being an edge point if its two-dimentional first order derivative is greater than a specified threshold. A set of such points that are connected according to a predefined criterion of connectedness is by definition an edge. First order derivative in an image implemented using the magnitude of the gradient.

Figure 3.12: Edge detected image

Figure 3.13: Dilated image digits

Figure 3.14: Filled image

Figure 3.15: Located digits with bounding boxes

For a function *f*(x,y), the gradient of f at coordinates (x, y) is defined as the two-dimensional column vector.

$$\nabla f = \begin{vmatrix} Gx \\ Gy \end{vmatrix} = \begin{vmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{vmatrix} \qquad \text{..... (3.3)}$$

The magnitude of this vector is given by

$$\nabla f = \text{mag}(\nabla f)$$
$$= (Gx^2 + Gy^2)^{1/2}$$
$$= \left[\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial x}\right)^2\right]^{1/2} \qquad \text{..... (3.4)}$$

It is common practice to approximate the magnitude of the gradient by using absolute values instead of squares and square roots:

$$\nabla f \approx |Gx| + |Gy| \qquad \text{..... (3.5)}$$

The difference between the third and first rows of the 3 x 3 image region approximates the derivative in the x-direction and the difference between the third and the first columns approximates the derivative in the y-direction.
Where

$$Gx = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)$$

and $$Gy = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7) \qquad \text{..... (3.6)}$$

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

(a)

-1	-2	-1
0	0	0
1	2	1

(b)

-1	0	1
-2	0	2
-1	0	1

(c)

Figure 3.16: (a) A 3x3 region of an image where the z's are gray level values (b) and (c) Sobel operators used to compute the gradient.

Sobel operators shown in Figure 3.16 (b) and (c) are used to implement Eq. (3.5) via the mechanics of spatial filtering given in Eq.(3.7).
Linear filtering of an image f of size M x N with a filter mask of size m x n is given by the expression:

$$g(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t)\ f(x+s, y+t)) \qquad \text{... ... ... (3.7)}$$

Where m=2a+1 and n = 2b+1 where a and b are nonnegative integers. Our mask size is 3 x 3, therefore m=n=3 and a = b=1.

For 3 x 3 mask, the response R at any point (x, y) in the image is given by

$$R= w_1z_1+w_2z_2+.....w_9z_9$$

$$= \sum_{i=1}^{9} w_i z_i \qquad \text{……..(3.8)}$$

The process consists of simply moving the filter mask from point to point in the image and the response is given by a sum of products of the filter coefficients and the corresponding image pixels in the area spanned by the filter mask. In Matlab, EDGE function finds edges in intensity image. It takes an intensity or a binary image I as its input, and returns a binary image BW of the same size as I, with 1's where the function finds edges in I and 0's elsewhere. EDGE supports six different edge-finding methods. In our experiment, we have used the Sobel method. The Sobel method finds edges using the Sobel approximation to the derivative. It returns edges at those points where the gradient of I is maximum.

Then the operation of image dilation of morphological image processing is performed as shown in Figure 3.13. Dilation is an operation that 'grows' or thickens objects in a binary image. Image dilation is used for adding pixels to the boundaries of objects in an image using structuring element of size 3x3 so that the operation thickens objects in a binary image. The rule used for the dilation operation is that the value of the output pixel is the maximum value of all the pixels in the input pixel's neighborhood. In a binary image, the morphological dilation function sets the value of the output pixel to 1 because one of the elements in the neighborhood defined by the structuring element is 1.

Mathematically, dilation is defined in terms of set operations. The dilation of A by B, denoted $A \oplus B$, $= \{z|(\hat{B})_z \cap A \neq \emptyset\}$ .....(3.9)

Where $\emptyset$ is the empty set, A is the input image and B is the structuring element. The reflection of set B denoted by $\hat{B}$ is defined as

$$\hat{B} = \{w|\ w = -b, for\ b \in B\} \qquad \text{..... (3.10)}$$

The translation of set B by point z= (z_1, z_2) denoted $(B)_z$, is defined as

$$(B)_z = \{c|\ c = a + z,\ for\ a \in B\} \qquad \text{..... (3.11)}$$

We have used flat structuring element of size 3 x 3 having all 9 neighbors' values of 1. The dilation of A by B is the set consisting of all structuring element origin locations where the reflected and translated B overlaps at least some portion of A.

Image dilation is used for adding pixels to the boundaries of objects in an image using structure element of size 3 x 3 so that the operation thickens objects in a binary image. The rule used for the dilation operation is that the value of the output pixel is the maximum value of all the pixels in the input pixel's neighborhood. In a binary image, the morphological dilation function sets the value of the output pixel to 1 because one of the elements in the neighborhood defined by the structuring element is 1.

Filling operation on the image is performed as shown in Figure 3.14. This operation fills holes in the binary image. This operation fills holes in the binary image. A hole is a set of background pixels that cannot be reached by filling in the background from the edge of the image. Reconstruction is a morphological transformation involving two images and a structuring element, instead of a single image and structuring element. One image, the marker, is the starting point for the transformation. The other image, the mask, constrains the transformation. The structuring element used defines the connectivity. If we choose the marker image, f_m to be 0 everywhere except on the image border, where it is set to 1—*f:*

$$fm(x,y) = \begin{cases} 1 - f(x,y) \text{ if } (x,y) \text{is on the border of } f \\ 0 \qquad\qquad\qquad\qquad \text{otherwise} \end{cases} \quad \ldots\ldots\ldots\ldots (3.12)$$

Then g =[R_f^c (fm)]^c has the effect of filling the holes in f . In Matlab, imfill performs this computation automatically when the optional argument 'holes' is used.

Then connected component analysis is performed on the filled image and basic properties such as area, centroid and bounding boxes of each object are found as shown in Figure 3.15. Connected component analysis is performed using 8-connected. Two foreground pixels p and q are said to be 8-connected if there exists an 8-connected path between them. The union of $N_4(P)$ and $N_D(P)$ are the 8-neighbors of p, denoted by $N_8(P)$. The pixels p and q are said to be 8-adjacent if q ∈ $N8(P)$. For any foreground pixel, p, the set of all foreground pixels connected to it is called the connected component

containing *p*. In Matlab, *bwlabel* computes all the connected components in a binary image. The syntax is: [L num] =bwlabel(Ifill) where Ifill is the input binary image after filling holes and the default value for the connectivity is 8-connected. Output L is the label matrix and num(optional) gives the total number of connected components found. Background pixels are labeled 0. Then only the basic properties of image regions (blob analysis) are measured using the function STATS = regionprops(L)

'Area' — Scalar; the actual number of pixels in the region. (This value might differ slightly from the value returned by bwarea, which weights different patterns of pixels differently). 'BoundingBox' — 1-by-ndims(L)*2 vector; the smallest rectangle containing the region. BoundingBox is [ul_corner width], where ul_corner is in the form [x y] and specifies the upper left corner of the bounding box. Width is in the form [x_width y_width] and specifies the width of the bounding box along each dimension. 'Centroid' – 1-by-ndims(L) vector; the center of mass of the region. Note that the first element of Centroid is the horizontal coordinate (or x-coordinate) of the center of mass, and the second element is the vertical coordinate (or y-coordinate). All other elements of Centroid are in order of dimension.

The equations below calculate the x and y centroid coordinates for a given labeled object:

x_centroid = sum(x_{ij} x i)/num_foreground_pixels

i = 0, … , NC-1 where NC (Number of Columns) is the width of the object

j = 0, … , NR-1 where NR (Number of Rows) is the height of the object

x={0,1} Indicates the value of the current pixel being examined

(Background/foreground)

y_centroid = sum(y_{ij} x j)/num_foreground_pixels

i = 0, … , NC-1 where NC (Number of Columns) is the width of the object

j = 0, … , NR-1 where NR (Number of Rows) is the height of the object

y = {0,1} Indicates the value of the current pixel being examined

(background/foreground)

Once each centroid has been calculated, the coordinates (x_centroid, y_centroid) denote the centre of the connected component or object being currently examined. For extracting the object, imcrop function is used. The syntax is:

I2 = IMCROP(I,RECT) where RECT is a 4-element vector with the form [XMIN YMIN WIDTH HEIGHT]; these values are specified in spatial coordinates and are already available in the properties of bounding box.

Before the image component is resized, it is necessary to remove the white spaces in the boxes. The sub-images are to be cropped sharp to the border of the character in order to standardize the sub-images. The image standardization is done by finding the maximum row and column with 1s and with the peak point, by increasing and decreasing the counter until meeting the white space, or the line with all 0s as shown in Fig.3.15.(a)

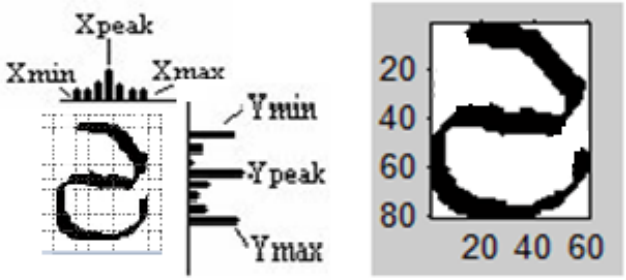


Figure 3.15:(a) Image standardization (b) resized image (60x80)

Then re-scaling the size of component (digits) to 60x80 using bicubic interpolation are performed. The function IMRESIZE resizes an image of any type using the specified interpolation method in MATLAB.

3.4 Segmentation of words

Figure 3.16 shows lines having four words are projected its pixel density vertically. It was observed that three low pixel densities marked as 1,2 and 3 are detected as word segmentation area.

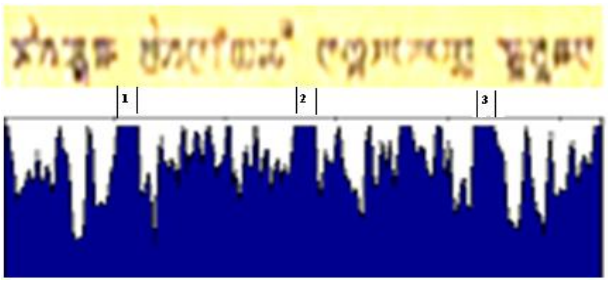


Figure 3.16: (a) Three segmentation areas with vertical projection profile

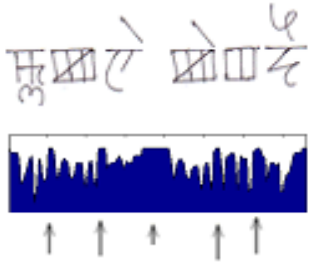


Figure 3.16: (b) One segmentation area with vertical projection profile

Based on the analysis of the vertical projections of each line, an area of low pixel density that exceeded a given threshold would be identified as an obvious segmentation area for word as shown in Figure 3.16.

This may be further analysed using connected component and bounding box analysis. Based on the rules pertaining to the height and width of the bounding boxes, the connected component is then split or merged as described by Cesar [71]. This algorithm involved the location of connected components followed by bounding box analysis. Connected component location involves searching the input image for connected foreground (black) regions. Upon location of these regions it is possible to decide whether the connected components should be merged or split. Bounding box analysis is often used following the location of connected components. The "bounding box" simply refers to the location and dimensions of a connected component. Bounding box analysis can be very helpful; it provides information such as the proximity of connected components. This information may be used to decide whether two adjacent connected components should be joined. The size of the bounding boxes may be used to decide whether connected components should be split. The recorded information of the BoundingBox is [*ul_corner width*], where ul_*corner* in the form [x y z] and specifies the upper left corner of the bounding box, width is in the form [x_width y_width] and specifies the width of the bounding box along each dimension. The location of the upper-most and bottom-most rows as well as the location of the left-most and right-most columns can be obtained. These values provide the bounding box locations for each connected component. The connected components were then split or merged based on rules pertaining to the height and width of the bounding boxes. Prior to splitting and merging, any small fragments were eliminated if their size (given by bounding box information) was smaller than some threshold.

3.5 Segmentation of Characters of a Word

A handwritten word having characters in different zones with horizontal projection of pixels is shown in Figure 3.17.

The three zones namely upper zone, mid zone and lower zone are shown based on the horizontal projection profile information. The locations are named and numbered based on the analysis of the row pixel densities indicated by upper line(1), headline (2), stroke width (2-3), base line(4) and lower line (5).

From the thresholded image in the segmented word, edge detected image components, dilated image, filled image and then connected component analysis are performed on the filled image as the same procedure is employed as in the case of segmentation of digits mentioned in previous section 3.1.5 . The corresponding results obtained are shown in Figure 3.17-Figure 3.22.

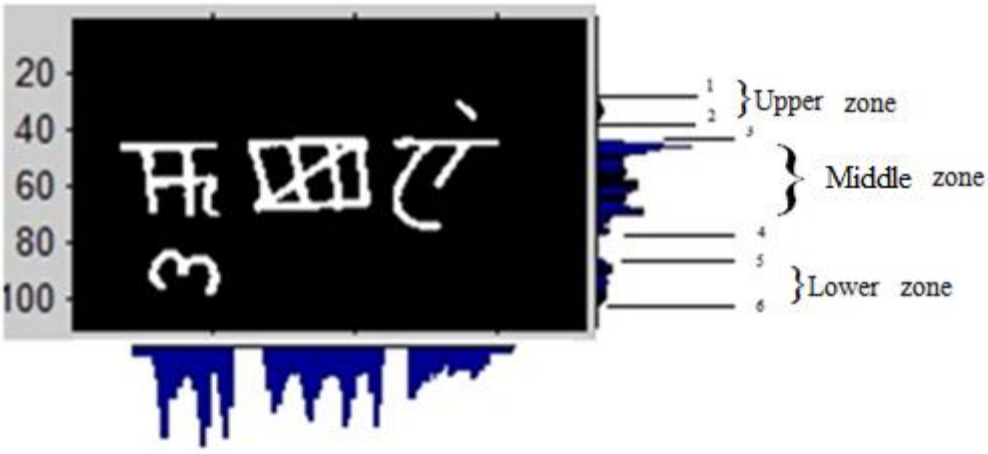


Figure 3.17(a) Word having isolated characters with different zones with horizontal and vertical projection profiles.

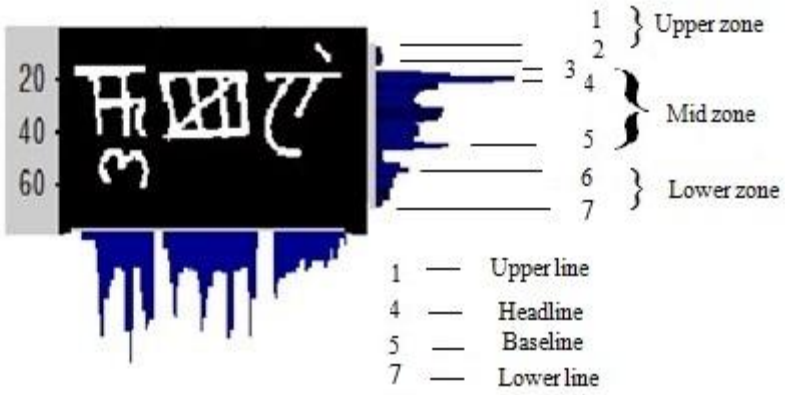


Figure 3.17(b) Word having overlapped characters with different zones with horizontal and vertical projection profiles.

In Figure 3.17(a) , the row coordinates of lines of upper zone are found by scanning rows of the deskewed negative binary input page for non zero sums of row as beginning of zone (Upper zone 1) and zero sums as the end of the zone (Upper zone 2) and Similarly, the locations of Middle zone (3 and 4) and Lower zone (5 and 6) are extracted and also the extracted text lines for Upper zone,Mid zone and Lower zone are saved in sub-image matrices L1,L2 and L3.

The column coordinates of characters of upper zone are found by scanning columns of L1 for non-zero sums of columns as beginning of character and zero sums as the end of the character and Similarly, the column coordinates of characters of Middle zone (3 and 4) and Lower zone (5 and 6) are extracted and stored in corresponding 1-D matrices *matL1,* matL2 *and* matL2.

The connected components in the binary image of upper zone line L1 are performed using the 8-nearest neighbor connected objects, the numbers of objects or characters N and the matrix L of the same size containing labels of the connected components are returned. The elements of L are integer values greater than or equal to 0. The pixels labeled 0 are the background. The pixels labeled 1 make up one object; the pixels labeled 2 make up a second object, and so on.

Then the *imcrop* procedure on the image L1 with bounding box *rect* is performed. *rect* is a four-element position vector [xmin ymin width height] that specifies the size and position of the crop rectangle. A detailed report based on the experiment on word recognition is presented in Section 6.1.4.

Words having overlapped characters as shown in Figure 3.17(b) are segmentedm using the zone information, upperline, Head line base line and lower line. A detailed report based on the experiment on word recognition is presented in Section 6.1.5.

3.6 Conclusion

In this chapter, preprocessing and segmentation are presented. The importance of preprocessing is highlighted and it is composed of deskew, conversion of RGB to grayscale intensity image, Thresholding (binarization) and noise removal are presented. For some steps, well known techniques found in the literature are used.

Then the method of segmentation of lines, words, digits and characters are presented. Segmentation of words using vertical and horizontal projection profile are presented and then further analysis with connected component and bounding box are performed. Segmentation of isolated and overlapped non-touching characters from a word are presented. The next chapter presents proposed hybrid feature extraction technique.

Chapter 4

Probabilistic Features (PF), Fuzzy Features (FF) and Hybrid Feature (HF) Extraction Technique

The features extraction is one of the most important aspects of handwritten character recognition system. The objective of feature extraction is to extract the salient information that needs to be applied to the recognition process. It reduces data dimensionality by determining certain feature properties that distinguish input patterns. A feature extraction technique called Hybrid Feature (HF) is proposed here. The probability features and (PF) and Fuzzy features (FF) are chosen due to their ability to successfully extract important features from images, which have enabled accurate recognition rates.

4.1 Feature Extraction

This section describes the proposed hybrid feature extraction technique in detail. The purpose of feature extraction is to get the most relevant and the least amount of data representation of the character images in order to minimize the within-class pattern variability while enhancing the between-class pattern variability. There are categories of features: statistic features, structural features and fuzzy features. Many researchers reported different categories of feature extraction methods for off-line recognition of isolated characters such as template matching; (2) deformable templates; (3) unitary image transforms; (4) graph descriptions; (5) projection histograms; (6) contour profiles; (7) zoning; (8) geometric moment invariants; (9) Zernike moments; (10) spline curve approximations; and (11) Fourier descriptors. The mentioned methods could be applied to one or more of the following character forms: (1) gray-level character images; (2) binary character images; (3) character contours; and (4) character skeletons or character graphs.

4.1.1 Feature Dimensionality Reduction and Selection

There are two types of feature dimensionality reduction. One is called feature selection, which uses some criteria to select fewer features from the original feature set. The second type uses an optimal or sub-optimal transformation to conduct feature dimensionality reduction. The latter is an information congregation operation rather than the operation of deleting less useful features.

Feature selection is an important step in OCR and HCR. In a large feature set (where, normally, the number of features is greater than 100), the correlation of features is complicated. Retaining informative features and eliminating redundant ones are a recurring research topic in pattern recognition. Generally speaking, feature extraction and feature dimensionality reduction serve two purposes: (1) to improve the training and testing efficiency, and (2) to improve the reliability of a recognition system.

4.1.2 Proposed Hybrid Feature Extraction Technique

Divergence distance measurement is one of the feature selection criteria. Intuitively, if the features show significant differences from one class to the other classes, the classifier can be designed more efficiently with a better performance.

We need to develop some methods to keep powerful discriminate features and at the same time to delete the less useful features in order to easily use random feature selection.

4.1.2.1 Recognition using K-L divergence

A commonly used distance measure density, and for its connection with information theory, is the Kullback-Leibler distance. The Kullback-Leibler divergence can be considered as a kind of a distance between the two probability densities, though it is not a real distance measure because it is not symmetric.

Gibbs' inequality [11]:

Suppose that

P= $\{p_{i,\ldots},p_n\}$ is a probability distribution. Then for any probability distribution Q= $\{q_{i,\ldots},q_n\}$, the following inequality between positive quantities (since p_i and q_i are positive numbers less than one) holds

$$-\sum_{i=1}^{n} p_i \log_2 p_i \leq -\sum_{i=1}^{n} p_i \log_2 q_i$$

With inequality if and only if $p_i = q_1$ for all $_i$. The information entropy of a distribution P is less than or equal to its cross entropy with any other distribution Q. The difference between the two quantities is the Kullback-Leibler divergence or relative entropy.

So the Kullback-Leibler divergence between two probability functions p(x) and q(y), is defined by:

$$D_{kl|p||q|} = \sum p(x) \log \frac{p(x)}{q(y)} \qquad (4.1)$$

It is the expected value of the log likelihood ratio. It therefore determines the ability to discriminate between two states of the world, yielding sample distributions p(x) and q(y). For recognising a test pattern x with another pattern y, the probability densities p(x) and q(y) using probabilistic features (PF) of the patterns are computed and then K-L divergence between the two probability densities are computed. The probabilistic features (PF) are presented in detail in the following Section.

The K-L divergence is zero if the two patterns are same and its value is low for nearly similar patterns.

4.1.2.2 Probabilistic Features (PF)

Probabilistic model may be thought of as a way of thinking about something that many things are the result of randomness. After the segmentation process, two handwritten digits written by two persons representing digit 1(one) before resizing the pattern are shown in Figure 4.1(a) and 4.1(b) and their binary values are shown in Figure 4.2(a) and Figure 4.2(b).

Figure 4.1: (a) Handwritten Digit 1 by first person (b) Handwritten Digit 1 by second person

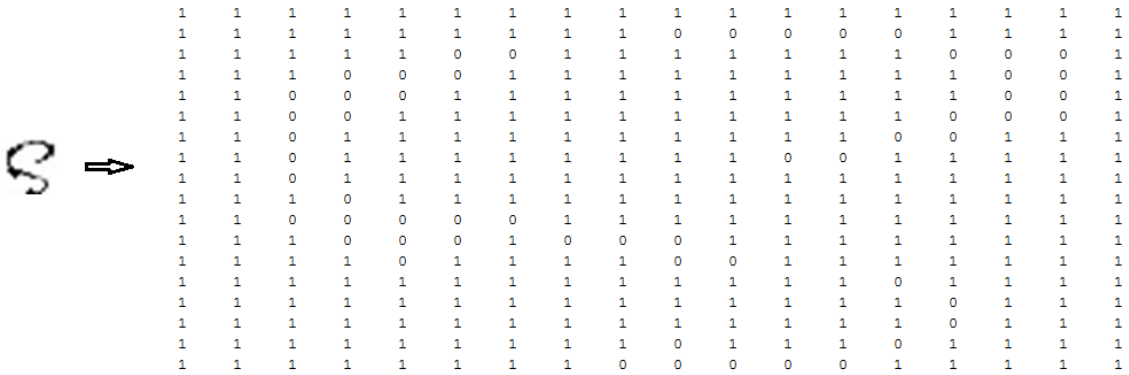


Figure 4.2(a). First person's segmented handwritten digit 1 and its binary values

(Image size is 18x18)

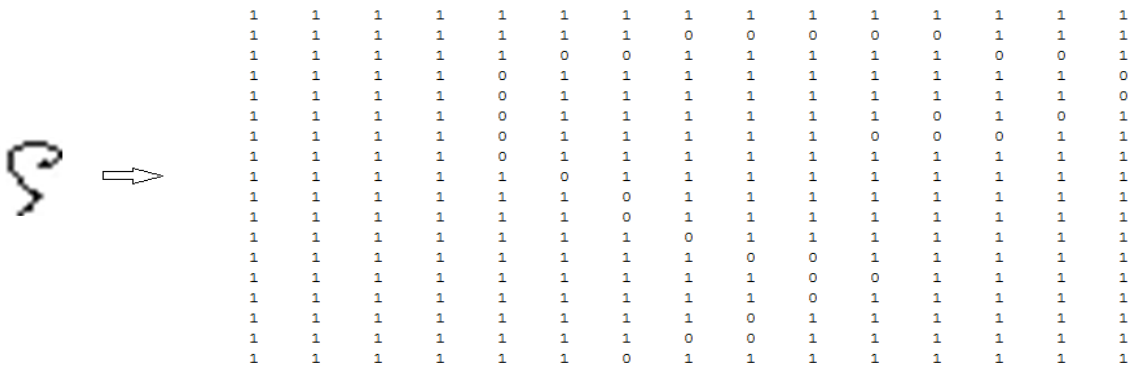


Figure 4.2(b): Second person's segmented handwritten digit 1 and its binary values

(Image size is 18x15)

For the two digits shown in Figure 4.1 (a) and Figure 4.1(b) representing 1(one) in Manipuri script written by two persons are to be compared for similarity using K-L divergence. The corresponding binary images shown in Figure 4.2 (a) and Figure 4.2(b) of handwritten digits written by two persons have image sizes of 18x18 and 18x15 respectively. As it is observed in the binary image that the zeros representing the black pixels for the pen tip movements, while writing the digits, are distributed randomly. The specific feature representation of the digit 1(one) are the distribution of the random black pixels while writing the symbol and are analogous to the *signature* of a character. From the x-y coordinates of the random black pixels of the binary images, the probabilistic densities as signatures are computed for representing as the Probabilistic Features (PF).

Two *signatures* are used to extract the probabilistic features of the digits and characters based on the given algorithms.

Signature 1 allows the characters to be rotated and Signature 2 gives a decent amount of scalability to the characters.

For finding the signatures of a character, the algorithms are as follows:

For computing *Signature 1* :

(a) Find the means of the X and Y data

From the binary image A, shown in Figure 4.2 (a), for all pixels having zero values i.e., for all black pixels (zero values) of A (i , j), compute the means of rows (X) and columns (Y).

(b) Take these means and complete a matrix of distances

Compute the matrix of distances, $E(i)= sqrt[(X-Xmean)^2 +(Y-Ymean)^2]$

(c) Normalize

Compute normalized matrix of distances by dividing with its maximum value, E = E /max (E);

(d) Bin these distances using histogram

Compute histogram of normalized matrix of distances 'E' in steps of 0.05 from 0 to 1. i.e., histogram1 = hist(E,0:0.05:1);

The MATLAB function 'hist' bins the elements of E into 21 equally spaced containers and returns the number of elements in each container.

Compute probability densities by dividing the number of elements in each container by the sum of all elements.

Thus, 21 features are extracted from the image as probability densities for Signature 1.

For computing *Signature 2:*

(a) Bin X location data

From the binary image A, shown in Fig.4.2(a), for all pixels having zero values i.e., for all black pixels (zero values) of A (i , j), bin rows (X) location data using 'hist' function with 5 bins thereby giving signature X.

(b) Bin Y location data

From the binary image A, shown in Fig.4.2(a), for all pixels having zero values i.e., for all black pixels (zero values) of A (i , j), bin column

(Y) location data using 'hist' function with 5 bins thereby giving signature Y.

(c) Normalize

Compute probability densities for signature X and signature Y respectively by dividing the number of elements in each container by the corresponding sums of all elements of signature X and signature Y.

Thus, 10 features are extracted from the image as probability densities for Signature 2.

Finally, the sum of Signature 1 and signature 2 gives a total number of 31 Probabilistic features (PF) of the binary image A, shown in Figure 4.2 (a). Similarly, Signature 1 and Signature 2 for the binary image B, shown in Figure 4.2 (b) are also computed.

Experiments were conducted to compare results for two digits by using kl-divergence based on the signatures as p(x) of the image A from the training set as a measure of probabilities with the p(y) of the image B of the second digit from the test set.

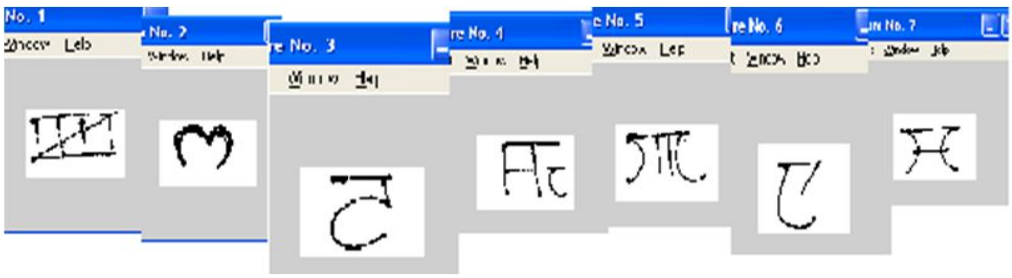


Figure 4.3: Some handwritten characters used for training set

Figure 4.4: Some random characters used for testing

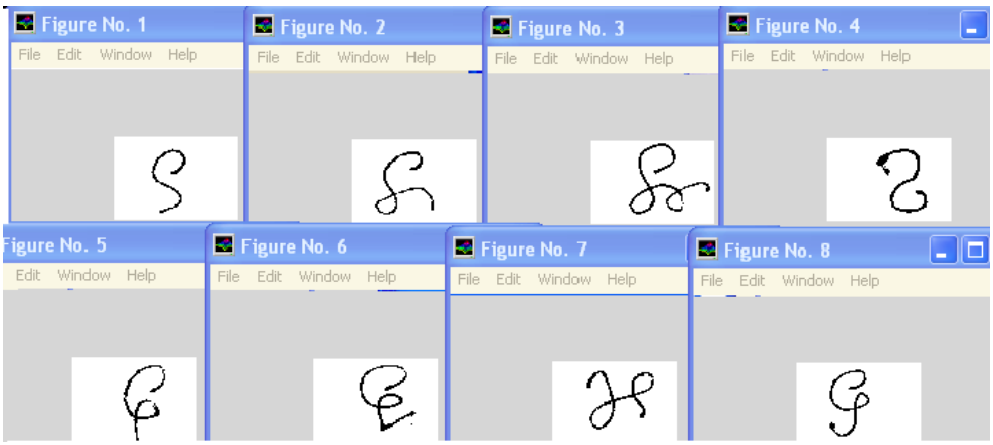


Figure 4.5: Some digits used for testing

Table 4.1 shows that divergence is 0 for the same pattern. Figure 4.5 shows some random digits used for testing from the test set.

Table 4.1 Training Set Vs Training Set

	1	2	3	4	5	6	7	8	9
1	0.0000	-0.4744	-0.4047	-0.4548	-0.3778	-0.4255	-0.2670	-0.7360	-0.5065
2	-0.2784	0.0000	-0.3498	-0.2650	-0.3538	-0.2988	-0.2012	-1.1012	-0.5754
3	-0.5048	-0.5663	0.0000	-0.6167	-0.4636	-0.2697	-0.3865	-0.6087	-0.3016
4	-0.5317	-0.5083	-0.7084	0.0000	-0.6340	-0.3000	-0.1762	-2.0626	-0.9596
5	-0.4888	-0.7257	-0.4706	-0.8031	0.0000	-0.5656	-0.3849	-1.1487	-0.6103
6	-0.4358	-0.6957	-0.4490	-0.3295	-0.5377	0.0000	-0.2316	-1.7469	-0.7822
7	-0.3582	-0.5760	-0.5065	-0.2535	-0.3378	-0.2172	0.0000	-1.5491	-0.7049
8	-0.3052	-0.7061	-0.3554	-0.6998	-0.5431	-0.4033	-0.3979	0.0000	-0.2952
9	-0.4628	-0.9464	-0.2928	-0.5088	-0.4072	-0.2166	-0.3174	-1.0024	0.0000

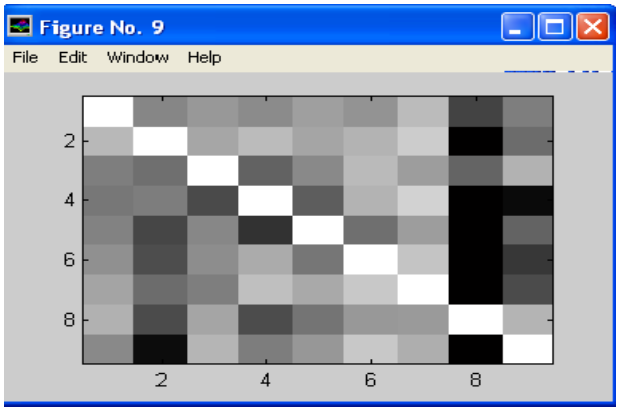


Figure 4.6: Colormap for 10 Digit

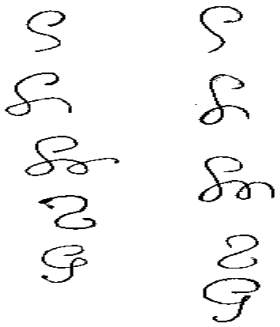


Figure 4.7: Visual Comparison of the matches of Digits from Training set (left) against Test set (right)

Colormap is a representation of the table 4.1 as an image. White being 0, black being negative values. The lighter the color, the closer the match. We can see "bands" of black when the majority of rigid digits are compared to those with curves. This is because of spiked frequencies in Signature 1 (a circle's edge is equidistant to its center). Table 4.2 shows the digit pattern matched with smallest divergence value.

Experiments were conducted to compute K-L divergence of the digits and characters from the training set against the same set for analyzing the results as shown in Figure 4.6 for 10 digits and Figure 4.8 for 27 characters. The results show the correct recognition of the digits and characters.

Table 4.2 Test results (recognized digits are highlighted) for random digits 1,2,3,4 and 8 against the training set

Test for digit 1	Test for digit 2	Test for digit 3	Test for digit 4	Test for digit 8
1 = -0.2816	-0.3462	-0.2637	-0.5397	-0.3360
2 = -0.4925	-0.2262	-0.1828	-0.4430	-0.3497
3 = -0.5682	-0.2305	-0.1278	-0.6045	-0.5832
4 = -0.8975	-0.2626	-0.3558	-0.2019	-0.6389
5 = -0.6059	-0.4917	-0.3441	-0.7335	-1.0054
6 = -0.6800	-0.2395	-0.1673	-0.4055	-0.5250
7 = -0.6612	-0.2167	-0.2165	-0.2488	-0.5403
8 = -0.5096	-0.5700	-0.4346	-0.8780	-0.3090
9 = -0.7370	-0.2879	-0.2294	-0.5961	-0.8143

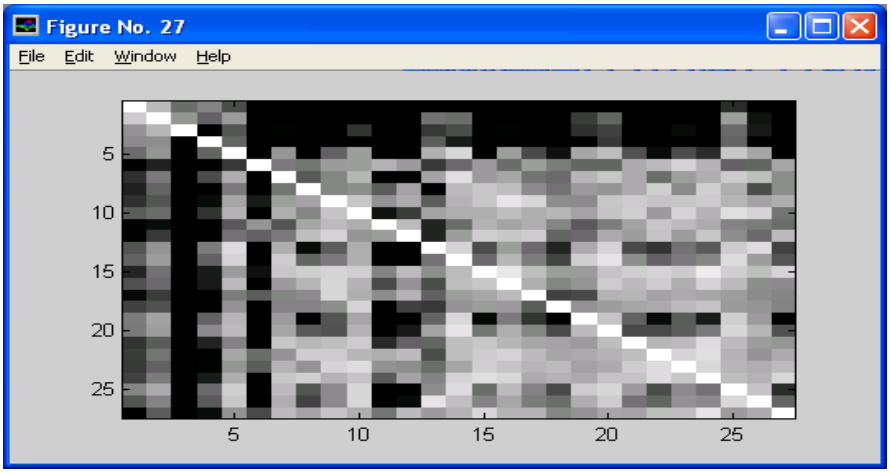


Figure 4.8: Colormap for 27 basic characters (Training set Vs Training set)

Table 4.3 Test results for random characters 6,5,7 and 9 against the training set

Test for char 6	Test for char 5	Test for char 7	Test for char 9
1 = -5.0933	-0.2260	-0.9615	-1.5448
2 = -3.3312	-0.4655	-0.3009	-0.4895
3 = -2.5448	-0.8594	-0.4818	-0.3900
4 = -4.5869	-0.3970	-0.2846	-0.4578
5 = -2.8715	**-0.0863**	-0.8011	-1.4430
6 = **-0.1331**	-0.7383	-0.4056	-0.2565
7 = -2.0179	-0.3902	**-0.1370**	-0.3951
8 = -0.8993	-0.5021	-0.3815	-0.3002
9 = -0.8066	-0.3939	-0.1796	**-0.0855**

Test results using kl-divergence on random digits and characters from a test set against the training set are shown in Table 4.2 and Table 4.3. The results were quite promising.

It is observed that Signature 1 allows the digits to be rotated because it is rotation invariant, while Signature 2 gives a decent amount of scalability to the digits because it is scale invariant.

4.1.2.3 Fuzzy Features (FF)

The feature extractor breaks down the normalized character image of 60x80 to 6x8 matrices by finding the sum of foreground values for each 10 by 10 blocks. Then compute the fuzzy values of the negative image of the character where the input range is 0 for black to 1 for white and the value in between thereby giving 48 inputs for the network. For 10 by 10 blocks, the Fuzzy membership function is (100 - Sum of foreground pixels)/100 for computing the fuzzy value ranging from 0 to 1. As shown in Figure 4.9, the fuzzy value for left-bottom 10 by 10 block is 0.3 which represents the background information of the character. The diagram shows for computing 48 fuzzy values as Fuzzy Features (FF) from a normalized digit or character.

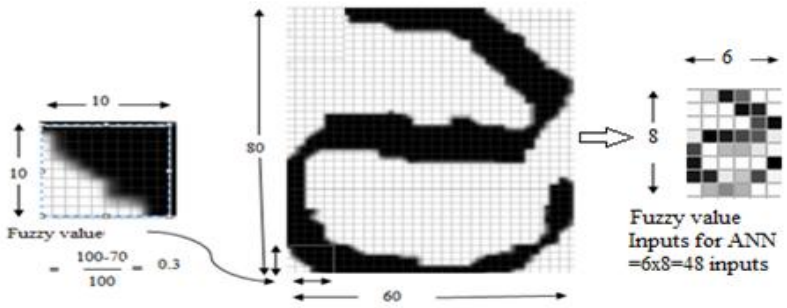


Figure 4.9: Fuzzy Features 48 Inputs of ANN

4.1.2.4 Hybrid Features (HF=PF+FF)

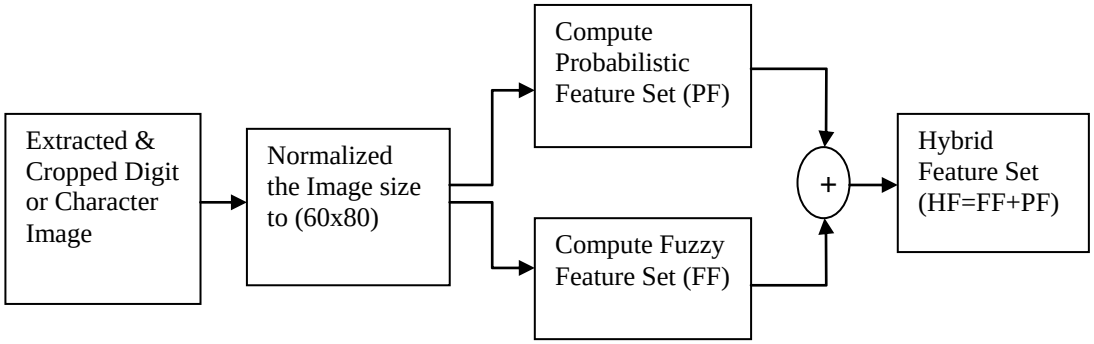


Figure 4.10: Schematic diagram of Hybrid Feature Extraction

Figure 4.10 shows the schematic diagram of the Hybrid features extraction based on Probabilistic and Fuzzy features.

The first feature is the probabilistic features set (PF) having 31 feature vectors. The second feature is fuzzy feature set (FF) having 48 feature vectors. And the combined features are the Hybrid Feature (HF) having 79 feature vectors. In order to compare these feature sets for the recognition of handwritten numerals, we conducted a series of recognition experiments based on the feature sets.

4.2 Gabor wavelets (filters) feature

Human visual system can be similarly represented by the Gabor wavelets (filters) with respect to the frequency and orientation. For texture representation and discrimination, Gabor wavelets (filters) have been found to be particularly appropriate. Many pattern analysis applications have been using Gabor filters. The most important advantage of Gabor filters is their invariance to illumination, rotation, scale, and translation. Furthermore, they can withstand photometric disturbances, such as illumination changes and image noise.

A two-dimensional Gabor filter is a Gaussian kernel function modulated by a complex sinusoidal plane wave, in the spatial domain and is defined as:

$$G(x, y) = \frac{f^2}{\pi\gamma\eta} exp(\frac{x'^2+\gamma^2 y'^2}{2\sigma^2}) exp\ (j2\pi f x' + \phi) \text{ --------- (4.2)}$$

$$x' = x\cos\theta + y\sin\theta$$

$$y' = -x\sin\theta + y\cos\theta$$

Where f is the frequency of the sinusoidal factor,

θ the orientation of the normal to the parallel stripes of a Gabor function,

ϕ is the phase offset,

σ is the standard deviation of the Gaussian envelope

and γ is the spatial aspect ratio which specifies the ellipticity of the support of the Gabor function.

The algorithm employs forty Gabor filters in 5(five) scales and 8(eight) orientations as shown in Figure 4.11 Magnitudes of Gabor filters and in Figure 4.12 , Real parts of GaborFilters [212].

For genraring the Gabor filter bank, array of u by v, whose elements are m by n matries; each matrix being a 2-D Gabor filter.

u : No. of scales (set to 5)

v : No. of orientations (set to 8)

m : No. of rows in a 2-D Gabor filter (an odd integer number set to 39)

n : No. of columns in a 2-D Gabor filter (an odd integer number set to 39)

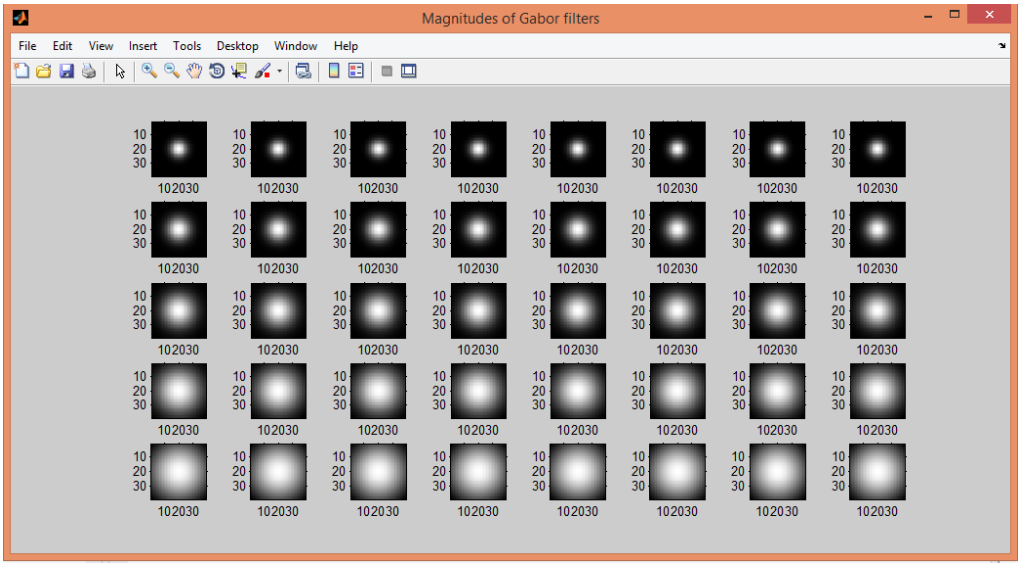


Figure 4.11 Forty (5x8) Magnitudes of Gabor filters, 2-D Gabor filter size of 39x39.

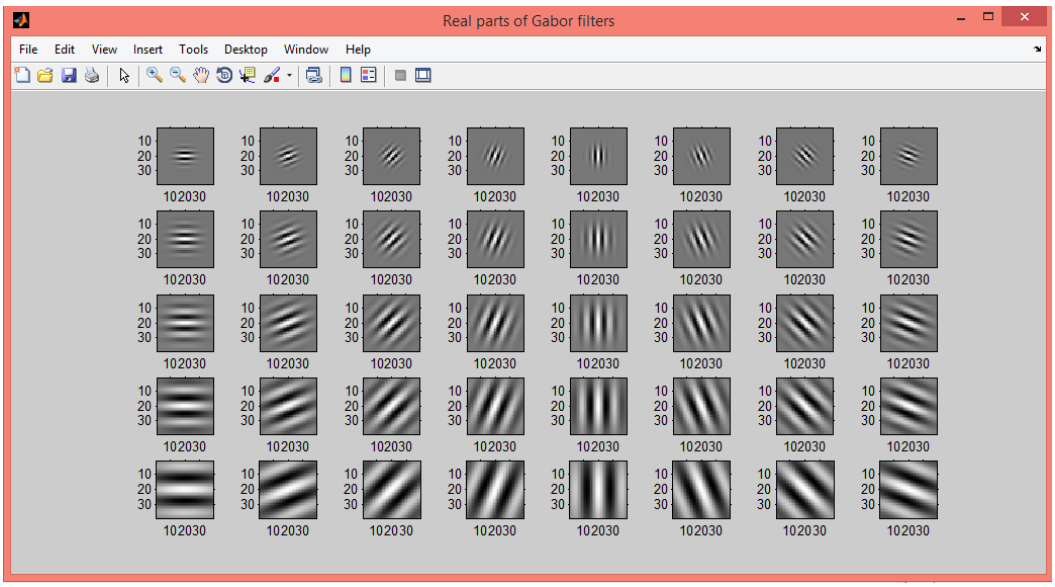


Figure 4.12 Forty (5x8) Real parts of Gabor filters

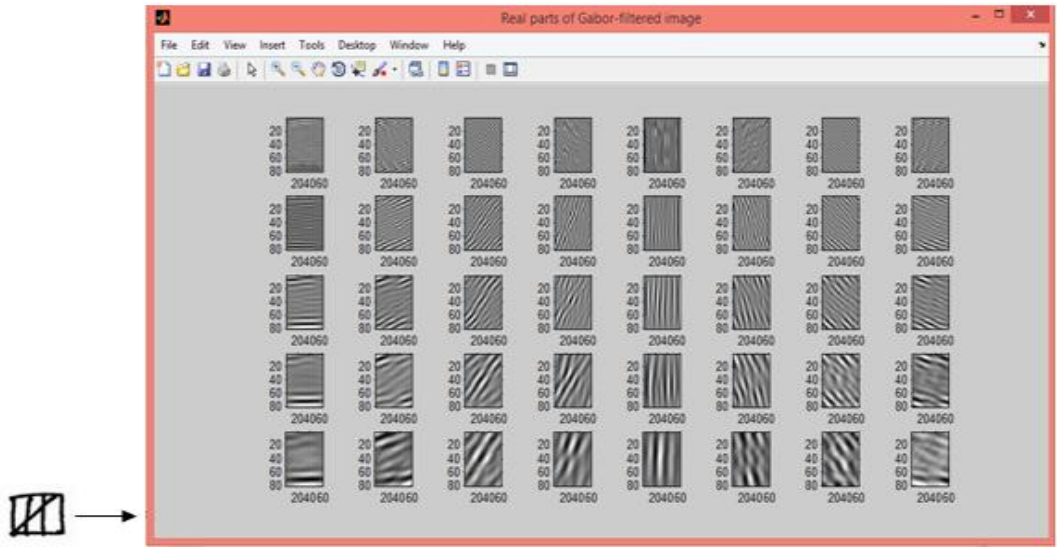


Figure 4.13(a) Applying Gabor filters on the character ह image size of 80x60

Figure 4.13(b) Forty (5x8) Real parts of Gabor Filtered character image ह with filtered size of 80x60.

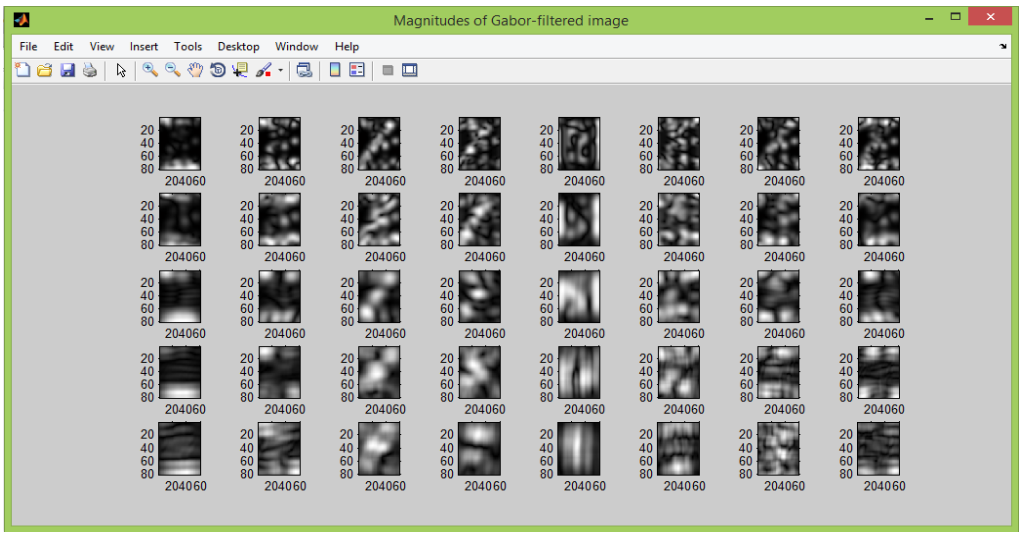


Figure 4.14 Forty (5x8) Magnitudes (complex modulus) of Gabor-Filtered image of character image with filtered size of 80x60.

After applying two dimensional convolution with Gabor filters on the segmented character image, it returns the central part of the convolution that is the same size as the image i.e., 80x60. The results of the real parts and magnitudes of the resulting image are shown in Figure 4.13 and Figure 4.14.

It is observed that the adjacent pixels in the image are highly correlated, so we can remove the information redundancy by downsampling the feature of the resulting images from Gabor filters. Gabor filters extracts the variations in different frequencies and orientations in the character.

Here, the size of the output feature vector

$$= \frac{\text{size of the image } (80\times60) \text{ x number of scales and orientations } (5\times8)}{\text{row and column downsampling factors } (4\times3)}$$

$$= \frac{80\times60 \times 5\times8}{4\times3} = 16000$$

Even after downsampling the feature, the feature vector is still very large. Feature dimensionality reduction is therefore, required for retaining informative features and eliminating redundant ones thereby giving feature

vector with reduced number of dimension. The experimental results are discussed more in detail in next Chapter 5, Section 5.2 .11 Performance Observations of Hybrid (PF+FF) features and Gabor wavelets features:
Gabor wavelet features from 16000 are reduced to 22 features after finding no. of dimensions with dimension reduction techniques using 'MLE' and mapping an output with 'PCA'.

4.3 Conclusion

In this chapter, proposed hybrid feature extraction technique is presented. Recognition using K-L divergence with probabilistic features of digits and characters are investigated and the results are discussed and presented. Fuzzy features are presented and Hybrid features are proposed for better recognition using MLP classifier. And the reduced 22 Gabor wavelet features are highlighted for classification.

Chapter 5

Performance Analysis of the Proposed Hybrid Features and Neural Networks Classifiers of the Recognition System

The research proposed in this thesis, focuses on the handwritten character recognition of Manipuri script. As mentioned previously, our research is the first report in this area for Manipuri Script but there are many reports already appeared for handwritten characters and numerals of some Indian languages. Nevertheless it is an extremely attractive and challenging area. Attractive because of the numerous applications that a handwritten character recognition system may be applied to. This recognition system presented in the thesis is based on the papers published and presented in the national conferences and workshops and international journals while the research are progressing. The reason for the area still remains challenging relates to the need for the development of higher accuracy, feature selection schemes and more carefully selected techniques for recognition. Some of the above issues are dealt with in the current proposal.

5.1 Characters of Manipuri Script

There are 56 character classes of Manipuri Script. It consists of 27 consonants (27 alphabets of Iyek Ipee), final consonants 8 Lonsum Iyek, 8 vowels (Cheitap Iyek letters), 3 khudam Iyek (Cheikhei, Lum, and Apun) and 10 numerals.

Sample data are acquired from 16 persons who are students and faculty members of the Department and are sampled on white paper pages. Each person has written the same 50 samples of the said character symbol in a page. These pages are scanned with 300 dpi. Some pages are in black and white gray scale format and some pages are in color (RGB) format. The two pages written by two persons having 50 samples each of the same character symbol are put in a single page. The data set has 5400 characters for 10 pages corresponding to 54 class characters (without the Lum symbol and Ee Lonsum symbol equivalent to Ee) of the script. In addition, there are 16

pages of 50 samples for digits (800 samples) and also 16 pages of 50 samples for 10 class mapum mayek characters (800 samples) . So, there are 7000 samples for digits and characters in total for the 54 classes.

Input Image Page containing deskewed text lines of digits or characters is preprocessed and the lines are segmented using segmentation program based on the algorithm from Section 3.1.1, Section 3.1.2. From the extracted text line, the digits or characters are segmented, cropped and then normalised the same to 60x80 using bicubic interpolation method.

For each digit or character of the Training Set as well as Test Set, the Hybrid Features (HF=PF+FF) as mentioned in Section 4.1.3.1 Probabilistic Features (PF), Section 4.1.3.2 Fuzzy Features (FF) Section 4.1.3.3 and Hybrid Features (HF=PF+FF) are extracted for training and testing. The additional feature vectors 4.1.3.4 Gabor wavelets (filters) feature after dimension reduction are also presented. Overview of the method is shown in Figure 5.1

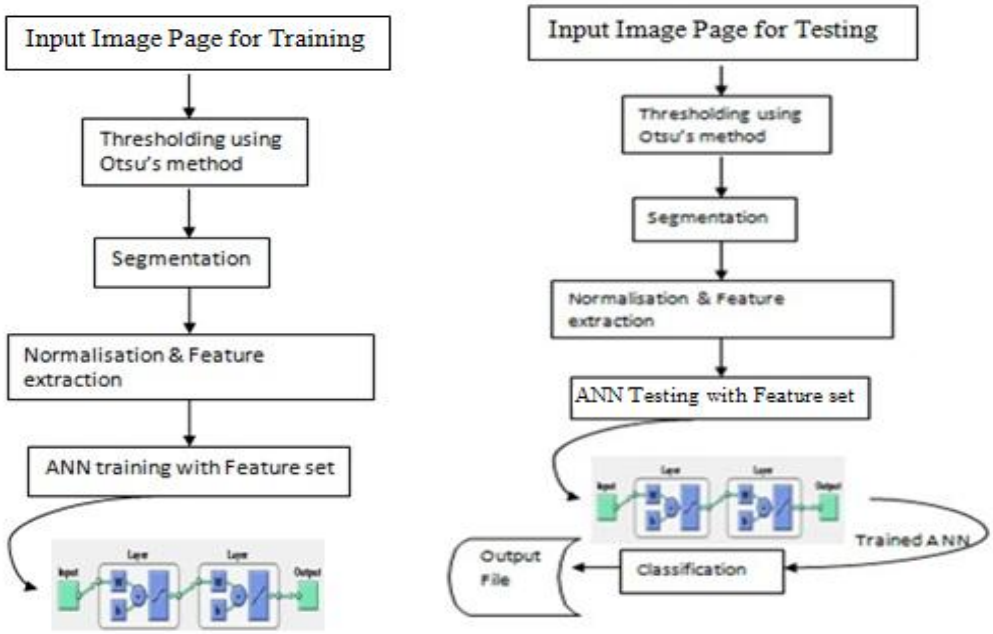


Figure 5.1: Overview of the method

5.2 Classification using MLPs

The total 54 character data classes (without the Lum symbol and Ee Lonsum equivalent to Ee) of Manipuri Script are divided into the following categories for training with 5 different Feed Forward Multilayer Perceptrons (MLPs) of backpropagation algorithm:

MLP1(A):

10 Numerals or Digits classification (Cheising Iyek):

10 classes (1, 2, 3...9,0)

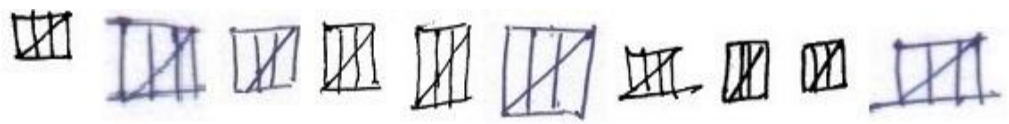
MLP2(B):

27 Characters (27 alphabets of Iyek Ipee) + 7 Lonsum Iyek:

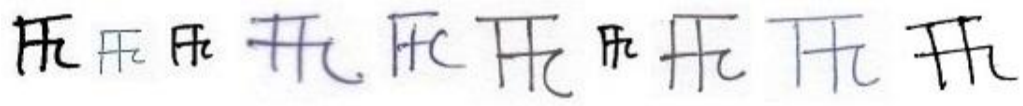
34 classes (1,2,3....34)

(Ee Lonsum of Lonsum Iyek is dropped as it is equivalent to Ee (I,E) of 27 alphabets Iyek Ipee.)

Kok→



Mit→



Ngou→

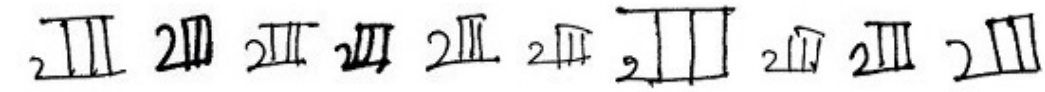


Figure 5.2: Handwritten Samples of , and characters written by different persons.

MLP3(B1):

Only 27 Characters (27 alphabets of Iyek Ipee)
without 7 Lonsum Iyek characters:

27 classes (1,2,3,...27)

MLP4(C): 8 vowels (Cheitap Iyek letters) + 2 khudam Iyek
(Cheikhei symbol for Fullstop, + Apun symbol for Sign of Ligature):

10 classes (1, 2, 3...10)

MLP5(D):

The total 44 character classes of Manipuri Script without the 10 numerals:

44 classes (1, 2, 3...44)

Five Multi Layer perceptrons (MLPs) neural networks are taken for training for each category of classes with different training sets and Test sets for the purpose of performance evaluation. And also using the feature vectors such as PF, FF, HF and Gabor wavelet features, the MLP neural networks are trained with the training sets and tested with the Test sets in order to evaluate the performance of the MLP classifiers. Several performance metrics have been used as given below:

5.2 .1 Confusion matrix

A confusion matrix contains information about actual class and predicted class given by a classification system. It is a specific table layout that allows visualization of the performance of an algorithm. Each column of the matrix represents the instances in a predicted class, which each row represents the instances in an actual class. The name stems from the fact that it makes it easy to see if the system is confusing two classes (i.e., commonly mislabeling one as another). The Table 5.1 shows the confusion matrix for a two class classifier.

		Predicted	
		Positive	Negative
Actual	Positive	TP	FN
	Negative	FP	TN

Table 5.1: Confusion Matrix

In the Table 5.1: Confusion Matrix,

True Positive (TP): Number of correct predictions that an instance is positive. A true positive instance is an actual positive sample classified correctly as positive by the classification system.

False Negative (FN): Number of incorrect predictions that an instance is negative. It is the case of a negative sample classified as belonging to the positive class, i.e., false negative.

False Positive (FP): Number of incorrect predictions that an instance is positive. A positive sample mistakenly classified as negative, named false positive.

True Negative (TN): Number of correct predictions that an instance is negative. A true negative instance is the case of a negative sample classified correctly as negative.

The following performance matrices can be calculated from the confusion matrix:

1. Accuracy:

Accuracy is a statistical measure of how well a binary classification test correctly identifies or excludes a condition. The accuracy of a classifier is given by

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+FN+TN} \quad (5.1)$$

2. Precision:

   Precision is a statistical measure that gives the number of correct positive predictions.

$$\text{Precision} = \frac{TP}{TP+FP} \quad (5.2)$$

3. Recall:

   Recall is a statistical measure that gives the number of positive examples that could be catched.

$$\text{Recall} = \frac{TP}{TP+FN} \quad (5.3)$$

4. F-Measure:

   F-Measure (F-score) is the harmonic mean of precision and recall.

$$\text{F-measure} = 2.\frac{Precision\ .\ Recall}{Precision\ +Recall} \quad (5.4)$$

5.2 .2 Classification of MLP (A)

10 Numerals or Digits classification (Cheising Iyek):

10 classes (1, 2, 3...9,0)

The 10 symbols are shown below in Figure 5.3:

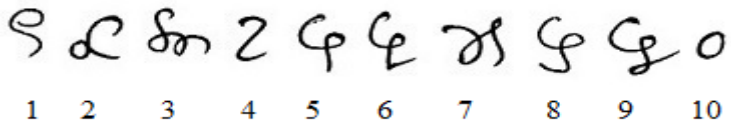


Figure 5.3: Ten (10) numerals / Digits of Manipuri Script(Cheising mayek)

In this section, research methodology of the digit recognition of our proposed system using the proposed hybrid feature set is being discussed.

Taret→

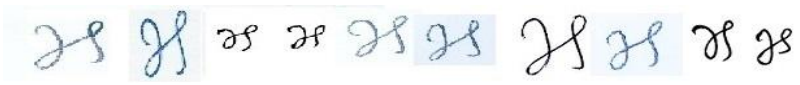


Figure 5.4: Ten different handwritten numerals 7 of 10 digits.

Three-Way data splits:

If model selection and true error estimates are to be computed simultaneously, the data should be divided into three disjoint sets [Ripley, 1996]

Training set: used for learning, e.g., to fit the parameters of the classifier.

In an MLP, we would use the training set to find the "optimal" weights with the back-propagation algorithm.

Validation set: used to select among several trained classifiers

In an MLP, we would use the Validation set to find the "optimal" number of hidden units or determine a stopping point for the back-propagation algorithm.

Test set: used only to assess the performance of a fully-trained classifier.

In an MLP, the Test set is to estimate the error rate after we have chosen the final model (MLP size and actual weights).

The error rate of the final model on validation data will be biased (smaller than the true error rate) since the validation set is used to select the final model. After assessing the final model on the test set, the model is not tuned any further.

Outline of the Procedure:

1) Divide the available data into training, validation and test set
2) Select architecture and training parameters
3) Train the model using the training set
4) Evaluate the model using the validation set
5) Repeat steps 2 through 4 using different architectures and training parameters
6) Select the best model and train it using data from the training and Validation sets
7) Assess this final model using the test set

Steps 3 and 4 have to be repeated for each of the K folds in case of Boostrap.

5.2 .3 Performances of Feature Selection methods

Performance of the network with different feature selection methods are given in following Table 5.2.

Feature Selection methods	No. of Features used	No. of Samples	No. of Training Samples	No. of Test Samples	No. of validation Samples	Recognition Rate
PF	31	250	150	50	50	90.00%
		500	300	100	100	93.00%
		600	360	120	120	93.33%
FF	48	250	150	50	50	100.00%
		500	300	100	100	95.00%
		600	360	120	120	97.50%
PF+FF Hybrid	79	250	150	50	50	98.00%
		500	300	100	100	98.00%
		600	360	120	120	97.50%
PF+FF Hybrid	79	750	450	150	150	**97.3%**
PF+FF Hybrid	79	1000	600	200	200	94.0%

Table 5.2: Feature Selection methods with Recognition rates

The MLP is the gradient Descent backpropogation with adaptive learning rate, 1 hidden Layer with 10 nodes, transfer functions for both layers_='logsig', Data division is random and maximum epoch is set with 5000.

It is observed from the Table 5.2 that the result varies according to the number of samples taken. The recognition rate of PF with a total of 31 features is the lowest among the three methods. Even though the features of FF with a total of 48 features and Hybrid (PF+FF) with a total of 79 features have significant differences, the performance of both methods seems to be almost equal to each other. But the most important fact that we observed is that the RR value of Hybrid (PF+FF) is almost consistent no matter how many samples taken, on the contrary the RR values of FF seems to fluctuate according to the change in the number of samples. So, the increased in the feature makes the Hybrid (PF+FF) feature selection method more *robust, efficient and reliable* than the remaining two methods.

5.2 .4 The Bootstrap

The bootstrap [67] [69] is a resembling technique with replacement. From a dataset of N samples, a fraction of them are randomly selected and are used for training as training set. The remaining examples that are not selected for training are used for testing. The same process is repeated for a specified number of K folds. True error is estimated as the average error rate on test data. For the said experiment, only 5 folds are taken as shown in Table 5.3.

Complete Dataset for Digits: {P1+P2+P3+P4+P5} =1000 Samples

Experiment Number	Training Set	Test Set
Experiment 1(Run1):	P2+P3+P4+P5	P1
Experiment 2(Run2):	P2+ P3+P4	P1+P5
Experiment 3(Run3):	P1+P2 +P3	P4 +P5
Experiment 4(Run4):	P1+ P4+P5	P2+P3
Experiment 5(Run5):	P1+ P2+ P3+ P4	P5

Table 5.3: Datasets of 5 folds for Digits

Off-line handwritten one thousand (1000) numerals written by two persons are used for the experimental purposes. Each person has written the same 50 symbols of the said numeral symbol. So there are 100 samples for each numeral symbol written by two persons in two pages as shown in Figure 5.6. These two pages are put in a single page for a particular digit for processing. So, there are 10 pages corresponding to 10 digits. The complete dataset is divided into 5 folds (k=5). There are 200 samples (20 samples x 10 pages) in a fold. These 200 samples in each fold are grouped by taking 20 samples for each digit from the corresponding 10 numeral pages. An example of a sample page for digit 3 is shown below:

Figure 5.5: A sample page having 100 samples for digit 3 written by two persons

5.2.5 ANN Classifier MLP1(A)

Neural network used for classification is the Feedforward Multilayer Perceptron (MLP) Backpropogation algorithm with Gradient descent network training function that updates weight and bias values according to gradient descent momentum and an adaptive learning rate. The network gets training as long as its weight, net input, and transfer functions have derivative functions. Backpropagation is used to calculate derivatives of performance with respect to the weight and bias variables. Each variable is adjusted according to the gradient descent with momentum. Sum of squared errors (sse) is a network performance function. It measures performance according to the sum of squared errors.

There are 79 feature values for the proposed Hybrid Features. The proposed Hybrid features, HF (79) = PF (31) + FF (48), where PF is probability features and FF is the Fuzzy features. Hybrid features extraction technique as mentioned in Figure 4.10 is performed for all the training samples. The Hybrid Features (HF) is selected from all the samples of the training set as well as the Test Set for training the network.

Experiment 1:

Training Set = {P2+P3+P4+P5} = 800 Samples

= 79 x 800 = 63200 feature values.

Test Set = {P1} = 200 Samples = 79 x 200

= 15800 feature values.

Number of input nodes is 79; number of output nodes is 10 for 10 digits. Experiment 1 is performed for observing the performance of the network for different hidden nodes varying from 10 to 50, increasing with 1 neuron for every performance test.

The parameters for the network are: network performance function is sum squared error (sse), goal is set to 0.01, number of epoch is set to 5000, sigmoidal activation functions for hidden and output layers, bias inputs for both layers weights and momentum coefficient is set to 0.95. Default value of learning rate is 0.01. Only the results of highest accuracies of the total training having different hidden units of the MLP1 (A) backpropagation network are given in the Table 5.4 below:

Hidden Nodes	Accuracy %
24	96.0
36	95.5
40	94.5
46	95.5
50	94.0

Table 5.4: Hidden Units with Accuracy% of MLP1 (A):Run1

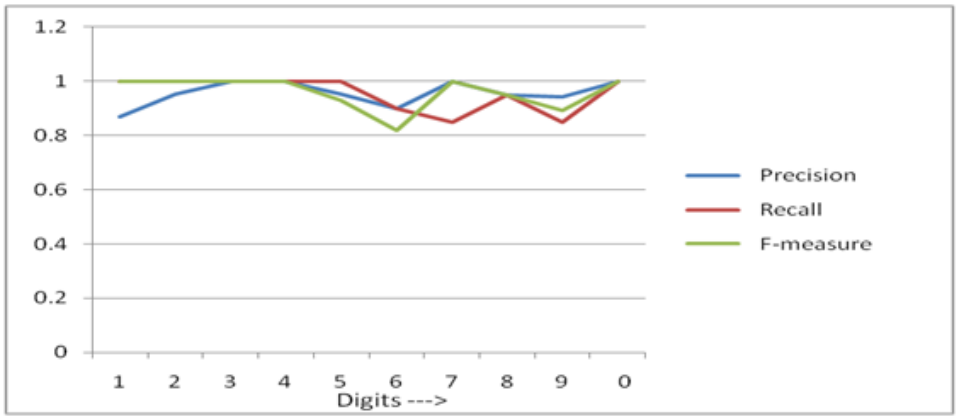


Figure 5.6: Performance of MLP1 (A) for Hidden units = 36 and Accuracy% = 95.5 %.(Run1)

Experiment 2:

Training Set = {P2+P3+P4} = 600 Samples = 79 x 600 = 47400 feature values.

Test Set = {P1+P5} = 400 Samples = 79 x 400 = 31600 feature values.

Hidden Nodes	Accuracy%
24	90.5
36	92.0
40	92.5
46	94.5
50	95.0

Table 5.5: Hidden Units with Accuracy% of MLP1 (A):Run2

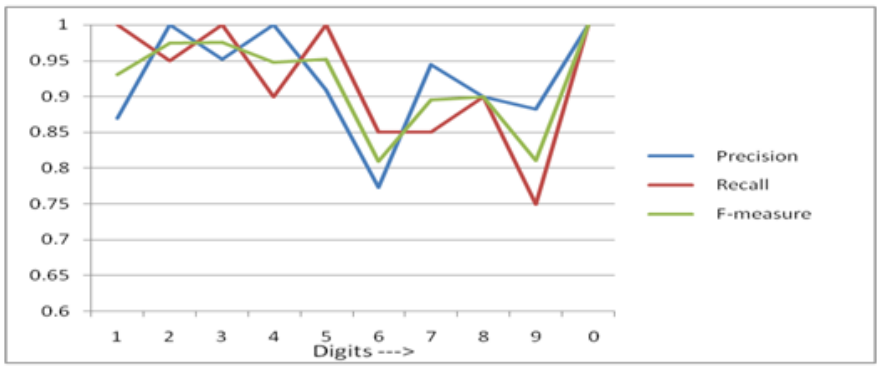


Figure 5.7: Performance of MLP1 (A) for Hidden units = 36 and Accuracy% = 92.0 %(Run2).

Experiment 3:

Training Set = {P1+ P2 +P3| = 600 Samples = 79 x 600 = 47400 feature values.

Test Set = {P4+ P5} = 400 Samples = 79 x 400 = 31600 feature values.

Hidden Nodes	Accuracy%
24	94.0
36	95.0
40	92.0
46	94.5
50	93.0

Table 5.6: Hidden Units with Accuracy% of MLP1 (A):Run3

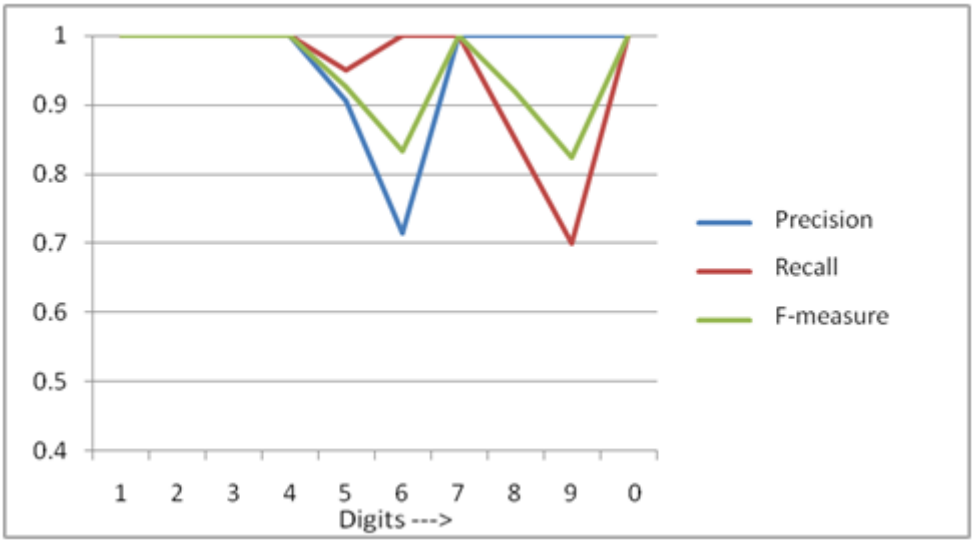


Figure 5.8: Performance of MLP1 (A) for Hidden units = 36 and Accuracy% = 95.0 %.(Run3)

Experiment 4:

Training Set = {P1+ P4+ P5} = 600 Samples = 79 x 600 = 47400 feature values.

Test Set = {P3+ P3} = 400 Samples = 79 x 400 = 31600 feature values.

Hidden Nodes	Accuracy%
24	23
36	27.5
40	24
46	25
50	23

Table 5.7: Hidden Units with Accuracy% of MLP1 (A):(Run4)

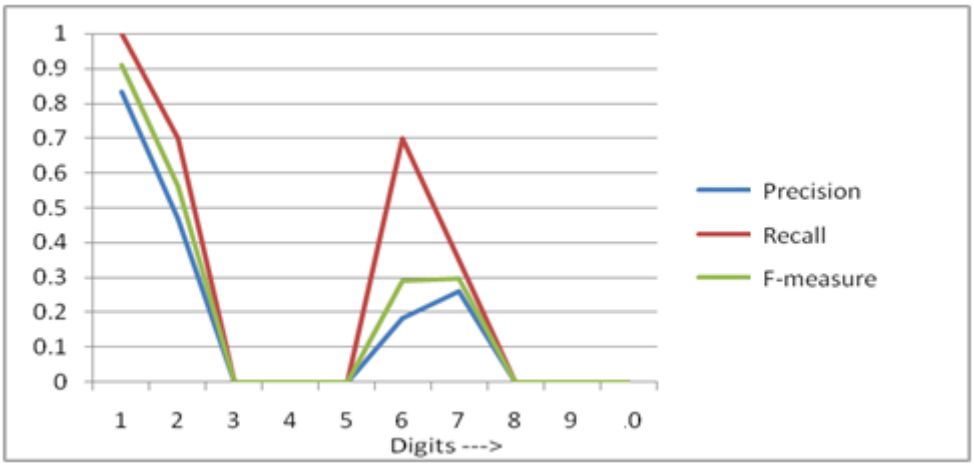


Figure 5.9: Performance of MLP1 (A) for Hidden units = 36 and Accuracy% = 27.5%.(Run4)

Experiment 5:

Training Set = {P1+ P2+ P3+P4} = 800 Samples

= 79 x 800 =63200 feature values.

Test Set = {P5}= 200 Samples = 79 x 200 = 15800 feature values.

Hidden Nodes	Accuracy%
24	95.5
36	96
40	96
46	94
50	95

Table 5.8: Hidden Units with Accuracy% of MLP1 (A):Run5

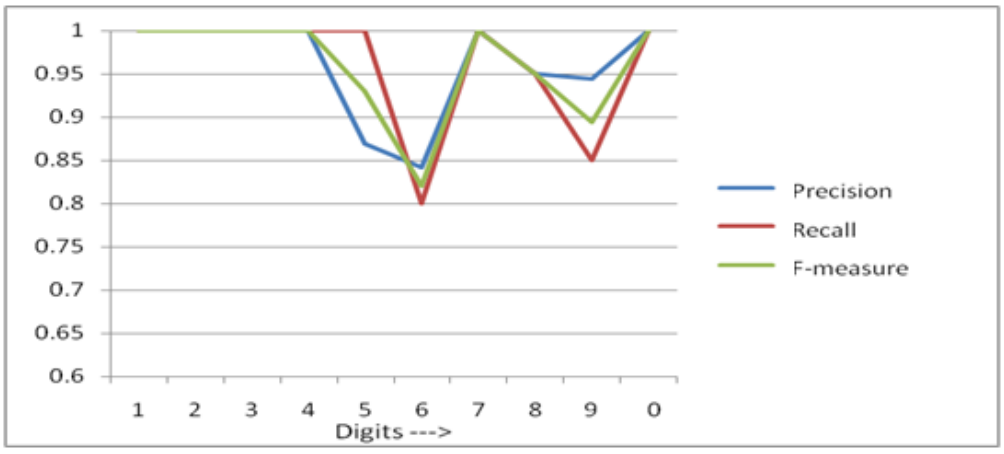


Figure 5.10: Performance of MLP1 (A) for Hidden units = 36 and Accuracy% = 96.0 %.(Run5)

Performance evaluation of the trained network with different Training sets and Test sets using Bootstrap (datasets of 5 folds for Digits given in Table 5.3) is given in Table 5.9 and it shows the recorded accuracies during

classification with hidden nodes varying from 10 to 50 by incrementing with single step.

	Run1	Run2	Run3	Run4	Run5	
Hidden Units	Accuracy%	Accuracy%	Accuracy%	Accuracy%	Accuracy %	Average Accuracy%
24	96	90.5	94	23	95.5	79.8
36	**95.5**	**92**	**95**	**27.5**	**96**	**81.2**
40	94.5	92.5	92	24	96	79.8
46	95.5	94.5	94.5	25	94	80.7
50	94	95	93	23	95	80

Table 5.9: Average Accuracy% of the Performance of MLP1 (A) for 5 fold experiments.

The Best Model selected using Bootstrap from the above experiments is MLP1 (A)(79-36-10) with Input units =79 (HF), Hidden Units = 36, Output Neurons =10 with the parameters mentioned above as the highest Average Accuracy% obtained is 81.2% for 5 Runs. The highest Accuracy% for Run5 is 96% for the model. The Bootstrap increases the variance that occurs in each fold [67][69] (Efron and Tibshirani, 1993). This is a desirable property since it is a more realistic simulation of the real-life experiment from which our dataset was obtained.

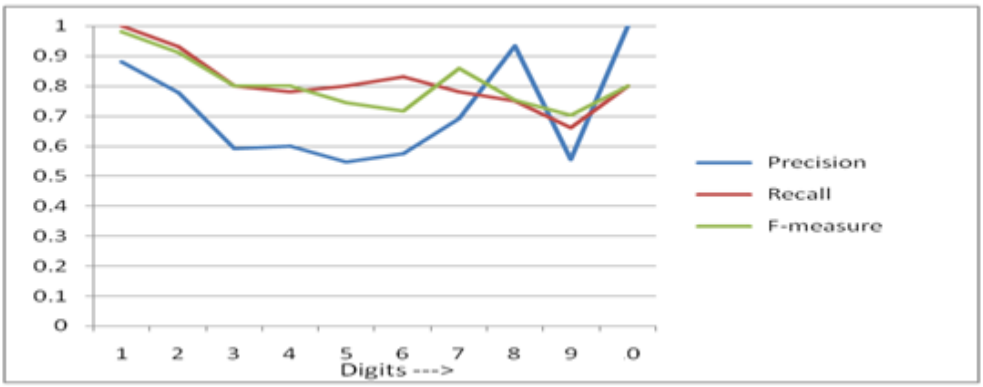


Figure 5.11: Performance of MLP1 (A) for Hidden units = 36 and Average Accuracy% = 81.2% for the selected model.

5.2 .6 ANN Classifier MLP2 (B)

27 Characters (27 alphabets of Iyek Ipee) + 7 Lonsum Iyek: 34 classes (1,2,3….34)

(Ee Lonsum of Lonsum Iyek is dropped as it is equivalent to Ee (I,E) of 27 alphabets Iyek Ipee.). The Figure 5.12 shown below is 34 character classes.

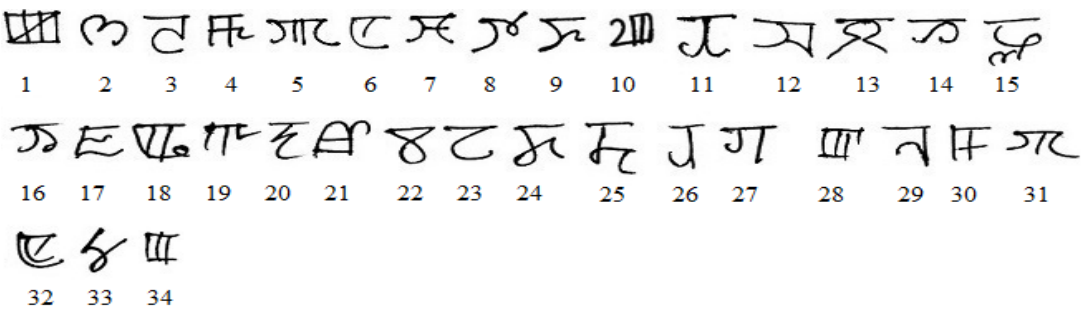


Figure 5.12: 34 characters is total of 27 Characters (27 alphabets of Iyek Ipee) and 7 Lonsum Iyek characters.

5.2 .7 Resilient Backpropagation (trainrp)

Multilayer networks typically use sigmoid transfer functions in the hidden layers. These functions are often called "squashing" functions, because they compress an infinite input range into a finite output range. Sigmoid functions are characterized by the fact that their slopes must approach zero as the input gets large. This causes a problem when we use steepest descent to train a multilayer network with sigmoid functions, because the gradient can have a very small magnitude and, therefore, cause small changes in the weights and biases, even though the weights and biases are far from their optimal values.

The purpose of the resilient backpropagation (Rprop) training algorithm is to eliminate these harmful effects of the magnitudes of the partial derivatives. Only the sign of the derivative can determine the direction of the weight update; the magnitude of the derivative has no effect on the weight update. The size of the weight change is determined by a separate update value. The update value for each weight and bias is increased by a factor delt_inc whenever the derivative of the performance function with respect to that weight has the same sign for two successive iterations. The update value is decreased by a factor delt_dec whenever the derivative with respect to that weight changes sign from the previous

iteration. If the derivative is zero, the update value remains the same. Whenever the weights are oscillating, the weight change is reduced. If the weight continues to change in the same direction for several iterations, the magnitude of the weight change increases.

Experiment for classification of 34 classes has been performed using MLP with the resilient backpropagation (Rprop) training algorithm with the following parameters:

Training Set size= 2720 samples (80% of total sample size of 3400, each character having 80 samples for the 34 labeled characters)
Test Set size= 680 samples (20% of 3400 samples, each having 20 samples for the 34 labeled characters)

MLP with Hybrid Features, Input nodes=79 units (Number of Hybrid feature values), Training the network with 155 hidden units (trial method), Output layer nodes: 34 Labeled Character Symbols, maximum epoch is set to 20,0000, Performance is Mean Squared Error(MSE), training is with Rprop(trainrp), learning rate is 0.1, momentum coefficient(mc) is 0.1, Data division is random(dividerand), Input nodes=79(number of hybrid feature values), Output layer nodes: 34 Labeled Character Symbols. And the performance is shown in the Figure 5.13.

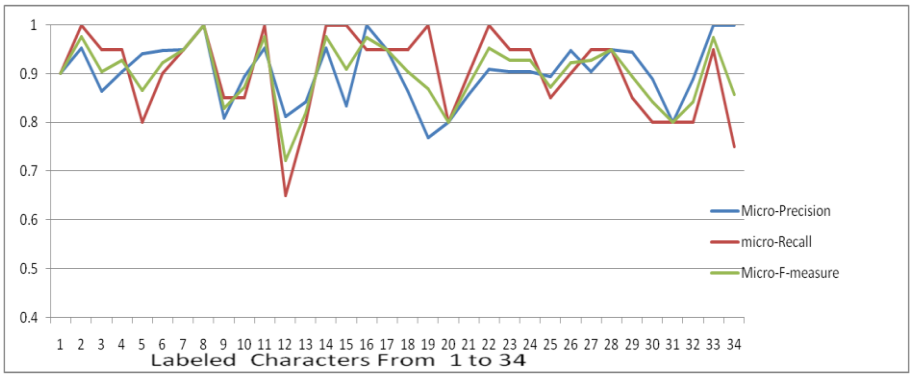


Figure 5.13: Performance of MLP2 (B) with (Rprop), Best Accuracy %: 90.1471%, Input nodes=79 nodes (Hybrid feature vectors), Hidden Layer Nodes: 155, Output layer nodes: 34 Labeled Character Symbols.

A second experiment is performed using MLP2 (B) with (traingdx) for observing the accuracies of the network for different hidden nodes varying from 30 to 176, increasing with 1 neuron for every run. The parameters for the network are: network performance function is sum squared error (sse), goal is set to 0.01, number epoch is set to 5000, and momentum coefficient is set to 0.95. Default value of learning rate is 0.01. The Accuracy%: 88.97% is obtained at Hidden Layer units =94 as shown in Figure 5.14.

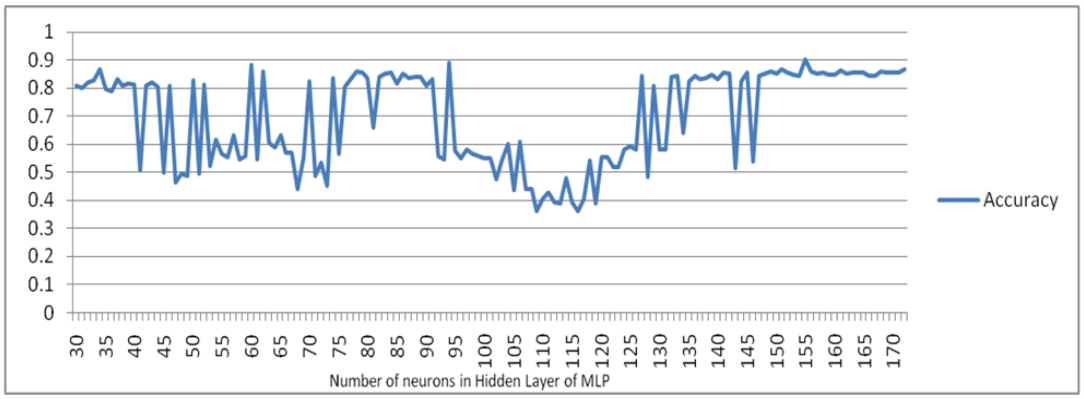


Figure 5.14: Best Accuracy %: 88.97%, MLP2 (B) with Hybrid Features, hidden layer having 94 nodes, Training Set size (80% of total sample size of 3400) is 2720 samples (34 characters, each character having 80 samples) and Test Set size of 680 samples (34 characters, each having 20 samples, 20% of 3400 samples), from hidden layer nodes from 30 to 176 with corresponding accuracies.

A third experiment is performed using MLP2 (B) with (traingdx) for observing the accuracies of the network for different hidden nodes varying from 60 to 300, increasing with 1 neuron for every run. The parameters for the network are: network performance function is sum squared error (sse), goal is set to 0.01, number epoch is set to 5000, and momentum coefficient is set to 0.95. Default value of learning rate is 0.01. The Accuracy%: 88.3824% is obtained at Hidden Layer units =269 as shown in Figure 5.15 below:

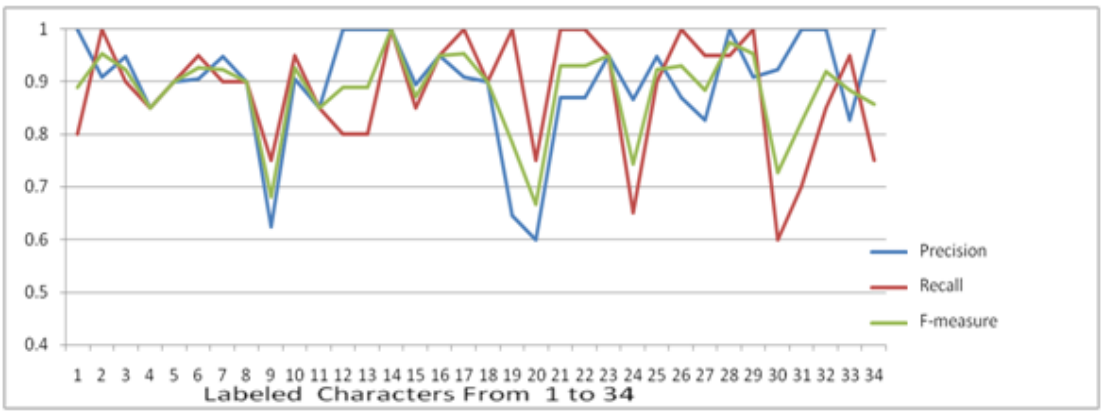


Figure 5.15: Performance of MLP2 (B) with (traingdx), Accuracy%: 88.3824%, MLP Input nodes=79 nodes (Hybrid feature vectors), Hidden Layer Nodes: 269, Output layer nodes: 34 Labeled Character Symbols.

5.2 .8 ANN Classifier MLP3 (B1)

27 Characters (27 alphabets of Iyek Ipee) without 7 Lonsum Iyek characters:

27 classes (1,2,3,...27)

Training Set = 1620 samples (60% of total sample size of 2700, each character having 60 samples for the 27 labeled characters)

Test Set1 = 270 samples (10% of 2700 samples, each having 10 samples for the 27 labeled characters)

Test Set2 = 540 samples (20% of 2700 samples, each having 20 samples for the 27 labeled characters)

Experiment 1. Training Set of 1620 samples are trained with the MLP3 (B1) which is gradient descent backpropagation with adaptive learning rate with performance set with sum squared error, mc =0.95, goal=0.01 and epoch=5000. The trained network is tested with *Test Set1* = 270 samples (10% of 2700 samples, each having 10 samples for the 27 labeled characters). The performance of Experiment 1 is shown in Figure 5.16 and Table 5.10.

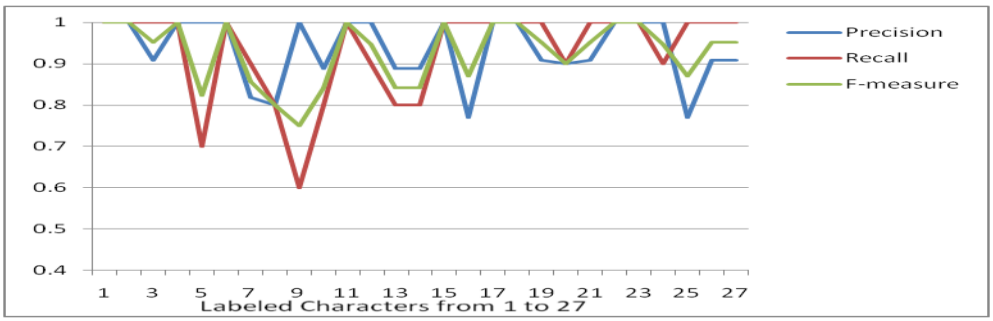


Figure 5.16: Accuracy%: 92.963 MLP (B1) with Hybrid Features, Input nodes=79 nodes (Hybrid feature vectors), Hidden Layer Nodes: 189, Output layer nodes: 27 Labeled Character Symbols as shown in the following Table 5.10 below:

Labeled Numbers of Characters	Precision	Recall	F-measure
1	1.000000	1.000000	1.000000
2	1.000000	1.000000	1.000000
3	0.909091	1.000000	0.952381
4	1.000000	1.000000	1.000000
5	1.000000	0.700000	0.823529
6	1.000000	1.000000	1.000000
7	0.818182	0.900000	0.857143
8	0.800000	0.800000	0.800000
9	1.000000	0.600000	0.750000
10	0.888889	0.800000	0.842105
11	1.000000	1.000000	1.000000
12	1.000000	0.900000	0.947368
13	0.888889	0.800000	0.842105
14	0.888889	0.800000	0.842105
15	1.000000	1.000000	1.000000
16	0.769231	1.000000	0.869565
17	1.000000	1.000000	1.000000
18	1.000000	1.000000	1.000000
19	0.909091	1.000000	0.952381
20	0.900000	0.900000	0.900000
21	0.909091	1.000000	0.952381
22	1.000000	1.000000	1.000000
23	1.000000	1.000000	1.000000
24	1.000000	0.900000	0.947368
25	0.769231	1.000000	0.869565
26	0.909091	1.000000	0.952381
27	0.909091	1.000000	0.952381

Table 5.10: Performance of MLP3 (B1)

The screenshot of the training of 27 Characters with 128 iterations with validation checks of Experiment 1 is shown in Figure 5.17.

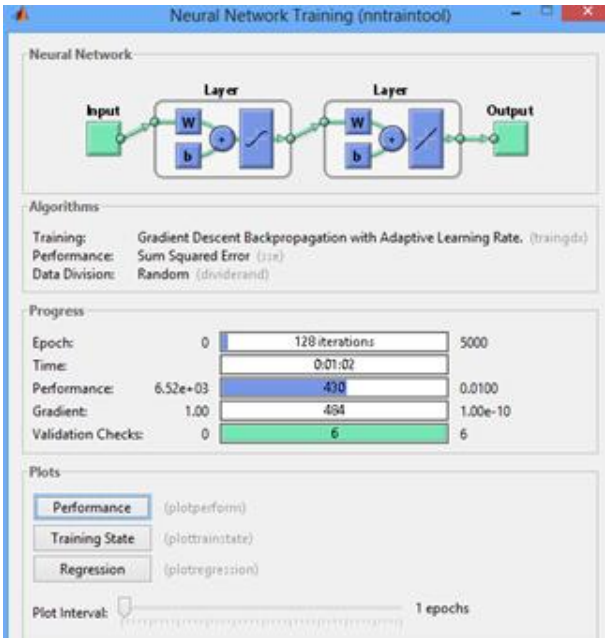


Figure 5.17: Training of 27 Characters with 128 iterations with 6 maximum validation failures, 5000 maximum number of epochs for training, .01 performance goal, 1.00e-10 minimum performance gradients.

Experiment 2. Training Set of 1620 samples are trained with the MLP3 (B1) which is gradient descent backpropagation with adaptive learning rate with performance set with sum squared error, mc =0.95, goal=0.01 and epoch=5000. The trained network is tested with *Test Set2* = 540 samples (20% of 2700 samples, each having 20 samples for the 27 labeled characters). The performance of Experiment 2 is shown in Figure 5.18.

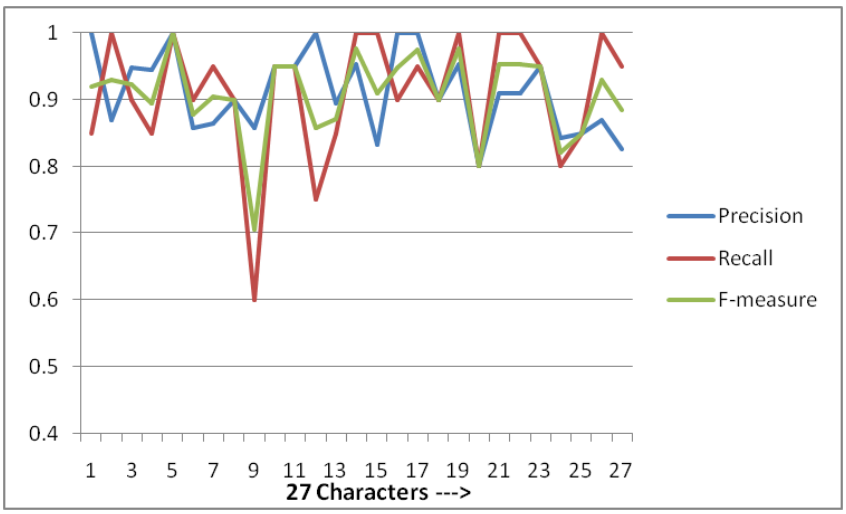


Figure 5.18: Accuracy%: 90.9259% of MLP 3(B1), Input nodes=79 nodes (Hybrid feature vectors), Hidden Layer Nodes: 189, Output layer nodes: 27 Labeled Character Symbols.

Experiment 3. Training Set of 1620 samples are trained with the MLP3 (B1) which is gradient descent backpropagation with resilient backpropagation (Rprop) training algorithm with performance set with mean squared error(mse), mc =0.1, goal=0.01 ,learning rate=0.1 and epoch=5000. The trained network is tested with *Test Set2* = 540 samples (20% of 2700 samples, each having 20 samples for the 27 labeled characters). The performance of Experiment 3 is shown in Figure 5.19.

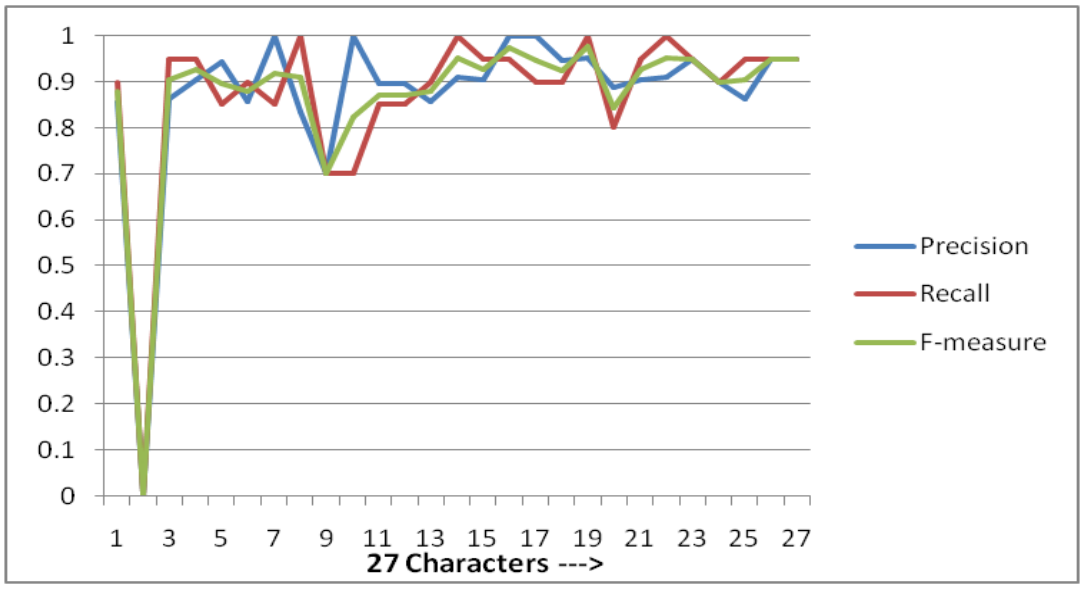


Figure 5.19: Accuracy%: 90.926% of MLP 3(B1) with Rprop, Input nodes=79 nodes (Hybrid feature vectors), Hidden Layer Nodes: 190, Output layer nodes: 27 Labeled Character Symbols.

5.2 .9 ANN Classifier MLP4(C)

8 vowels (Cheitap Iyek letters) + 2 khudam Iyek (Cheikhei symbol for Fullstop, + Apun symbol for Sign of Ligature):

10 classes (1, 2, 3...9, 0)

The Figure 5.20 shown below is 10 character classes.

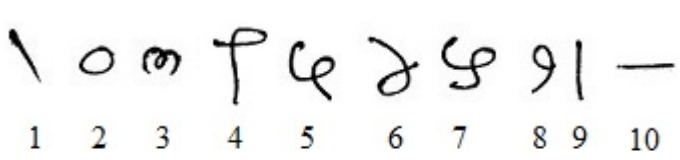


Figure 5.20: 10 characters classes = 8 vowel Characters (Cheitap Iyek letters) + 2 khudam Iyek.

A single vertical line is taken for training the neural net as the symbol of end of line (Full stop) at symbol number 9 in the list in Figure 5.20 instead of the actual two vertical parallel lines.

Training Set = 600 samples (60% of total sample size of 1000, each character having 60 samples for the 10 labeled characters)

Test Set = 400 samples (40% of 1000 samples, each having 40 samples for the 10 labeled characters)

MLP4(C) is Feedforward Multilayer Perceptron(MLP) Gradient descent Backpropogation with adaptive learning rate. The proposed Hybrid features, HF (79) = PF (31) + FF (48), are selected from all the samples of the training set (79 x 600) as well as the Test Set(79x400) for testing network. The network parameters set for the experiment are: performance is sse, biases for both layers, goal is 0.01, momentum coefficient =0.95 and epoch = 5000. Performances are recorded during the training process of the network from hidden units 10 to 50 by increasing the hidden neurons by one. The graph in Figure 5.21 shows the recorded accuracies with the corresponding hidden layer neurons. Highest accuracy% of **97.5%** is obtained when the network has **28** neurons in the hidden layer.

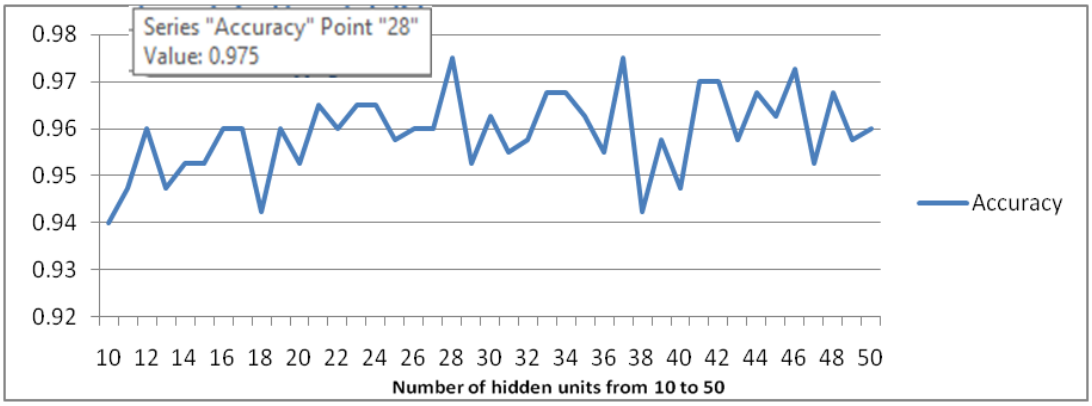


Figure 5.21: Accuracy%: 97.5% MLP4(C), Input nodes=79 nodes (Hybrid feature vectors), Hidden Layer neurons by increasing with one step size from 10 to 50, Output layer nodes: 10 Labeled Character Symbols, Accuracy%: 97.5% at 28 hidden layer units.

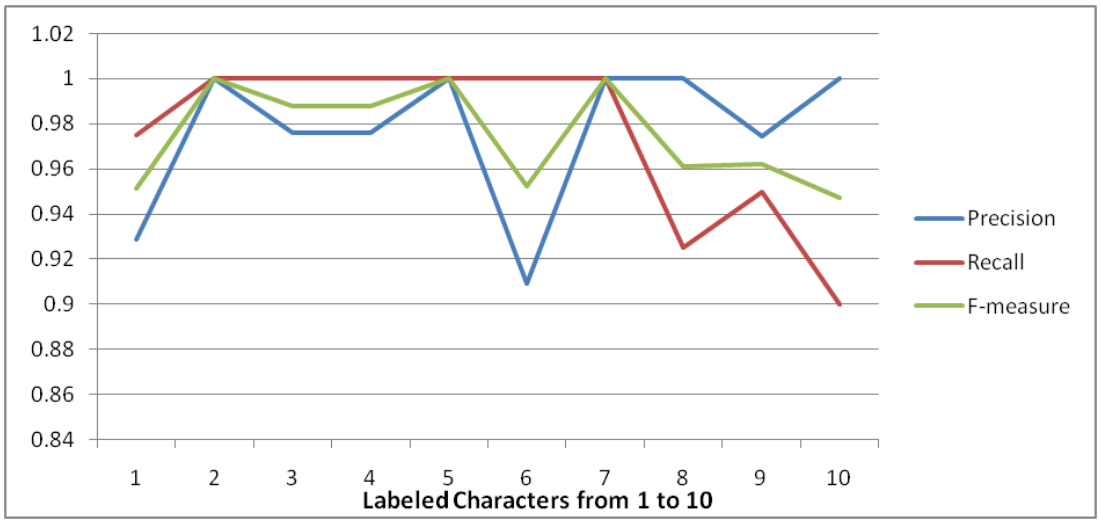


Figure 5.22: Performance of the MLP4(C), Accuracy%: 97.5%, Input nodes=79 nodes (Hybrid feature vectors), Hidden Layer neurons: 28, Output layer nodes: 10 Labeled Character Symbols.

5.2 .10 ANN Classifier MLP5 (D)

The total 44 character classes of Manipuri Script (without considering 10 numerals) are taken for the experiment:

44 classes (1, 2, 3...44)

Training Set = 2640 samples (60% of total sample size of 4400, each

Character having 60 samples for the 44 labeled characters)

Test Set = 880 samples (20% of 4400 samples, each having 20 samples for the 44 labeled characters). The proposed Hybrid features, HF (79) = PF (31) + FF (48), are selected from all the samples of the training set (79 x 2640) as well as the Test Set(79x880) for training the network.

MLP5 (D) is Gradient descent Backpropagation with adaptive learning rate and is configured with the following parameters. The proposed Hybrid features, HF (79) = PF (31) + FF (48), are selected from all the samples of the training set (79 x 2640) as well as the Test Set(79x880) for training the network. Performance is measured with respect to Mean Squared Error, Data division is random, epoch is set 1000, momentum coefficient is 0.9, learning rate coefficient is 0.1, minimum gradient is set with 1e-15, goal is 0.0 and sigmoid transfer functions are used for both hidden and output layers. Highest accuracy% obtained is 87.95% when hidden units are 409.

The performance is shown in Figure 5.23, Figure 5.24 and Figure 5.25.

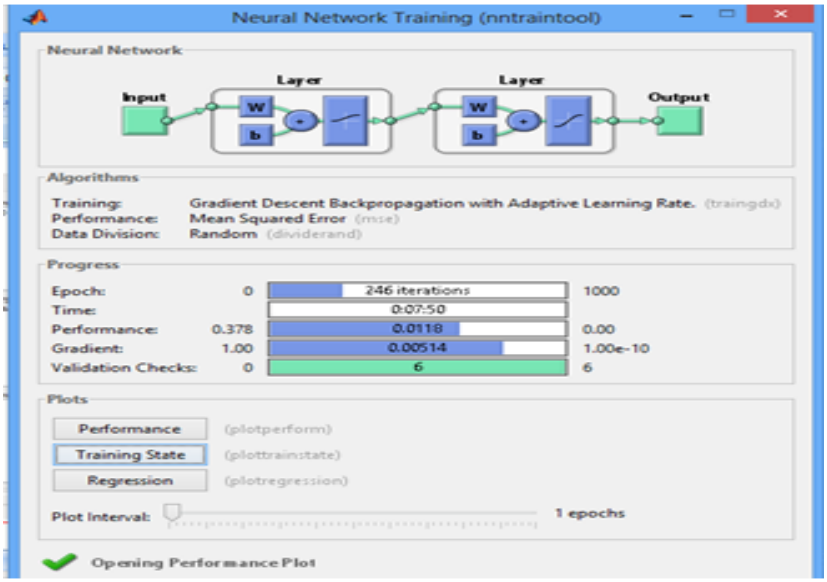


Figure 5.23: Training stops with the 6 validation checks for Accuracy% =87.95%. Training of 44 Characters with 246 iterations with 6 maximum validation failures, 1000 maximum number of epochs for training, .01 performance goal , 1.00e-10 minimum performance gradients.

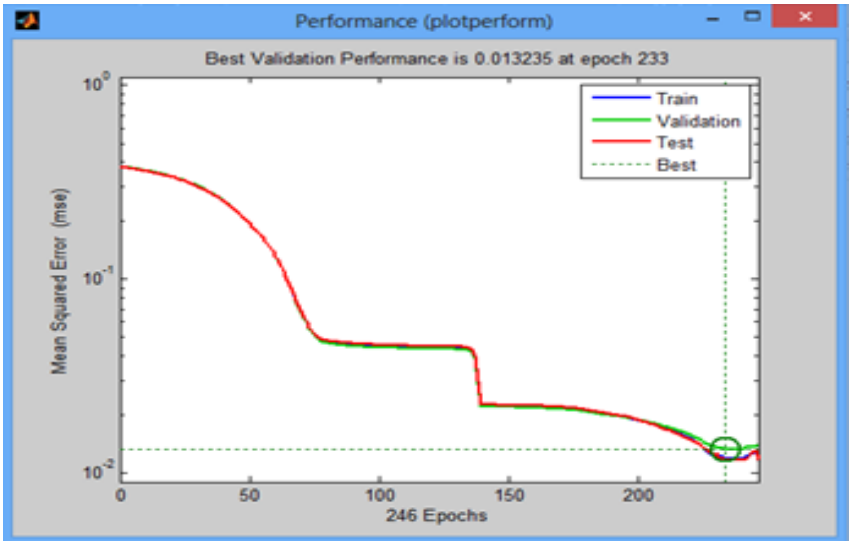


Figure 5.24: Plot of Performance (MSE) of the training.

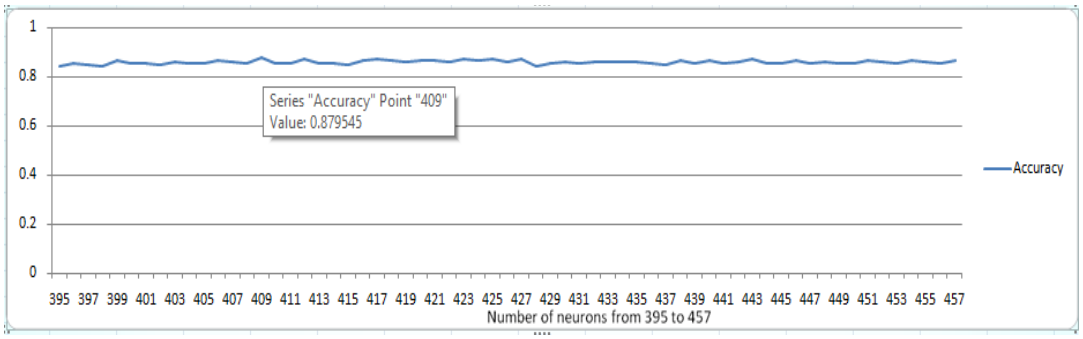


Figure 5.25: Plot of Accuracy with number of units in hidden layer from 395 to 457, highest accuracy% obtained is 87.95% when hidden units is 409.

5.2 .11 Performance Observations of Hybrid (PF+FF) features and Gabor wavelets features

The first 10 character of the total 44 character classes of Manipuri Script are taken for training under consideration for observations.

Characters are segmented and then normalised the same to 60x80 using bicubic interpolation method. Each feature set i.e., Probabilistic features (PF),Fuzzy features (FF) sets, Hybrid feature set and Gabor wavelets features are computed and tested for recognition rates as shown in Table 5.11: Performance of Feature sets: Hybrid (PF+FF) features and Gabor wavelets features.

For Hybrid feature set, the network is the gradient Descent backpropogation with adaptive learning rate, 1 hidden Layer with 10 nodes, output nodes is 10 for 10 characters, sigmoid transfer functions for both layers. The network parameters set for the experiment are: performance is sse, biases for both layers, goal is 0.01, momentum coefficient =0.95 and epoch = 5000.

There are sixteen pages written by 16 persons, each page having 50 characters. Each character class is having 5 samples per page. The training set size is 600 character samples out of total 800 samples and test set size of 200 samples.

Number of Samples = 800 samples (16 different writers)
Number of Samples per writer 50 x 16 = 800 samples (single writer)
Number of Training Samples = 600 samples (each character has 60 samples)
Number of Test Samples = 200 samples (each character has 20 samples)

Three different feature sets i.e., probabilistic features(31), fuzzy features(48) and combination of both features giving a total of 79 features(31+48) are used to train the feed forward backpropagation neural network separately for the purpose of comparative study as shown in Table 5.11. Back-propagation networks provide a very effective method for performing supervised nonlinear classification. As is common in pattern classification problems, the efficiency of the network is strongly affected by the quality of the input features used to train the network. In back-propagation networks, the number of weights in the fully connected network increases as the product of the input neurodes and the hidden neurodes plus the product of the hidden neurons and the output neurons.

The MLP used *for the* Gabor feature vectors is trained using gradient descent backpropagation with learning rate = 0.1 and no. of epoch=5000.

The classification experiments using MLP are conducted using this Gabor features on 10-class characters (from 1st class to 10th class) out of 27 alphabets of Iyek Ipee. Observations on different performances corresponding to different values of training parameters using Hybrid feature sets and Gabor wavelets features are summarized in Table 5.11.

The recognition rate for the Gabor feature vectors is 98.0 %(for mixed writers). The performance is shown in Table 5.12 and Figure 5.26: Performance of MLP4, Accuracy%: 98.0.0%, Input nodes=22 nodes (Gabor wavelet feature vectors), Hidden Layer neurons: 10, Output layer nodes: 10 Labeled Character Symbols.

Method	**No. of Features used**	**No. of Samples**	**No. of Training Samples**	**No. of Test Samples**	**Recognition Rate**
1. HF(PF+FF) Hybrid Feature	79	800 (different multiple writers)	600	200	91.5%
2. HF(PF+FF) Hybrid Feature	79	800 (Single writer)	600	200	94.0%
3. Gabor wavelet features(GF)	22	800 (different multiple writers)	600	200	**98.0%**

Table 5.11: Performance of Feature sets: Hybrid (PF+FF: 79) features and Gabor wavelets features (GF: 22)

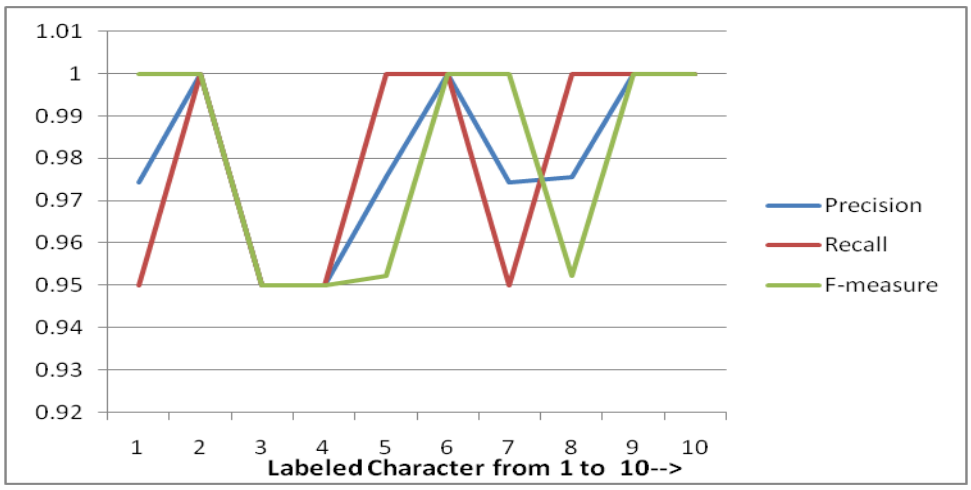


Figure 5.26: Performance of MLP4, Accuracy%: 98.0%, Input nodes=22 nodes (Gabor wavelet feature vectors), Hidden Layer neurons: 10, Output layer nodes: 10 Labeled Character Symbols.

Method 1: HF (PF+FF) Hybrid Feature

The Recognized characters in sequence corresponding to the handwritten page are:

1	2	3	4	5	6	7	8	9	0
1	2	3	4	5	6	7	8	9	0
1	2	3	4	5	6	7	8	9	0
1	2	3	4	5	6	7	8	9	0
1	2	3	4	5	6	7	8	9	0
1	2	3	4	5	6	7	8	9	0
1	2	3	4	5	6	7	8	***8***	0
1	2	3	4	5	6	***5***	8	9	1
1	2	3	4	5	***0***	7	8	9	0
1	2	3	4	5	6	7	***9***	9	0
1	2	3	4	5	6	7	8	9	0
1	2	3	4	5	6	7	8	9	0
1	2	3	4	5	6	***8***	***2***	9	0
1	2	3	4	5	6	***5***	8	***6***	0
1	2	3	4	5	6	7	8	9	1
1	2	3	4	***7***	6	7	8	9	1
1	2	3	4	5	6	7	8	9	1
1	2	***0***	4	5	6	7	8	9	1
1	2	3	***9***	5	6	7	8	9	0
1	2	***6***	***4***	5	6	7	8	9	0

HIT= Number of correctly recognized characters for the class,

MISS= Number of misrecognized characters for the class,

RR= Recognition Rate

For 1 : HIT=20 MISS=0 TOTAL=20 RR=100.00 %

For 2 : HIT=20 MISS=0 TOTAL=20 RR=100.00 %

For 3 : HIT=18 MISS=2 TOTAL=20 RR=90.00%

For 4 : HIT=19 MISS=1 TOTAL=20 RR=95.00 %

For 5 : HIT=19 MISS=1 TOTAL=20 RR=95.00 %

For 6 : HIT=19 MISS=1 TOTAL=20 RR=95.00 %

For 7 : HIT=17 MISS=3 TOTAL=20 RR=85.00%

For 8 : HIT=18 MISS=2 TOTAL=20 RR=90.00 %

For 9 : HIT=18 MISS=2 TOTAL=20 RR=90.00 %

For 10 : HIT=15 MISS=5 TOTAL=20 RR=75.00 %

The no. of classes is: 200; Recognized Rate is: 91.5%; Error Rate is: 9.289617%

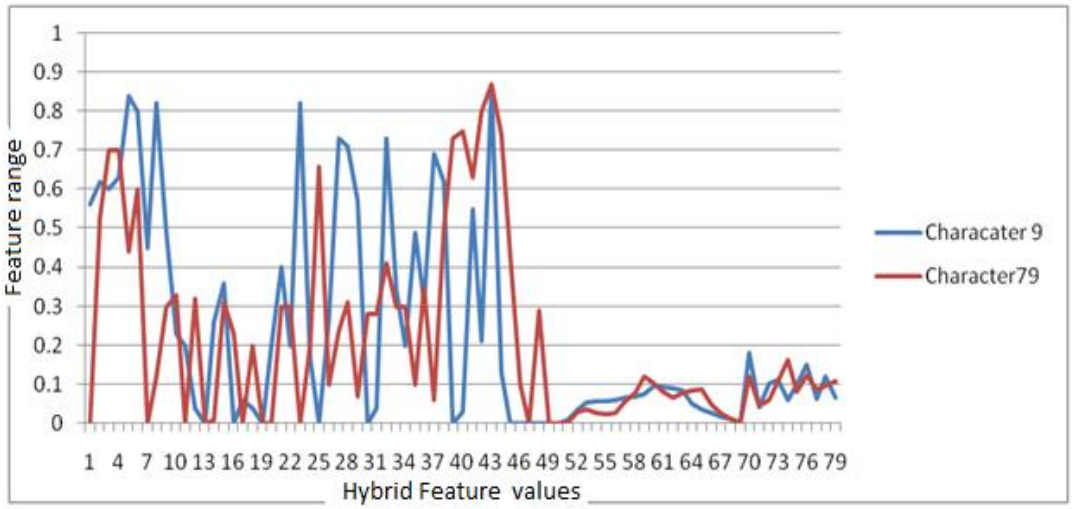


Figure 5.27: Plot showing HF vectors of Character 9 and Character 79.

Character number 9 is written as  and

79th Character is  which is often confused as Character number 8 

Method 3: Gabor wavelet features (GF)

Labeled Numbers of Characters	Precision	Recall	F-measure
1	1.000000	0.950000	0.974359
2	1.000000	1.000000	1.000000
3	0.950000	0.950000	0.950000
4	0.950000	0.950000	0.950000
5	0.952381	1.000000	0.975610
6	1.000000	1.000000	1.000000
7	1.000000	0.950000	0.974359
8	0.952381	1.000000	0.975610
9	1.000000	1.000000	1.000000
10	1.000000	1.000000	1.000000

Table 5.12: Performance of Gabor wavelets features.

The Recognized characters in sequence corresponding to the handwritten page are:

Row1 - 1 2 3 4 5 6 7 8 9 0
Row2 - 1 2 3 4 5 6 7 8 9 0
Row3 - 1 2 3 4 5 6 7 8 9 0
Row4 - 1 2 3 4 5 6 7 8 9 0
Row5 - 1 2 3 4 5 6 7 8 9 0
Row6 - 1 2 3 4 5 6 7 8 9 0
Row7 - 1 2 3 4 5 6 7 8 9 0
Row8 - 1 2 3 4 5 6 7 8 9 0
Row9 - 1 2 3 4 5 6 7 8 9 0
Row10 - ***5*** 2 3 4 5 6 7 8 9 0
Row11 - 1 2 3 4 5 6 7 8 9 0
Row12 - 1 2 3 4 5 6 7 8 9 0
Row13 - 1 2 3 4 5 6 ***8*** 8 9 0
Row14 - 1 2 3 4 5 6 7 8 9 0
Row15 - 1 2 3 4 5 6 7 8 9 0
Row16 - 1 2 3 4 5 6 7 8 9 0
Row17 - 1 2 3 4 5 6 7 8 9 0
Row18 - 1 2 3 4 5 6 7 8 9 0
Row19 - 1 2 ***4*** ***3*** 5 6 7 8 9 0
Row20 - 1 2 3 4 5 6 7 8 9 0

HIT= Number of correctly recognized characters for the class,
MISS= Number of misrecognized characters for the class,
RR= Recognition Rate

For 1 : HIT=**19** MISS=1 TOTAL=20 RR=95.00 %
For 2 : HIT=20 MISS=0 TOTAL=20 RR=100.00 %
For 3 : HIT=**19** MISS=1 TOTAL=20 RR=95.00 %
For 4 : HIT=**19** MISS=1 TOTAL=20 RR=95.00 %
For 5 : HIT=20 MISS=0 TOTAL=20 RR=100.00 %
For 6 : HIT=20 MISS=0 TOTAL=20 RR=100.00 %
For 7 : HIT=**19** MISS=1 TOTAL=20 RR=95.00 %
For 8 : HIT=20 MISS=0 TOTAL=20 RR=100.00 %
For 9 : HIT=20 MISS=0 TOTAL=20 RR=100.00 %
For 10 : HIT=20 MISS=0 TOTAL=20 RR=100.00 %

The number of Test characters is: 200

Recognition Rate (R*R*): = $\frac{Number\ \ of\ correctly\ \ recognized\ \ characters}{Total\ \ Number\ \ of\ Test\ characters}$ x100 =98.0%

Error Rate: = $\frac{Number\ \ of\ misrecognised\ \ \ characters}{Total\ \ Number\ \ of\ \ characters\ \ \ classified}$ x 100= 2.040816%

Out of 200 test characters, Character 5 is misclassified as 1 in Row 10. Character 8 is misclassified as 7 in Row 13. Character 4 is misclassified as 3 and Character 3 is misclassified as Character 4 in Row 19.

It is observed from the Table 5.11 that the Gabor wavelets features method is more robust, efficient and reliable feature set.

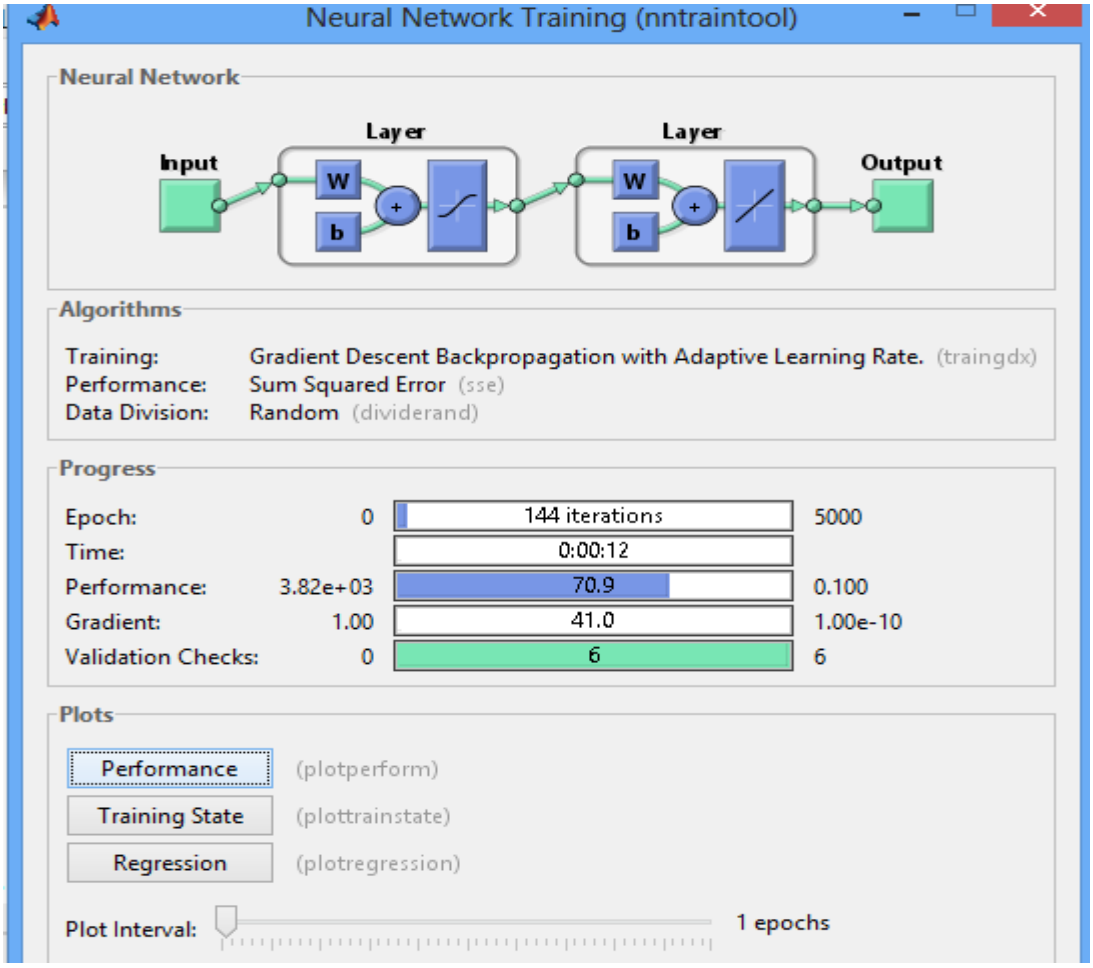


Figure 5.28: Neural Network Training for the performance of MLP4, Accuracy%: 98.0%, Input nodes=22 nodes (Gabor wavelet feature vectors), Hidden Layer neurons: 10, Output layer nodes: 10 Labeled Character Symbols, trained with 144 iterations with 6 maximum validation failures, 5000 maximum number of epochs for training, .01 performance goal , 1.00e-10 minimum performance gradients.

5.2 .12. Data sets

Some samples from the training sets are shown in Figure 5.29, 5.30, 55.31 and 5.32.

Figure 5.29: Samples of 10 characters from 27 characters set.

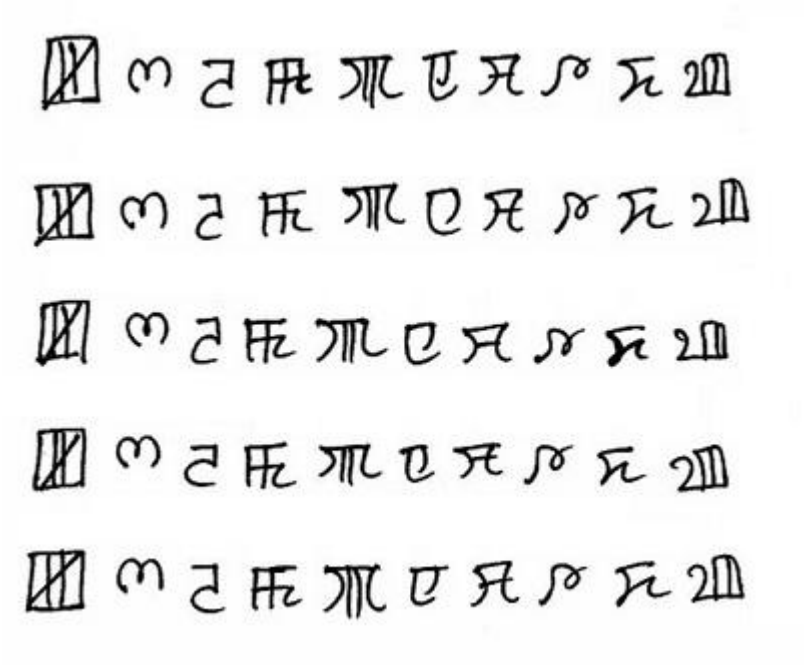


Figure 5.30: Samples of 10 characters from 27 characters set.

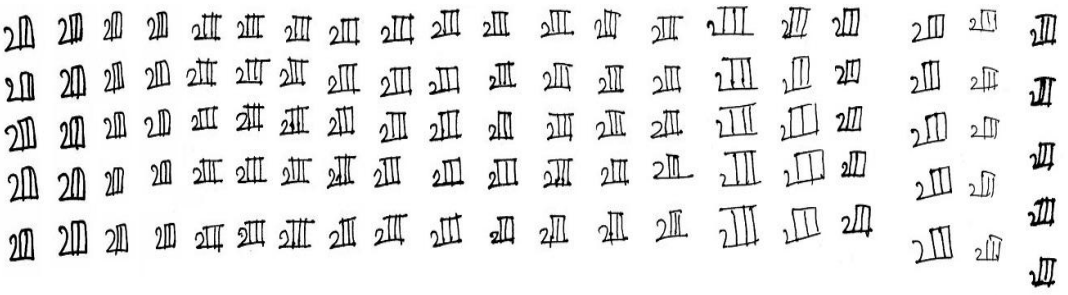


Figure 5.31: Samples of 100 samples for the Character .

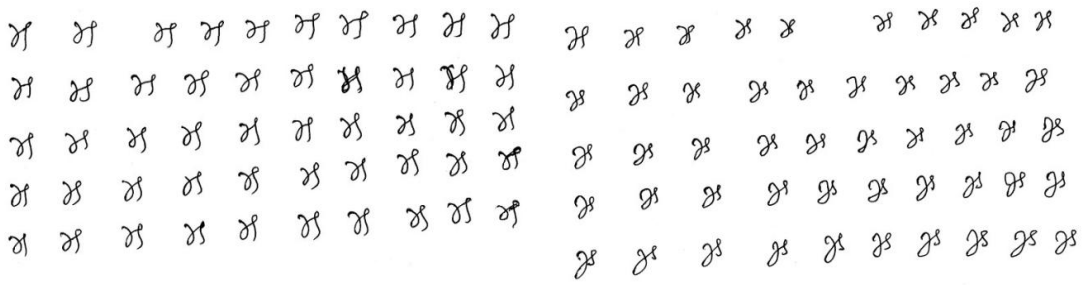


Figure 5.32: Samples of 100 samples for the Digit number 7.

5.3 Conclusion

Performance analysis of the proposed hybrid features of the Recognition System based on the experimental results is presented. The recognition results of digits as well as characters with proposed hybrid features are significantly promising.

The overall steps required for the recognition of digits and characters from an input image page are discussed. The investigations and performances of the proposed feature set namely Hybrid (PF+FF) is presented and lastly the performance observations of Hybrid (PF+FF) features and Gabor wavelets features are highlighted.

Chapter 6
Isolated Handwritten Character Recognition System of Manipuri Script

This chapter describes the prototype isolated handwritten recognition system implemented for the case study of classification of isolated handwritten digits and characters. The stages of the system are explained in the order of the processing. Some general issues and options related to the implementation are addressed. Based on the experimentations using the training and testing sets of five groups of 54 characters of Manipuri Script and using the trained ANN classifiers in Chapter 5, the recognition system for the digits, characters and a typical word from the test input page are presented in this chapter.

6.1 Recognition System

The block diagram of the Handwritten Character Recognition system of Manipuri Script (HCRMS) is shown in Figure 6.1.

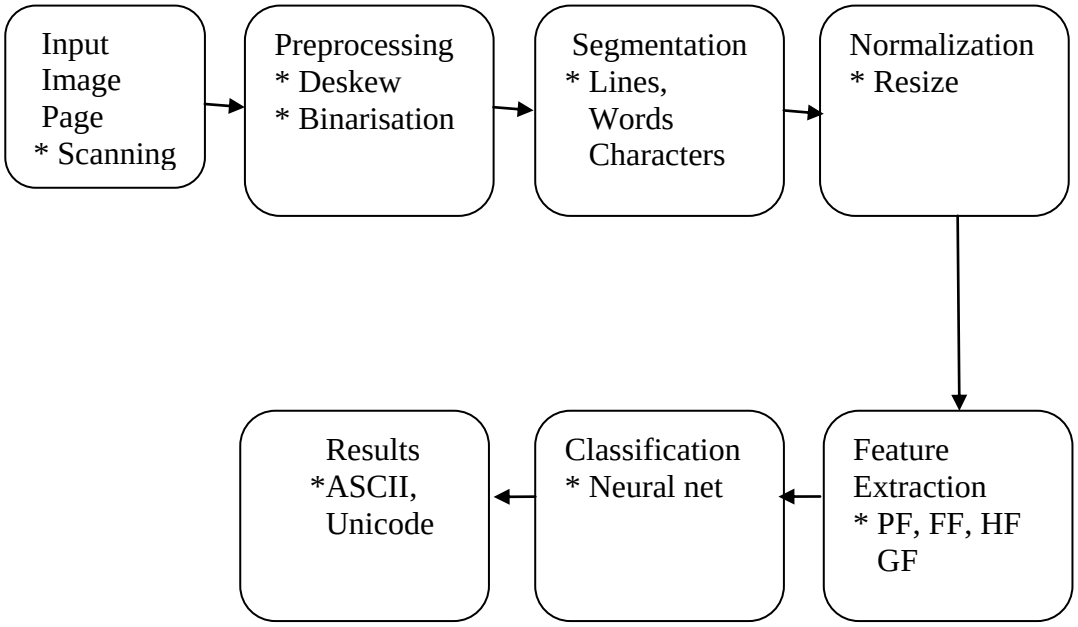
The stages of the technique are as follows:

(i) Scanning (ii) Preprocessing (iii) Segmentation (iv) Normalisation

(v) Feature extraction (vi) Recognition/Classification techniques.

Preprocessing of Input Image Page containing deskewed text lines is performed. The image is thresholded and all connected components (objects) that have fewer than 10 pixels as salt and pepper noise are removed. The lines are segmented using segmentation program based on the algorithm from Section 3.1.1, Section 3.1.2. From the extracted text line, the digits or characters are segmented, cropped and then normalised the same to 60x80 using bicubic interpolation method.

For each digit or character of the Test Set, the Hybrid Features (HF=PF+FF) as mentioned in Section 4.1.2.4 are extracted for testing. The Gabor wavelets (filters) features after dimension reduction are also presented. A block diagram of Handwritten Character Recognition of Manipuri Script (HCRMS) is shown in Figure 6.1.



(PF: Probabilistic Features, FF: Fuzzy Features and HF: Hybrid Features)

Figure 6.1: A block diagram of a System for Handwritten Character Recognition of Manipuri Script (HCRMS)

Each feature set are tested separately for recognition using the previously trained weights of the MLP classifier. The recognized character is then coded for its equivalent Unicode in 'UTF-8' encoding scheme and the codes are written in text file for displaying using a font 'Eeyek Unicode.ttf' for the Manipuri script. The software used for program development is MATLAB R2013a which is currently not supporting the Manipuri Script in Unicode standard [64,65]. The process is repeated for all the digits and characters respectively in the extracted line. Then the whole process is repeated for any existing remaining text lines in the input file.

A typical word of the Manipuri language 'Mukna' is tested for recognition using the previously trained two neural networks. If the test characters are from the Middle zone, then previously trained neural network MLP2 (B) (Rprop) is selected for classification and if the test characters are from the Upper and Lower zones, the then previously trained neural network MLP4(C) is selected.

6.1.1 Recognition system for isolated Digits

The first stage in isolated handwritten character recognition is the data acquisition. There are primarily two ways of acquiring images: scanners and cameras. Depending on the needs and facilities, both of them can produce either binary or gray-scale images with a varying number of bits per pixel. The acquisition of color images is also practicable, but seldom used, due to the large increase in the amount of data and only moderate gain in recognition accuracy. The scanning resolution may vary but it is typically between 200 and 400 dots per inch in both spatial directions. The resolution needed is inversely proportional to the expected size of the characters. The recognition system should be made independent of the original image size and resolution by image normalization.

In the prototype system, the training and test samples were scanned by using automatic feed for A4-sized paper documents. The resolution was 300 dots per inch in both directions.

The results of the digits recognition of an input image page using the trained neural network MLP1 **(**A**),** (Hidden units = 36,Accuracy% = 95.5%) of Experiment 1 (Run1:Trainining set: P2+P3+P4+P5 and test set: P1) of section 5.2 .5 ANN Classifier of chapter 5 are shown in Figure 6.2.

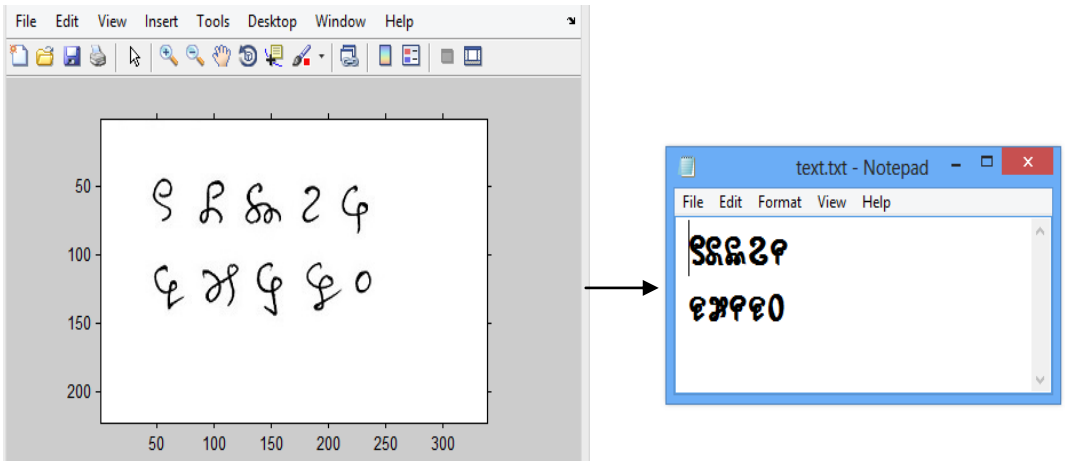


Figure 6.2(a) Input Image Page (b) Output Page with Correctly recognized digits

The outputs of the steps of the segmentation and recognition process of the input image page are shown in the followings:

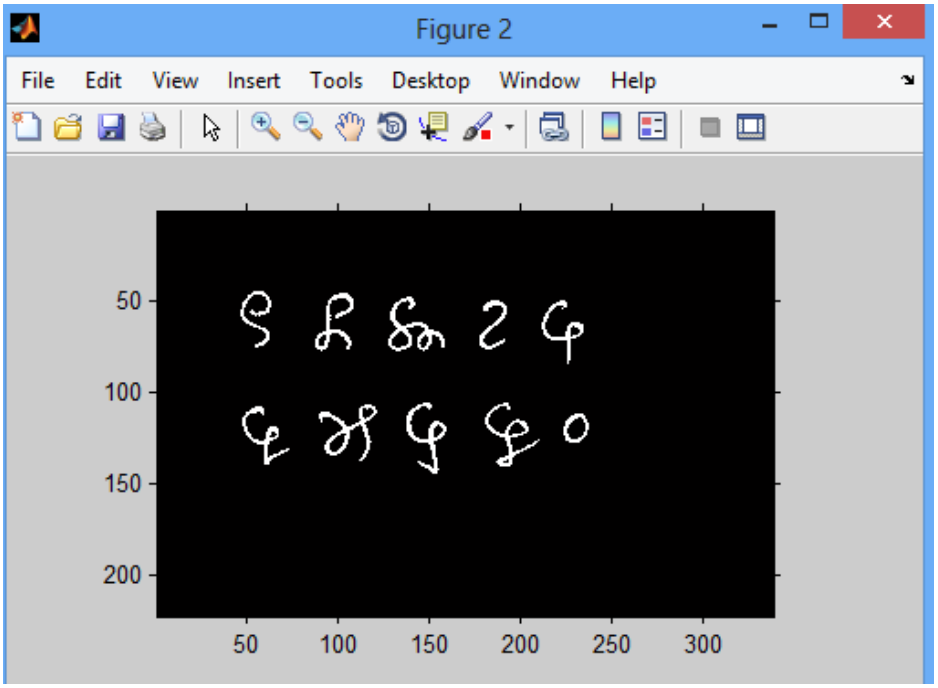


Figure 6.3: Image Negative of the binary input image page

The deskewed input image page containing handwritten digits is thresholded using Otsu's algorithm and negative binary image is shown in Figure 6.3. Then, from the binary image all connected components (objects) that have fewer than 10 pixels as salt and pepper noise are removed, producing another binary image page and its complement is shown in Figure 6.4. The default connectivity is 8 for two dimensions.

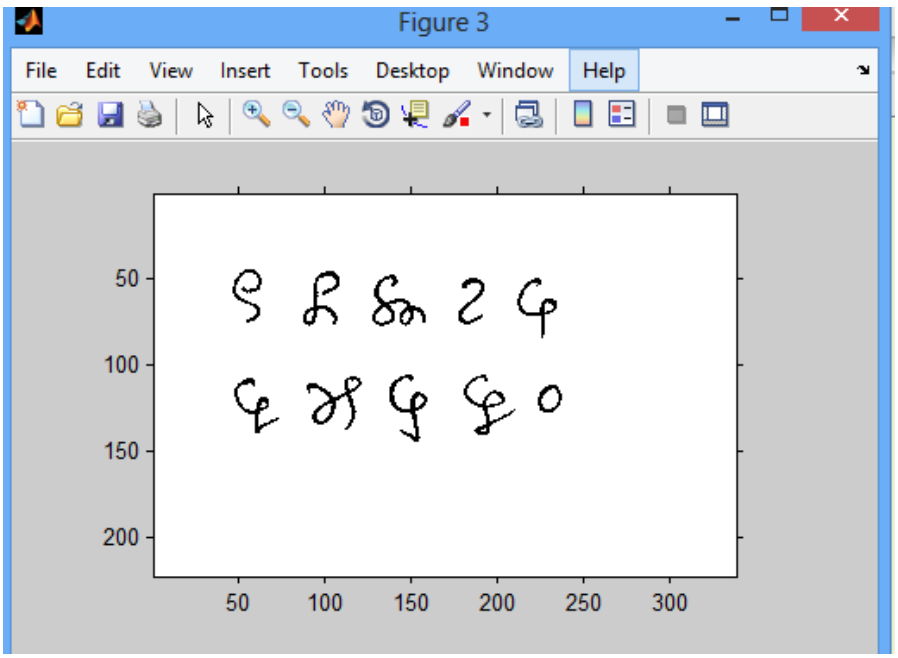


Figure 6.4: Image negative of the binary input image after removing salt and pepper noises fewer than 10 pixels.

6.1.2 Line Segmentation for isolated digits

Algorithm:

1. Read input image to matrix , *im*
2. Convert to gray scale if '*im*' is RGB.
3. Compute threshold, T of image *im* using OTSU's algorithm.
4. Convert to binary image *im* using threshold, T.
5. Remove noise such as objects containing fewer than 10 pixels.
6. Find sum of pixels for each row and store in 'RowSum' array.

To find blank lines-

7. Using regionprops function find length (Area) of all zeros '0' pixel regions in RowSum array.(size equal to row length of the image)
8. Using regionprops function, find 'PixelIdxList' which is p-element vector containing the linear indexes of the pixels in the zero pixel region.

To find Text lines-

9. Similarly, find length of all ones '1's pixel region in RowSum array.
10. Similarly, find 'PixelIdxList' for all ones '1's.
11. find indexes of regions >=5 in Area and store in LongRegion
12. By vertical concatenation of matrices for zero regions, store the result in the matrix 'Indexes'. So the Indexes contains all the row wise locations for blank lines.
13. Assign *im* (Indexes)=0; The output image C is shown in Figure 6.5.
14. Compute compliment c*im* = ~ *im*

Line Segmentation-

15. Store size of c*im* to [x, y], find rowwise sum to column matrix *mat1*.
16. mat2=y-mat1;
17. mat3=mat2~=0;
18. mat4=diff(mat3);
19. index1 = find(mat4);
20. I1=index1; 1 and -1 indicators are found as the beginning and end of text lines.
21. Then for every row index in *Idex1* to next index location, all pixels are extracted from image *cim* and results are stored in *mat5* which contains all pixels of the segmented line1 as shown in Figure6.6. By incrementing the index pointer, the next text line number2 may be similarly segmented.

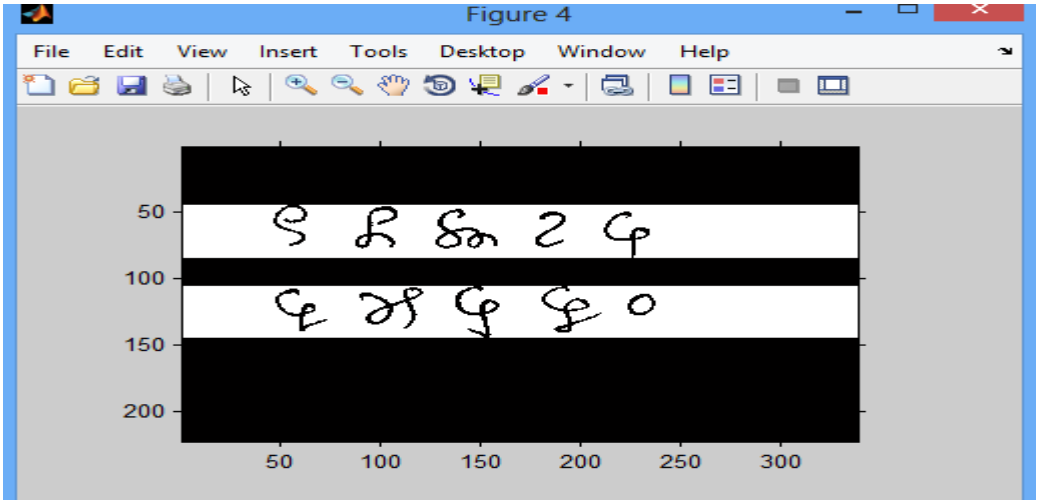


Figure 6.5: Image C having black background lines between the text lines

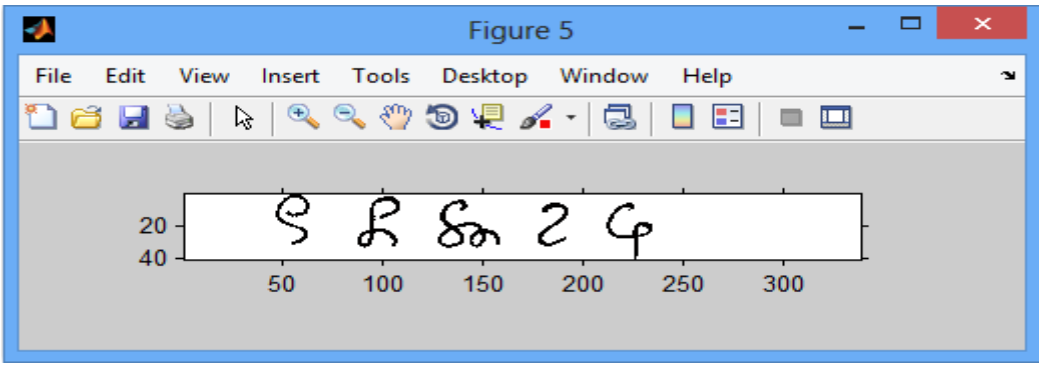


Figure 6.6: Segmented Line number 1.

6.1.3 Segmentation of isolated Digits

In case of digit segmentation, the following steps are performed.

22. Compliments of *mat5 are* stored in a matrix *award*.
23. The sums of columns of the segmented line in *award* are found.
24. From step 7 as mentioned in line segmentation onwards are performed in terms of columns for the digit segmentation. The vertically segmented digits are stored in the matrix *mat5char*. The segmented digits are shown in Figure 6.7.

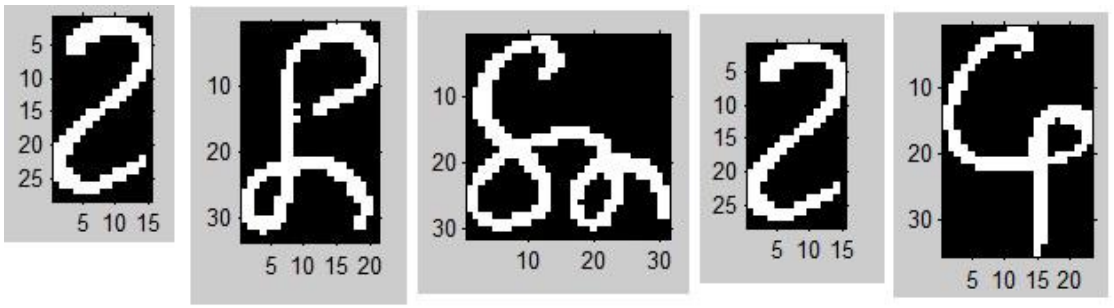


Figure 6.7: Segmented digits of line number 1.

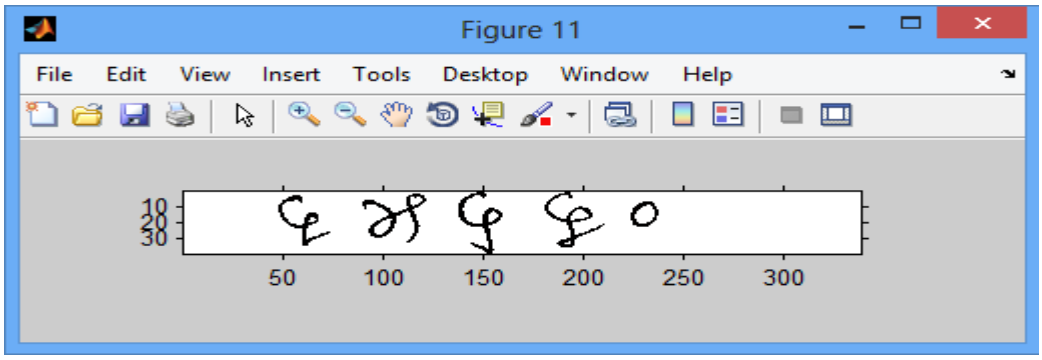


Figure 6.8: Segmented Line number 2.

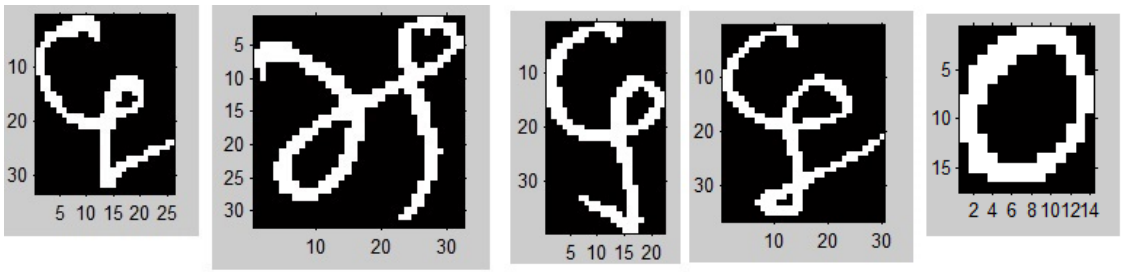


Figure 6.9: Segmented digits of line number 2.

The segmented line 2 and its segmented digits are shown in Figure 6.8and Figure 6.9respectively.

6.1.4 Recognition of isolated Digits

Connected component analysis is performed on the segmented line of digits and properties of image regions such as area, centroid and bounding boxes are measured using *regionprops* function of MATLAB. Then, using *imcrop* function, the image is cropped, with the parameters of the bounding box which is four -element position vector [xmin ymin width height] that specifies the size and position of the crop rectangle. Then, the Hybrid Feature vectors (HF) are computed using negative of the cropped image and the feature vectors is classified using the trained MLP1 **(A).** The results after recognition of the digits are written in the Windows Notepad as shown in Figure 6.2(b).

6.1.5 Recognition system for isolated Characters

Some results of the isolated character recognition system are presented in this Section.

Input Image Page of characters as shown in Figure 6.10 is shown below:

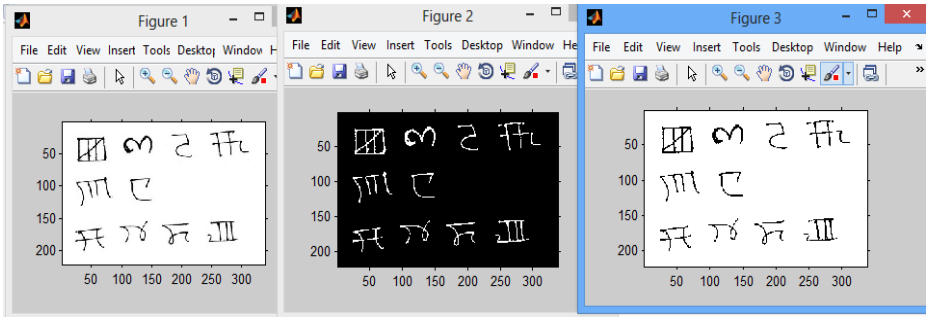


Figure 6.10: Input image page, Negative and Noise free binary image page

The segmented line 1 and characters are shown in Figure 6.11 and segmented line 2, line 3 and segmented characters are shown in Figure 6.12.

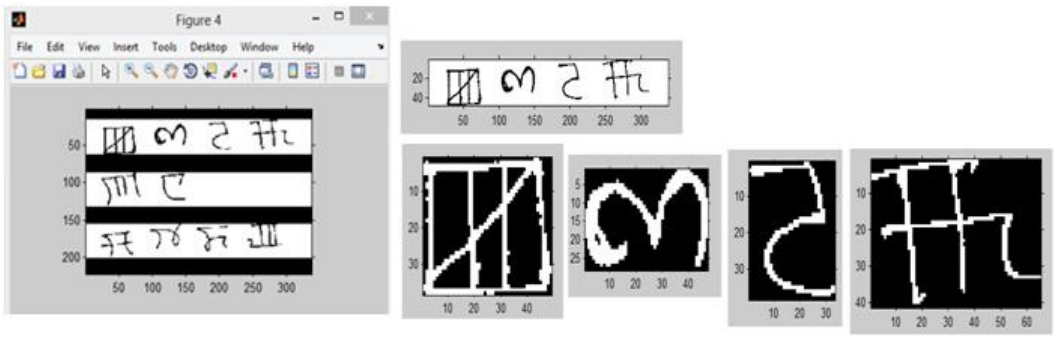


Figure 6.11: Segmented lines 1 and characters of the Input image page

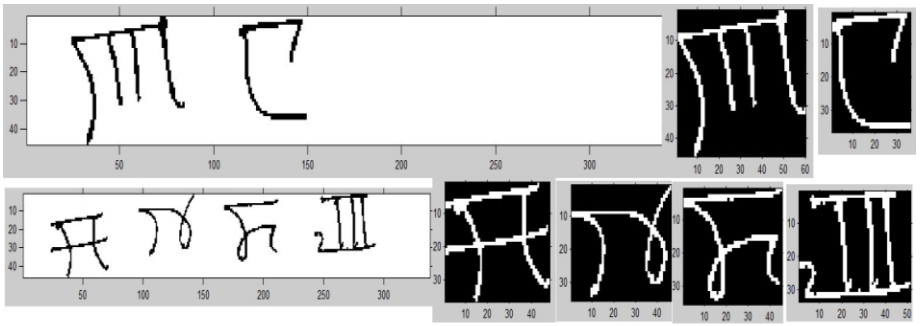


Figure 6.12: Segmented line 2, line 3 and segmented characters

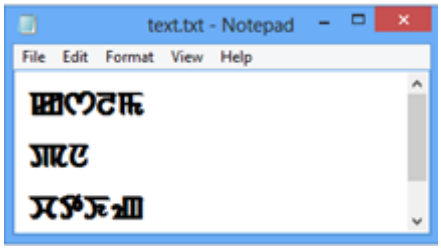


Figure 6.13: Recognition result of the input image page of Figure 6.10.

The trained neural network MLP2 (B), classifier of Section 5.2 .6 Classification of MLP2 (B) for 34 classes, is used for the purpose of testing the Hybrid Feature vectors (HF) of the extracted characters of the lines of the input image page. The recognized characters of the input image page are shown in Figure 6.13.

The configuration of neural network MLP2 **(**B) and Accuracy% are:

MLP2 (B) with (traingdx), Accuracy%: 88.3824%, Input nodes=79 nodes (Hybrid feature vectors), Hidden Layer Nodes: 269, Output layer nodes: 34 Labeled Character Symbols.

6.1.6 Recognition system for a word (with isolated characters)

The following algorithm is implemented for the purpose of recognition process of isolated characters only in order to test the trained neural network and it is not a generalized algorithm for a word with the following problems:

i. Problem of broken characters
ii. Problem of overlapped characters
iii. Problem of touching characters

Line Segmentation for isolated characters:

After thresholding the deskewed input image page of characters, as shown in Figure 6.14, using Otsu's algorithm and from the negative binary image page, all connected components (objects) that have fewer than 10 pixels as salt and pepper noise are removed using *bwareaopen* procedure thereby producing another binary image page for further processing as shown in Figure 6.1 5

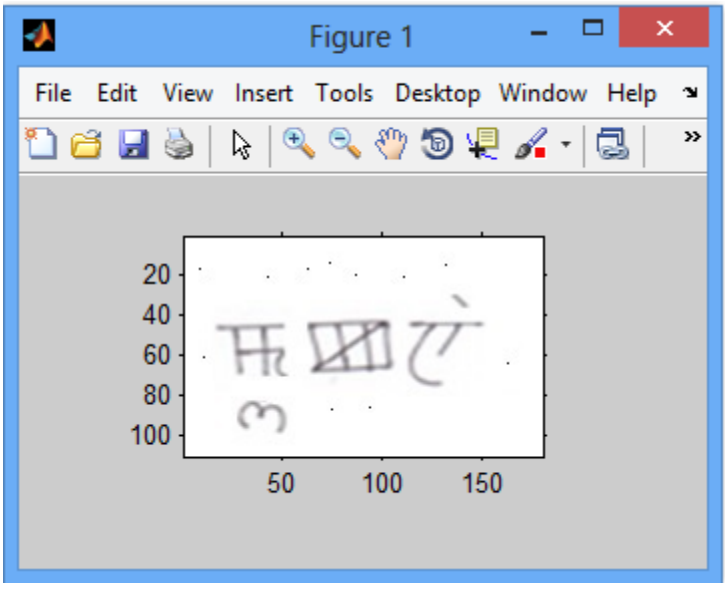


Figure 6.14: A noisy image sample of word 'Mukna'

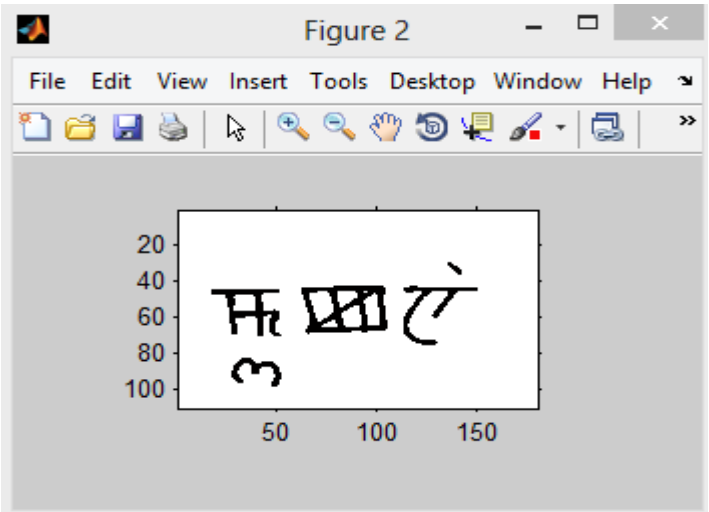


Figure 6.15: Thresholded word 'Mukna' without noise

The default connectivity is 8 for two dimensions. The horizontal projection profiles (HPP) and vertical projection profile (VPP) are shown in Figure 6.14.

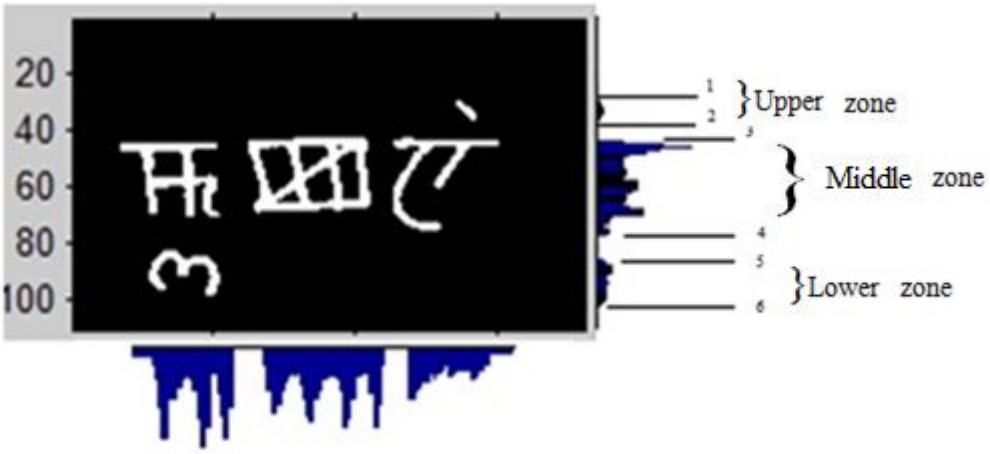


Figure 6.16: HPP and VPP of a word image 'Mukna'

The steps for *line detection, character extraction and recognition using previously trained neural network* in the negative binary image page shown in Figure 6.16 are:

Line detection:

1. The row coordinates of lines of upper zone are found by scanning rows of the deskewed negative binary input page for non zero sums of row as beginning of zone (Upper zone 1) and zero sums as the end of the zone (Upper zone 2) and are stored in 1-D matrix *matE*. The extracted Upper zone is stored in matrix L1 as shown in Figure 6.17.

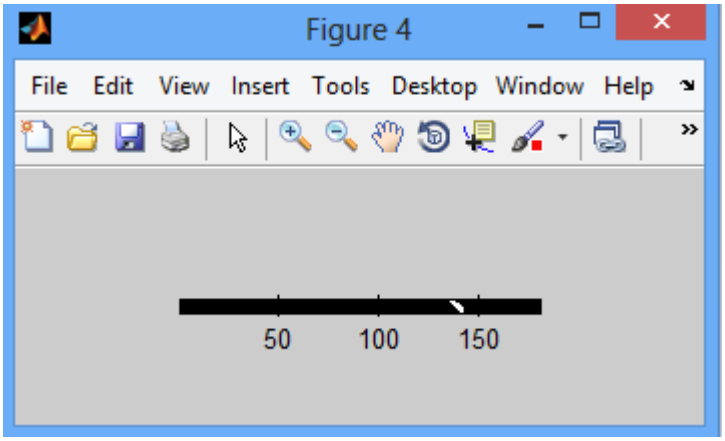


Figure 6.17: Upper zone of word image 'Mukna'

2. Similarly, the locations of Middle zone (3 and 4) and Lower zone (5 and 6) are extracted and stored in 1-D matrix *matE* and also the extracted text lines for Mid zone and Lower zone are saved in sub-image matrices L2 and L3 as shown in Figure 6.18 and Figure 6.19.

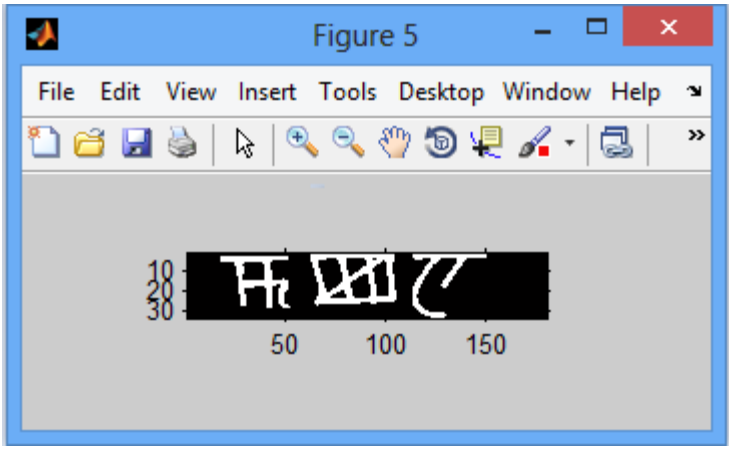


Figure 6.18: Middle zone of word image 'Mukna'

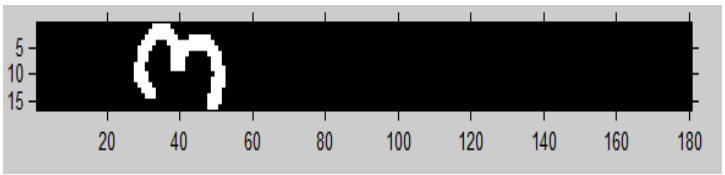


Figure 6.19: Lower zone of word image 'Mukna'

3. The column coordinates of characters of upper zone are found by scanning columns of L1 for non-zero sums of columns as beginning of character and zero sums as the end of the character and are stored in 1-D matrix *matL1*.

Similarly, the column coordinates of characters of Middle zone (3 and 4) and Lower zone (5 and 6) are extracted and stored in corresponding 1-D matrices matL2 *and* matL2.

Character extraction and Recognition:

4. The connected components in the binary image of upper zone line L1 are performed using the 8-nearest neighbor connected objects, the numbers of objects or characters N and the matrix L of the same size containing labels of the connected components are returned. The elements of L are integer values greater than or equal to 0. The pixels labeled 0 are the background. The pixels labeled 1 make up one object; the pixels labeled 2 make up a second object, and so on.
5. The 'BoundingBox' of the character is found by measuring the properties of the image region. It is a rectangle *rect* containing the region, a 1-by-Q *2 vector, where Q is the number of image dimensions: ndims (L), ndims (BW), or numel (CC.ImageSize). BoundingBox is [ul_corner width], where:

   Ul_corner is in the form [x y] and specifies the upper-left corner of the bounding box.

   widths in the form [x_width y_width] and specifies the width of the bounding box along each dimension
6. Then the *imcrop* procedure on the image L1 with bounding box *rect* is performed using the syntax, bw_8060 = imcrop (L1, *rect*), crops the image L1. *rect* is a four-element position vector [xmin ymin width height] that specifies the size and position of the crop rectangle. The list of crop image before size normalization is shown in Figure 6.20.

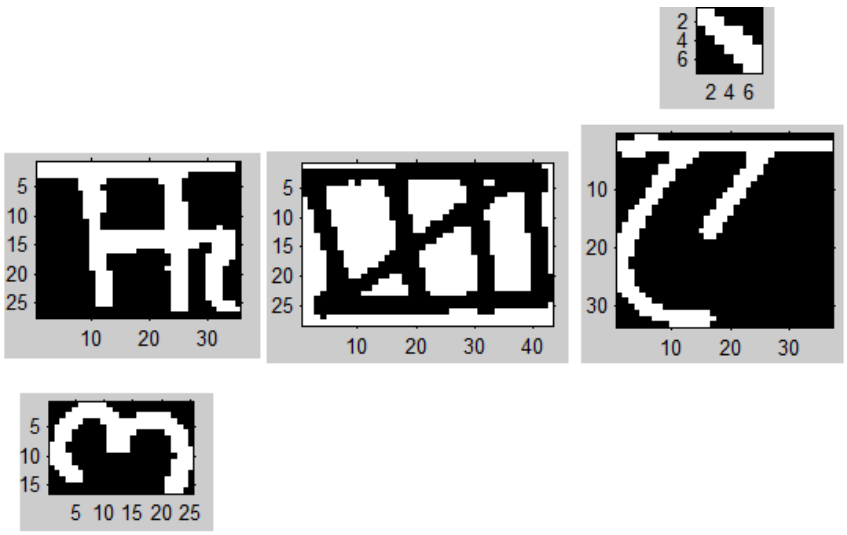


Figure 6.20: List of extracted characters from three zones.

7. The crop image, bw_8060 is resized using imresize function with rows and columns as (80, 60). The method is bicubic interpolation. The resized image for the vowel character in Lower zone is shown in Figure 6.21.

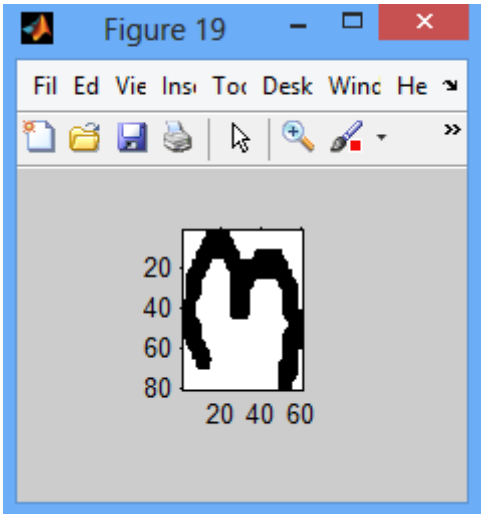


Figure 6.21: Vowel character 'u' in Lower zone

8. The proposed Hybrid Feature (HF) is extracted from the resized and normalized bw_8060 binary image and is stored in *charvec* which has 79 feature values.
9. Then , the recognition process for this 79 features as Hybrid Feature (HF) is simulated using the previously trained neural network model MLP4(C) as mentioned in Section 5.2 .9. This network model is selected only if the test character is either from the Upper zone or Lower zone. Performance of the MLP4(C) is given in Figure 5.23. Accuracy%: 97.5% MLP4(C), Input nodes=79 nodes (Hybrid feature vectors), Output layer nodes: 10 Labeled Character Symbols, 28 units at hidden layer.

   This trained model is selected as the classifier because the trained characters are from Upper and Lower zones only with one exception, Full stop character.
10. If the test characters are from the Middle zone, then the trained neural network MLP2 (B) (Rprop) is selected ,(Accuracy %: 90.1471%, Input nodes=79 nodes (Hybrid feature vectors), Hidden Layer Nodes: 155, Output layer nodes: 34 Labeled Character

Symbols) or MLP2 (B) (traingdx) Accuracy%: 88.3824%, Input nodes=79 nodes (Hybrid feature vectors), Hidden Layer Nodes: 269, Output layer nodes: 34 Labeled Character Symbols is selected.

11. The maximum value and corresponding character position number are found after simulation and the number is mapped to the corresponding Unicode of the script.
12. From the matrices *matL1, matL2 and matL3* for column information and *matE* for row information, the sequence of characters to be printed on Notepad text file is arranged and the characters are printed. For the given word, the information of the three zones in *matE* are [(30, 36), (43, 75), (83, 98)]. The column information of the three zones are matL1=(136,142),matL2= [(18,52),(65,105),(114,150)] and matL3=(28,52).

The average character width is computed from middle zone column information *matL2* and the value is 37. By sorting the starting column locations of the matL1, matL2 and matL3 of the three zones i.e., (18, 28, 65,114 and 136), the sequence of characters to be printed of the word are arranged. The character position may also be found by dividing the starting column information in matL1 i.e, fix (136/3) =3. The letter in upper zone is found to be attached with 3rd character of the middle zone. The recognized output is shown in Figure 6.22.

Figure 6.22: Output Text in Notepad

6.1.7 Recognition system for a word with isolated and overlapped characters

The following algorithm is implemented for the purpose of recognition process of isolated characters but overlapped only in order to test the trained neural network for a word shown in Figure 6.23 and it is not a generalized algorithm with the problem of touching characters.

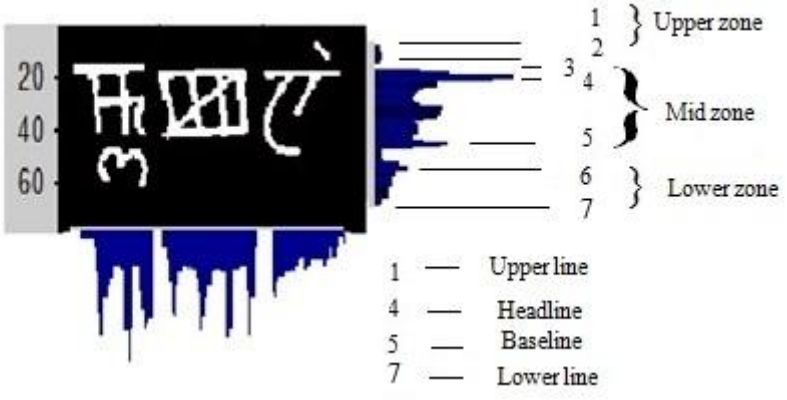


Figure 6.23: HPP and VPP of a word with overlapped characters

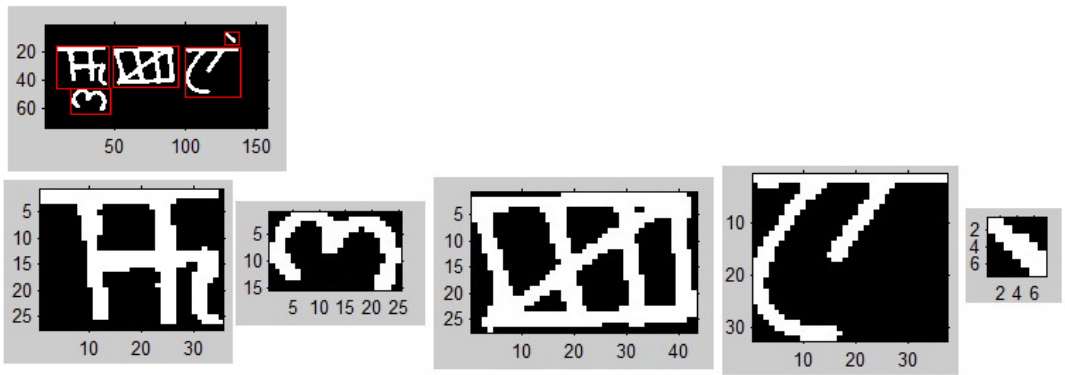


Figure 6.24: Segmented characters

From the binary image, after objects having fewer than 10 pixels are removed, connected component analysis is performed and bounding boxes are computed for each character. The row and column coordinates of each label for the object are recorded. While scanning the rows of the image, the row location of Upper line 1 is found when the sum of rows is not equal to zero. When the scanning is continued for further rows, the row coordinate of Upper line 2 is found when sum of rows equal zero. Further scanning the

rows, row for Mid zone 3 is found when the row is not equal to zero and row coordinate for Lower line is found when sum of rows equal zero. The row coordinate for Head line 4 is found when scanning the rows, where the row sum is the maximum. Then row coordinate for Baseline 5 is found by scanning from the maximum row towards the Head line 4 where the row sum is the maximum. The characters in the Mid zone are the base characters. Average character height (ACH) is computed from running total of sum of heights of base characters divided by the number of characters. If the row coordinates of characters is greater than Headline 4 and less than Baseline 5 are the base characters. By using the row and column coordinates, the image of the character is cropped and from the cropped negative image is resized to (80, 60) for the size normalization. Then Hybrid Feature vectors are extracted for classification.

For the classification of characters in upper zone, if the row coordinate of the character is less than Headline 4, then , the recognition process for this 79 features as Hybrid Feature (HF) is simulated using the previously trained neural network model MLP4(C) as mentioned in Section 5.2 .9. This network model is selected only if the test character is either from the upper zone or lower zone. Performance of the MLP4(C) is given in Figure 5.23. Accuracy%: 97.5% MLP4(C), Input nodes=79 nodes (Hybrid feature vectors), Output layer nodes: 10 Labeled Character Symbols, 28 units at hidden layer.

For the classification of characters in lower zone, if the row coordinates of the character is greater and equal to the Baseline 5, the Hybrid Feature (HF) is simulated using the previously trained neural network model MLP4(C).

If the test characters are from the Middle zone, then the trained neural network MLP2 (B) (Rprop) is selected. The network has the Accuracy %: 90.1471%, Input nodes=79 nodes (Hybrid feature vectors), Hidden Layer Nodes: 155, Output layer nodes: 34 Labeled Character Symbols and MLP2 (B) (traingdx) Accuracy%: 88.3824%, Input nodes=79 nodes (Hybrid feature vectors), Hidden Layer Nodes: 269, Output layer nodes: 34 Labeled Character Symbols.

The maximum value and corresponding character position number are found after simulation and the number is mapped to the corresponding unicode of the script. Then the recognised characters are written into the Windows Notepad as shown in Figure 6.25.

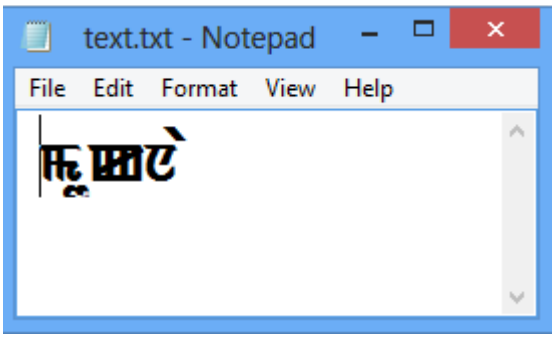


Figure 6.25: Recognised characters

Also, the word recognition performance can be very significantly improved by using a lexicon. However, these are not the issues addressed by the thesis.

The comprehensive experiments demonstrated that the proposed recognition system with hybrid features can achieve very encouraging results. The experimental results show that the choice of the features affects the performance of the classifier and the proposed feature set results better recognition rate. The generalization of the recognition process has been improved with the size and slant invariant signatures features of the proposed probabilistic feature method. Experimental results indicate that the proposed recognition system performs well and is robust to the writing variations that exist between persons and for a single person at different instances, thus being promising for user independent character recognition system and tolerant to random noise degradations of the characters.

Experiments were conducted on Lenovo S10, Intel[R] Atom[TM] CPU N270 @ 1.60 GHz, 0.99 GB of RAM. The training time of trainings for 5 categories of character sets are in order of days.

MNIST data files [66] are used for experiments. The 4 pixel padding around the digits are removed and pixel values are normalized to the [0...1] range and images are 20 x 20 pixels. Only 2400 out of 60,000 digits are trained using MLP (trainrp) with proposed 79 hybrid features and tested

with 230 digits from 10,000 test digits. Hidden layer nodes with 45 gives an accuracy of 89.1304%.

6.2 Conclusion

In this chapter, the handwritten character recognition system of Manipuri Script (HCRMS) is presented. The recognition systems of an input image page for digits and characters using the trained ANNs classifiers in Chapter 5 are presented. Then the recognition systems for a word with isolated characters and word with overlapped characters are presented in this chapter.

Chapter 7

Conclusion and Future Scope

This research work presents a prototype IHCRMS system for segmenting lines, words, non-touching characters and isolated digits and recognizing the handwritten isolated digits and non-touching characters from a deskewed input image file. Line segmentation from the deskewed input file is performed. From the extracted line, extraction of digits using connected component analysis is presented. Analysis with different strategies for segmenting non-touching characters of handwritten word in different zones with the analysis of vertical, horizontal profiles and connected component analysis are performed. After size normalization of the extracted component, probability features based on the size and slant invariant signatures features are extracted. The handwriting recognition results using K-L divergence with probabilistic features are shown. Fuzzy feature extraction technique from the resized component with zoning is presented. Then Hybrid feature is proposed by combining the two feature sets for the better recognition rates of the characters using two layers feed forward backpropagation neural network (ANN). There are *five* ANNs trained for the different categories of characters.

- 1st neural network for 10 digits, MLP1 (A),
- 2nd neural network for the 34 characters, MLP2(B),
  (27 alphabets of Iyek Ipee + 7 Lonsum Iyek)
- 3rd neural networks for 27 basic characters, MLP3 (B1),
- 4th neural network for 10 characters (8 vowels -Cheitap Iyek letters + 2 khudam Iyek- Cheikhei symbol for Fullstop and Apun symbol for Sign of Ligature), MLP4(C) and
- 5th neural network for 44 characters without 10 numerals, MLP5 (D).

By splitting the character patterns into these separate classes, it eliminates the problems of ambiguity faced when similar confusing character pairs are presented to a single network.

The experimental results show that the choice of the features affects the performance of the classifier and the proposed hybrid feature set gives better recognition rate. The generalization of the recognition process has been improved with the size and slant invariant signatures features of the proposed probabilistic feature method. Experimental results indicate that the recognition system performs well and is robust to the writing variations that exist between persons and for a single person at different instances, thus being promising for user independent isolated digits and characters recognition and tolerant to random noise degradations of the characters. The recognition performance obtained may be enhanced further by experimenting on and choosing the best set of features and classifiers.

7.1 Thesis Contributions

The recognition of isolated digits and characters are addressed in this research. First contributions were made on the recognition of isolated digits and afterwards on isolated character recognition.

The contributions in this thesis are summarised as follows:

(a) Line extraction algorithm from the thresholded input file containing the deskewed lines of handwritten digits, segmentation of isolated handwritten digits from the extracted line and also line segmentation of handwritten characters as well as segmentation of non-touching characters.

(b) A method for fuzzy features (FF) extraction technique is presented.

(c) A method for feature extraction technique using probabilistic (PF) model. The features extracted are investigated and tested for recognition using K-L divergence technique and the results are presented. Also, the same features are investigated and tested for recognition using MLP classifier and the results are presented.

(d) The *proposed hybrid feature(HF)* set is the combination of the said two features sets (PF+FF) giving a total of 97 features and the recognition tests are performed using feed forward backpropagation neural network (MLPs) classifiers with promising recognition rates.

(e) Five trained Feed forward backpropagation neural networks (MLPs) are presented using the HF feature vectors for the recognition of digits as well as the alphabets of Manipuri script.

(f) Results of the recognition of 10 class characters using Gabor feature vectors are highlighted.

7.2 Future Scope

If future research focus not only on the further development of the techniques described, but also on making the techniques applicable in computer systems that will actually be used in the field. Therefore, we foresee an important role for the domain experts in future research: if they are consulted at an early stage in the development, we believe that achievements can be made that prove to be of interest to both sciences as practice.

The following directions may be investigated as future scope in order to improve the performance.

An algorithm may be investigated for correct segmentation of touching characters in different zones, broken characters, and overlapped characters.

- An algorithm may be investigated for word spotting.
- The recognition system can use rejection strategies to reject those characters with relatively low confidence values rather than taking a risk to misrecognize them.
- Different features have different discrimination merits for different recognition purposes. Therefore, fine features should be further investigated for the hybrid feature extraction methods. The development of a new theory and algorithm in feature extraction is also important.
- More training samples are to be used and more hierarchical levels of the classification are to be employed. Investigations of using ensemble classifiers and a cascade classifier are to be investigated.
- Different classifiers such as SVM, CNN and deep learning systems are to be investigated.

References

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

[2] Philip D Wasserman, *Neural Computing Theory and Practice*, Van No Strand Reinhold,New York, 1989.

[3] Jyh-Shing Roger Jang,Chuen-Tsai Sun and Eiji Mizutani, *Neuro-Fuzzy and Soft Computing*, Prentice-Hall of India, Private Limited, New Delhi,2002.

[4] Christopher M. Bishop, *Pattern Recognition and Machine Learning*, Springer,2006.

[5] Wangkhemcha Chingtamlen, *A short history of Kangleipak (Manipur) part-II*,Kangleipak Historical & Cultural Research Centre,Sagolband Thangjam Leirak, Imphal, 2007.

[6] Ng.Kangjia Mangang, *Revival of a closed account, a brief history of kanglei script and the Birth of phoon(zero) in the world of arithmetic and astrology*, Sanmahi Laining Amasung Punshiron Khupham(Salai Punshipham), Lamshang, Imphal, 2003.

[7] T.C.Hodson ,*The Meitheis*, Low price publications, Delhi, 1908.

[8] Y.Tamphajao, *Philosophy & Science on Meetei alphabets*, 1992.

[9] Three Resource Persons for the Directorate of Education(s), Govt. of Manipur, *Meetei Mayek Tamnaba Mapi Lairik*, eyek maru Sanaba Thourangba Kanglup, Kangleipak, 1996.

[10] A. G. Ramakrishnan, The magic of automated recognition of handwriting, Current Science, Vol. 107, No.2, 159-160, 25 July 2014.

[11] https://en.wikipedia.org/wiki/Gibbs%27_inequality

[12] Jack M.Zurada,Introduction to Articicial Neural Systems,Jaico Publishing House, New Delhi, 1997.

[13] G. Boccignone, A. Chianese, L.P. Cordella, and A. Marcelli. Recovering Dynamic Information from Static Handwriting. Pattern Recognition, 26(3):409 – 418, 1993.

[14] P.M. Lallican, C. Viard-gaudin, and S. Knerr. From Off-Line To On-Line Hand writing Recognition. In In Proceedings of the 7th International Workshop on Frontiers in Handwriting Recognition, pages 303–312, Amsterdam, 2004.

[15] S. Mori, C. Y. Suen and K.Yamamoto, Historical Overview of OCR Research and Development, *Proceedings of the IEEE*, 80, 1029-1058, 1992.

[16] K. Roy, S. Vajda , U. Pal and B. B. Chaudhuri , *IEEE* Proceedings of the 9th Int'l Workshop on Frontiers in Handwriting Recognition (IWFHR-9 2004),580-585, 2004.

[17] K. Roy, S. Vajda, U. Pal, B. B. Chaudhuri and A. Belaid, *A System for Indian Postal Automation*, *IEEE* Proceedings of the Eight International Conference on Document Analysis and Recognition (ICDAR'05) , vol.2, 1060-1064,2005.

[18] S. N. Srihari and S. W. Lam, Character Recognition, *Technical Report*, CEDAR-TR-95-1, 1995.

[19] S. Srihari. High-performance reading machines. *Proc. of the IEEE*, 80(7) :1120–1132, 1992.

[20] C. Y. Suen, and R. Legault, C. Nadal, M. Cheriet, and L. Lam, Building a New Generation of Handwriting Recognition Systems, *Pattern Recognition Letters*, 14, 305-315, 1993.

[21] J. Mantas. An overview of character recognition methodologies. *Pattern Recognition*, 19(6):425–430, 1986.

[22] T. Berger L. L. Lee and E. Aviczer. Reliable on-line human signature verification systems. *IEEE Trans. Patterns Analysis and Machine Intelligence*, 18(6):643–647, 1996.

[23] J. C. Simon. Off-line cursive word recognition. *Proc. of the IEEE*, 80(7): 1150–1161, 1992.

[24] S.W. Lee. Off-line recognition of totally unconstrained handwritten numerals using multilayer cluster neural network, *IEEE Trans. Patterns Analysis and Machine Intelligence*, 18(6),648–652, 1996.

[25] R. G. Casey and E. Lecolinet. A survey of methods and strategies in character segmentation. *IEEE Trans. Patterns Analysis Machine Intelligence*, 18, 690-706, 1996.

[26] Flavio Bortolozzi, Alceu de Souza Britto Jr., Luiz S. Oliveira and Marisa Morita, Recent Advances in Handwriting Recognition, *Proceedings of the International Workshop on Document Analysis*, 1-30, March 8-11, Kolkata, India, 2005

[27] Marisa Emika Morita, Automatic recognition of handwritten dates on brazilian bank cheques, *PhD Thesis*, École De Technologie Supérieure Université Du Québec, June, 2003.

[28] M. Blumenstein, Intelligent Techniques for Handwriting Recognition,*PhD Thesis*, School of Information Technology, Faculty of Engineering and Information Technology, Griffith University-Gold Coast Campus, 2000.

[29] C. Y. Suen, Character Recognition by Computer and Applications in *Handbook of Pattern Recognition and Image Processing*, Young, T. Y., Fu, King-Sun, Academic Press Inc., San Diego, CA, 569-586, 1986.

[30] S. Impedovo, L. Ottaviano and S. Occhinegro, Optical Character Recognition – A Survey, *International Journal of Pattern Recognition & Artificial Intelligence*, 5, 1-24, 1991.

[31] V. Nguyen and M. Blumenstein. Techniques for Static Handwriting Trajectory Recovery: A Survey. In Proceedings of the 9th International Workshop on Document Analysis Systems, pages 463–470, Boston, MA, USA, 2010.

[32] O. D. Trier, A. K. Jain, 1995, Evaluation of Binarisation Methods for Document Images, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17, 312-315, 1995.

[33] O. D. Trier, A. K. Jain, 1995, Goal-Directed Evaluation of Binarization Methods, *IEEE Transactions on Pattern Analysis and Machine Intelligence*,17, 1191-1201,1995.

[34] A. T. Abak, U. Baris, B. Sankur, The Performance of Thresholding Algorithms for Optical Character Recognition, *Proceedings of the Fourth International Conference Document Analysis and Recognition (ICDAR '97), Ulm, Germany*, 697-700, 1997.

[35] VishwaBharat@tdil, Oct.2003, ISSN No. 0972-6454, Language Technology Flash, DIT, Minister of Communications & Information Technology, http://tdil.mit.gov.in

[36] Ridler T and Calvard S, *Picture thresholding Using an interactive Selection Method*, IEEE Trans. On Systems,Man and Cybernetics, 8, 630-632, 1978.

[37] Rafael C. Gonzalez, Richard E.Woods and Steven L.Eddins, *Digital Image Processing Using MATLAB*, First Impression, Dorling Kindersley(India) Pvt. Ltd., licensees of Pearson Education in South Asia, 2006.

[38] Otsu N. A Threshold Selection Method from Gray-Level Histogram. *IEEE Transactions on Systems, Man, and Cybernetics* 9, 62-66, 1979**.**

[39] Yan Solihin, C.G. Leedham, The Multi-stage Approach to Grey-Scale Image Thresholding for Specific Applications, *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 2000.

[40] Solihin Y., C.G. Leedham, Integral Ration: A New Class of Thresholding Techniques for Handwritten Image, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 21, No.8, 761-768, August 1999.

[41] N.R. Pal and S.K.Pal, A Review on Image segmentation techniques, *Pattern Recognition, 26,* 1277-1294, 1993.

[42] N.R. Pal and S.K.Pal, Entropy Thrsholding, *Signal Processing*, 16, 97-108, 1989.

[43] N.R. Pal and S.K.Pal, Entropy: A new Definition and Its Applications, *IEEE Trans.Systems, Man and Cybernetics*, 21, 1260-1270, 1991.

[44] Punam K. Saha, Jayaram K.Udupa, Optimum Image Thresholding via Class Uncertainty and Region Homogeneity, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(7):689-706, July 2001.

[45] P.K.Sahoo, S.Soltani, A.K.C Wong, A survey of thresholding techniques, *Computer Vis., Graph., Image Processing*, 41, 233-260, 1988.

[46] W.H.Tsai, Moment-preserving thresholding: a new approach, *Comput.Vis.,Graph.,Image Processing*, 29, 377-393, 1985.

[47] J.N.Kapur, P.K.Saho, A.K.C Wong, A new method for gray level picture thresholding using the entropy of the histogram, *Comput.Vis.,Graph, Image Process*, 29, 273-285, 1985.

[48] S.U.Lee and S.Y.Chung, A comparative performance study of several global thresholding techniques for segmentation, *Comput.Vis.,Graph.,Image process, 52, 171-190, 1990.*

[49] J.Kittler and J.Illingworth, On threshold selection using clustering criteria, *IEEE Trans., Systems, Man and Cybernetics*, SMC-15, 652-655, 1985.

[50] J.Kittler and J.Illingworth, Minimum error thresholding, *Pattern Recognition,*19(1):41- 47, 1986.

[51] Chun-Ming Tsai and Hsai-jian Lee, binarization of Color Document Images via Luminance and Saturation Color Features, *IEEE Trans., on Image Processing*, 11, (4): 434-451, April 2002.

[52] W. Niblack., *An introduction to Digital Image Processing*, Englewood Cliffs., NJ: Prentice-Hall, 115-116, 1986.

[53] J.Bernsen, Dynamic thresholding of gray-level images, *proc. 8th Int.Conf.Pattern Recognition*, Paris, 1251-1255, 1986.

[54] J.Z.Liu and W.Q.Li, The automatic thresholding of gray-level pictures via two-dimensional otsu method(in Chinese), *Acta Automat Sinica*, 19, 101-105, 1993.

[55] J. Gong, L.Li, W.Chen, Fast recursive algorithms for two-dimensional thresholding, *Pattern Recognition*, 31(3):295-300, 1998.

[56] Y.H.Tseng, H.J.Lee, Document image binarization by two-layer block extraction and background intensity determination, *Comput. Vis., Graph.,Image Process.*, 2000.

[57] A. Gupta, M. V. Nagendraprasad, A. Liu, P. S. P. Wang and S. Ayyadurai, An Integrated Architecture for Recognition of Totally Unconstrained Handwritten Numerals, *International Journal of Pattern Recognition and Artificial Intelligence*,7, 757-773, 1993.

[58] A. B. S. Hussain, G. T. Toussaint and R. W. Donaldson, Results Obtained using a Simple Character Recognition Procedure on Munson's Handprinted Data, *IEEE Transactions on Computing*, 21, 201-205, 1972.

[59] G. Srikantan, S. W. Lam, S. N. Srihari, Gradient-based Contour Encoding for Character Recognition, *Pattern Recognition*, 29, 1147-1160, 1996.

[60] C. Arcelli, A Width Independent Fast Thinning Algorithm, *IEEE Trans. Pattern Analysis and Machine Intelligence, 7*, 463-474, 1985.

[61] L. Lam, S.W Lee, C. Y. Lee, Thinning Methodologies-A Comprehensive Survey, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 14, 869-885,1992.

[62] O. Baruch, Line Thinning by Line Following, *Pattern Recognition Letters*, 8, 271- 276,1988.

[63] T. Pavlidis, A Thinning Algorithm for Discrete Binary Images, *Computer Graphics and Image Processing*, 13, 142-157, 1980.

[64] http://std.dkuug.dk/jtc1/sc2/wg2/docs/n3158.pdf

[65] http://tabish.freeshell.org/eeyek/unicode.html

[66] http://yann.lecun.com/exdb/mnist/

[67] http://research.cs.tamu.edu/prism/lectures/pr/pr_l13.pdf

[68] K.Y.Wong, R.G.Casey, F.M.Wahl, Document Analysis system, *IBM .Research and Development*, 26(6):647-656, 1982.

[69] http://cindy.informatik.uni-bremen.de/cosy/teaching/CM_2011/Eval3/pe_efron_93.pdf

[70] C. G. Leedham and P. D. Friday Isolating Individual Handwritten Characters, *Proceedings of the IEE Colloquium on Character Recognition and Applications*, Savoy Place, 4/1-4/7, 1989.

[71] M. Cesar, An Algorithm for Segmenting Handwritten Postal Codes, *International Journal of Man-Machine Studies*, 33,, 63-80, 1990.

[72] P. Gader, M. Magdi and J-H Chiang, Segmentation-based Handwritten Word Recognition, *Proceedings of the 5th USPS Advanced Technology Conference*, 215-226, 1992.

[73] C. Y. Suen, M. Berthod and S. Mori, Automatic Recognition of Handprinted Characters-The State of the Art, *Proceedings of the IEEE*, 68, 469-487, 1980.

[74] N. D. Tucker and F. C. Evans, A Two-step Strategy for Character Recognition using Geometrical Moments, *Proceedings of the 2nd International Conference on Pattern Recognition*, 223-225, 1974.

[75] A. B. S. Hussain, G. T. Toussaint and R. W. Donaldson, Results Obtained using a Simple Character Recognition Procedure on Munson's Handprinted Data, *IEEE Transactions on Computing*, 21, 201-205, 1972.

[76] J. R. Ullmann, Experiments with the n-tuple Method of Pattern Recognition, *IEEE Transactions on Computing*, 18, 1135-1137, 1969.

[77] A. L. Knoll, Experiments with Characteristic Loci for Recognition of Handprinted Characters, *IEEE Transactions on Computing*, 18, 366-372, 1969.

[78] V.K. Govindan and A. P. Shivaprasad, Character Recognition - A Review, *Pattern Recognition*, 23, 671-683, 1990.

[79] G. H. Granlund, Fourier Preprocessing for Hand Print Character Recognition, *IEEETransactions on Computers*, 21, 195-201, 1972.

[80] S. Impedovo, B. Marangelli and A. M. Fanelli, A Fourier Descriptor Set for Recognizing Nonstylised Numerals, *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-8, 640- 645, 1978.

[81] M. Shridhar and A. Badreldin, High Accuracy Character Recognition using Fourier and Topological Descriptors, *Pattern Recognition*, 17, 515-524, 1984.

[82] E. Persoon and K-S. Fu, Shape Discrimination using Fourier Descriptors, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8, 388-397, 1986.

[83] A. Krzyzak, S. Y. Leung and C. Y. Suen, Reconstruction of Two-Dimensional Pattern from Fourier Descriptors, *Machine Vision and Applications*, 2, 123-140, 1989.

[84] Y. Lu, S. Schlosser and M. Janeczko, Fourier Descriptors and Handwritten Digit Recognition, *Machine Vision Applications*, 6, 25-34, 1993.

[85] M. D. Garris, J. L. Blue, G. T. Candela, D. L. Dimmick, J. Geist, P. J. Grother, S. A. Janet, C. L. Wilson, NIST Form-Based Handprint Recognition System, *Technical Report NISTIR 5469*, National Institute of Standards and Technology, Gaithersburg, MD 20899, U.S.A, 1994.

[86] M. Kushnir, K. Abe and K. Matsumoto, Recognition of Handprinted Hebrew Characters using Features Selected in the Hough Transform Space, *Pattern Recognition*, 18, 103-114, 1985.

[87] O. D. Trier, A. K. Jain and T. Taxt, Feature Extraction Methods for Character Recognition-A Survey, *Pattern Recognition*, 29, 641-662, 1996.

[88] G. Srikantan, S. W. Lam, S. N. Srihari, Gradient-based Contour Encoding for Character Recognition, *Pattern Recognition*, 29, 1147-1160, 1996.

[89] H. Takahashi, A Neural Net OCR using geometrical and zonal pattern features, *Proceedings of the 1st International Conference on Document Analysis and Recognition (ICDAR '91), St Malo,France*, 821-828, 1991.

[90] J. Cai and Z-Q Liu, Integration of Structural and Statistical Information for Unconstrained Handwritten Numeral Recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21, 263-270, 1999.

[91] F. Kimura and M. Shridhar, Handwritten Numerical Recognition based on Multiple Algorithms, *Pattern Recognition*, 24, 969-983, 1991.

[92] T. Pavlidis, A Vectorizer and Feature Extractor for Document Recognition, *Computer Vision and Graphic Image Processing*, 35, 111-127, 1986.

[93] S. Kahan, T. Pavlidis and H. S. Baird, On the Recognition of Printed Characters of any Font and Size, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9, 274-288, 1987.

[94] L. Lam and C. Y. Suen, Structural Classification and Relaxation Matching of Totally Unconstrained Handwritten Zip-Code Numbers, *Pattern Recognition*, 21, 19-31,1988.

[95] C. Y. Suen, C. Nadal, R. Legault, T. A. Mai, and L. Lam, Computer Recognition of Unconstrained Handwritten Numerals, *Proceedings of the IEEE*, 80, 1162-1180, 1992.

[96] C. Y. Suen, and R. Legault, C. Nadal, M. Cheriet, and L. Lam, Building a New Generation of Handwriting Recognition Systems, *Pattern Recognition Letters*, 14, 305-315, 1993.

[97] S.Britto Jr., R.Sabourin, F.Bortolozzi, C.Y.Suen, Foreground and Background Information in an HMM-Based Method for Recognition of Isolated Characters and Numeral Strings, *9th International Workshop on Frontiers in Handwriting Recognition*, Kokubunji, Tokyo, Japan, 371-376 October 26-29, 2004.

[98] A. K. Jain, R. P. W. Duin, and J. Mao. Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4-37, 2000.

[99] C. Y. Suen, C. Nadal, R. Legault, T. A. Mai, and L. Lam. Computer recognition of unconstrained handwritten numerals. *Proc. of the IEEE*, 80(7):1162–1180, 1992.

[100] P. D. Gader, B. Forester, M. Ganzberger, A. Billies, B. Mitchell, M. Whalen, and T. Youcum, Recognition of handwritten digits using template and model matching, *Pattern Recognition*, 5(24):421-431, 1991.

[101] G. Dimauro, S. Impedovo, G. Pirlo, and A. Salzo. Automatic bankcheck processing: A new engineered system. S.Impedovo et al, editor, *International Journal of Pattern Recognition and Artifcial ntelligence,* World Scientifc, 467-503, 1997.

[102] S. L. Xie and M. Suk. On machine recognition of hand-printed chinese character by feature relaxation. *Pattern Recognition,* 21(1):1-7, 1988.

[103] D. Guillevic and C. Y. Suen. Cursive script recognition applied to the processing of bank cheques. *In Proc. of 3th International Conference on Document Analysis and Recognition,* 11-14, Montreal-Canada, August 1995.

[104] L. Mico and J. Oncina, Comparison of fast nearest neighbour classifier for handwritten character recognition, *Pattern Recognition Letters,* 19(3-4):351-356, 1999.

[105] L. R. Rabiner, A tutorial on hidden markov models and selected applications in speech recognition. *Proc. of IEEE,* 77(2):257-286, 1989.

[106] A. Kundu, Y. He, and M. Chen. Alternatives to variable duration hmm in handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 20(11):1275-1280, 2002.

[107] I. S. I. Abuhaiba and P. Ahmed. A fuzzy graph theoretic approach to recognize the totally unconstrained handwritten numerals. *Pattern Recognition,* 26(9):1335-1350, 1993.

[108] P. D. Gader, J. M. Keller, and J. Cai. A fuzzy logic system for detection and recognition of street number fields on handwritten postal addresses. *IEEE Transactions on Fuzzy Systems,* 3(1):83-95, 1995.

[109] B. Lazzerini and F. Marcelloni. A linguistic fuzzy recognizer of off-line handwritten characters. *Pattern Recognition Letters,* 21(4):319-327, 2000.

[110] N. E. Ayat, M. Cheriet, and C. Y. Suen. Optimization of the SVM kernels using an empirical error minimization scheme. *In Proc. of the International Workshop on Pattern Recognition with Support Vector Machine,* 354-369, Niagara Falls-Canada, August 2002.

[111] H. Byun and S. W. Lee. Applications of support vector machines for pattern recognition. *In Proc. of the International Workshop on Pattern Recognition with Support Vector Machine,* 213-236, Niagara Falls-Canada, August 2002.

[112] Pham Anh Phuong,Ngo Quoc Tao, Luong Chi Mai, speeding up Isolated Vietnamese Handwritten Recognition by combining SVN and statistical features, *International Journal of Computer Sciences and Engineering Systems,* 2(4):258-261, 2008.

[113] H. Y. Kim and J. h. Kim. Handwritten korean character recognition based on hierarchical randon graph modeling. *In Proc. 6th International Workshop on Frontiers of Handwriting Recognition (IWFHR),* 557-586, Taegon-Korea, August 1998.

[114] M. Shridhar and A. Badreldin. Recognition of isolated and simply connected handwritten numerals. *Pattern Recognition,* 19(1):1-12, 1986.

[115] G. P. Zhang. Neural networks for classification: a survey. *IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews,* 30(4):451-462, 2000.

[116] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner,Gradient-based learning applied to document recognition, *Proc. of IEEE,* 86(11):2278-2324, 1998.

[117] B. Zhang, M. Fu, H. Yan, and M. A. Fabri. Handwritten digit recognition by adaptative-subspace self organizing map. *IEEE Trans. on Neural Networks,* 10:939-945, 1999.

[118] O. Matan, J. C. Burges, Y. LeCun, and J. S. Denker, Multi-digit recognition using a space displacement neural network, In J. E. Moody, S. J. Hanson, and R. L. Lippmann, editors, *Advances in Neural Information Processing Systems,* 4, 488-495, Morgan Kaufmann, 1992.

[119] E. Lethelier, M. Leroux, and M. Gilloux. An automatic reading system for handwritten numeral amounts on french checks. *In Proc. 3th International Conference on Document Analysis and Recognition (ICDAR),* 92-97, Montreal-Canada, August 1995.

[120] L. Ling, M. Lizaraga, N. Gomes, and A. Koerich. A prototype for brazilian bankcheck recognition. In S.Impedovo et al, editor, *International Journal of Pattern Recognition and Artificial Intelligence,*World Scientific, ,549-569, 1997.

[121] J. Kittler, M. Hatef, R. Duin, and J. Matas, On combining classifiers, *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 20(3):226-239, 1998.

[122] A. Britto Jr., R. Sabourin, F. Bortolozzi, and C. Y. Suen. A two-stage hmm based system for recognizing handwritten numeral strings. *In Proc. 6th International Conference on Document Analysis and Recognition (ICDAR),*396-400, Seattle-USA, September 2001.

[123] L. S. Oliveira, R. Sabourin, F. Bortolozzi, and C. Y. Suen, Automatic recognition of handwritten numerical strings: A recognition and verification strategy. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 24(11):1438-1454, 2002.

[124] C. Y. Suen, K. Liu, and N. W. Strathy. Sorting and recognizing cheques and financial documents. *In Proc. of 3rd IAPR Workshop on Document Analysis Systems*, 1-18, Nagano-Japan, November 1998.

[125] C-L.Liu,H.Sako, H.Fujisawa, Effects of classifier structures and training regimes on integrated segmentation and recognition of handwritten numeral strings, *IEEE Trans on Pattern Analysis and Machine Intelligence,*26(11):1395-1407.

[126] C-L.Liu, K.Nakashima, H.Sako, H.Fujisawa, Handwritten digit recognition benchmarking of state-of-the art technique, *Pattern Recognition,* 36, 2271-2285, 2003.

[127] G.S.Lehal and Chandan Singh, A Gurumukhi script recognition system, *International Conference on Pattern Recognition,* 2557-2560, 2000.

[128] M. K. Jindal, R. K. Sharma and G. S. Lehal, Structural Features for Recognising Degraded Printed Gurumukhi Script, *Fifth International Conference on Information Technology: New Generations*, 668-673, 2008.

[129] G.S.Lehal , Chandan Singh and Ritu Lehal, A Shape based Post Processor for gurumukhi OCR, **,** *International Conference on Document Analysis and Recognition (ICDAR'2001),*1105-1109, 2001.

[130] B.B Choudhuri, and U. Pal., A Complete Printed Bangla OCR System, *Pattern* ***Recognition,*** vol.31.No.5,pp.531-549,1998.

[131] Sinha R.K, Mahabala , Machine recognition of Devnagari script. *IEEE Trans. Systems Man Cybern.*, pp.435–441, 1979.

[132] B.B Choudhuri, and U.Pal, *Printed Devnagari script OCR system***,** Vivek, vol.10.pp.12 -24, January, 1997.

[133] Veena Bansal, Integrating knowledge sources in Devnagari text recognition*. Ph.D. Thesis*, IIT Kanpur, 1999.

[134] S.N.S Rajasekaran and B.L. Deekshatulu, Recognition of printed Telugu characters., *Computer Graphics Image Processing*, 6, pp.335–360, 1977.

[135] P. V. S Rao. & T. M. Ajitha, Telugu Script Recognition - a Feature Based Approach. *Proceedings of ICDAR, IEEE* pp.323-326, 1995.

[136] Atul Negi, Bhagvati Chakravarthy and B.Krishna ,An OCR system for Telugu. *Proc. Of 6th Int. Conf. on Document Analysis and Recognition IEEE Comp. Soc. Press*, USA, pp. 1110-1114,2001.

[137] K Wong., R Casey and F Wahl, Document analysis system. *IBM Journal of Research and Development*, Vol 26(6), 1982.

[138] G.Nagy, S. Seth, and M. Vishwanathan, A prototype document image analysis system, *Technical journals*, *Computer*, 25(7), 1992.

[139] Pujari Arun K , C Dhanunjaya Naidu & B C Jinaga , An Adaptive Character Recognizer forTelugu Scripts using Multiresolution Analysis and Associative Memory, *ICVGIP*, Ahmedabad, 2002.

[140] Seethalakshmi R., Sreeranjani T.R., Balachandar T.,Abnikant Singh, Markandey Singh, RitwajRatan, Sarvesh Kumar, Optical Character Recognition for printed Tamil text using Unicode. *Journal of Zhejiang University* SCI 6A(11) pgs.1297-1305, 2005.

[141] Anuradha Srinivas, Arun Agarwal, C.R.Rao, Telugu Character Recognition. *Proc. Of International conference on systemics, cybernetics, and informatics*, Hyderabad, pp. 654- 659, 2007.

[142] Ping Zhang, Reliable Recognition of Handwritten Digits Using A Cascade Ensemble Classifier System and Hybrid Features,*PhD Thesis*, Department of Computer Science and Software Engineering, Concordia University, Montreal, Quebec, Canada, April, 2006.

[143] Renu Vig and Tangkeshwar Thokchom, Segmentation and Recognition using Neural networks, WSEAS,ISPRA-IMARS_IEHS,Chiclana,Cadiz,Spain, june12-16,2002.

[144] Th.Tangkeshwar, P.K.Bansal, Renu Vig and H.S.Kasana ,A novel approach to Off-line Handwritten character recognition of Manipuri Script, *National Conference on Bioinformatics Computing, NCBC'05*, TIET, Patiala, pp365-371 March 18-19, 2005.

[145] Th. Tangkeshwar, P.K.Bansal, Renu Vig and H.S.Kasana, Off-line Handwritten Manipuri character recognition using K-L divergence by Probabilistic Model, *National Workshop on Trends in Advanced Computing, NWTAC'06*, Tejpur University, pp163-169,January 23-24, 2006.

[146] Th. Tangkeshwar, P.K.Bansal, Renu Vig and Seema Bawa, Off-line handwritten Character recognition of Meetei (Manipuri) Script, *ManipurInformation Technology Expo 2006, MITEX'06*, Main Stadium, Khuman Lampak, Imphal, Dec7-9, 2006.

[147] Th.Tangkeshwar, P.K.Bansal, Renu Vig and Seema Bawa, Recognition of off-line handwritten digits of Manipuri script using artificial Neural networks, *National Conference on Trends in Advanced Computing*, Tejpur University, pp203-207, March 22-23, 2007.

[148] Th.Tangkeshwar, P.K.Bansal, Renu Vig and Seema Bawa, A Neural Based Recognition Technique for Handwritten Character of Manipuri Script, *International Conference on Automation and Mechatronics, ICAM 2009*, Paper ID code: HK39366, World Congress on Science, Engineering and Technology, *WCSET 2009*, Hong Kong, March 23-25, 2009. (accepted for publication)

[149] Th. Tangkeshwar, P.K.Bansal, Renu Vig and Seema Bawa, Off-line Handwritten Digit Recognition of Manipuri Script, International Journal of Computer Sciences and Engineering Systems, IJCSES,China, vol.4. No.2 April, 2010.

[150] Th. Tangkeshwar, P.K.Bansal, Renu Vig and Seema Bawa, Recognition of Handwritten Character of Manipuri Script, *Journal of Computers (JCP, ISSN 1796-203X)*,Vol.5, No.10, 2010

[151] S.W. Lee. Off-line recognition of totally unconstrained handwritten numerals using multilayer cluster neural network. *IEEE Trans. Patterns Analysis and Machine In - telligence*, 18(6):648–652, 1996.

[152] Mantas, J., An overview of character recognition methodologies. Pattern Recognition, 1986. 19(6): p. 425-430.

[153] Govindan, V. and A. Shivaprasad, Character recognition—a review. Pattern Recognition, 1990. 23(7): p. 671-683.

[154] Jain, A.K. and D. Zongker, Representation and recognition of handwritten digits using deformable templates. Pattern Analysis and Machine Intelligence, IEEE Transactions on,1997. 19(12): p. 1386-1390.

[155] Shridhar, M. and A. Badreldin, A high-accuracy syntacticrecognition algorithm for handwritten numerals. Systems, Man and Cybernetics, IEEE Transactions on, 1985(1): p. 152-158.

[156] Chen Yan & Graham Leedham, "Decompose-Threshold Approach to Handwriting Extraction in Degraded historical Document Images", Proceedings of the 9th International Workshop on Frontiers in Handwriting recognition, pp 239-244, kokubunji, Tokyo, Japan,October, 2004.

[157] J. Sauvola, M. Pietikainen, Adaptive document image binarization, Pattern Recognition 33 (2000) 225–236.

[158] Lawrence O'Gorman, Michael J.Sammon, Michael Seul, Practical Algorithms for Image Analysis:description,examples,programs and projects-2nd ed., Cambridge University Press, 2008.

[159] Muhammad Arif Mohamad, Dewi Nasien, Haswadi Hassan, Habibollah Haron,A Review on Feature Extraction and Feature Selection for Handwritten Character Recognition, International Journal of Advanced Computer Science and Applications, (IJACSA), Vol. 6, No. 2, 2015.

[160] S.Uchida and M. Liwicki. Analysis of Local Features for Handwritten Character Recognition. In Proceedings of the 20th International Conference on Pattern Recognition, pages 1945–1948, Istanbul, Turkey, 2010.

[161] Vamvakas, G., B. Gatos, and S.J. Perantonis, Handwritten character recognition through two-stage foreground sub-sampling. Pattern Recognition, 2010. 43(8): pp. 2807-2816.

[162] Chacko, B., et al., Handwritten character recognition using wavelet energy and extreme learning machine. International Journal of Machine Learning and Cybernetics, 2012. 3(2): p. 149-161.

[163] Wang, X. and A. Sajjanhar, Polar Transformation System for Offline Handwritten Character Recognition, in Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing 2011, R. Lee, Editor. 2011, Springer Berlin Heidelberg. p. 15-24.

[164] [164] Yang, Y., X. Lijia, and C. Chen, English Character Recognition Based on Feature Combination. Procedia Engineering, 2011. 24(0): p. 159-164.

[165] Chel, H., A. Majumder, and D. Nandi, Scaled Conjugate Gradient Algorithm in Neural Network Based Approach for Handwritten Text Recognition, in Trends in Computer Science, Engineering and Information Technology, D. Nagamalai, E. Renault, and M. Dhanuskodi,Editors. 2011, Springer Berlin Heidelberg. p. 196-210.

[166] AlKhateeb, J.H., et al., Offline handwritten Arabic cursive text recognition using Hidden Markov Models and re-ranking. Pattern Recognition Letters, 2011. 32(8): p. 1081-1088.

[167] Rajput, G. and R. Horakeri, Handwritten Kannada Vowel Character Recognition Using Crack Codes and Fourier Descriptors, in Multidisciplinary Trends in Artificial Intelligence, C. Sombattheera, et al.,Editors. 2011, Springer Berlin Heidelberg. p. 169-180.

[168] Sharma, D. and P. Jhajj, Comparison of Feature Extraction Methods for Recognition of Isolated Handwritten Characters in Gurmukhi Script, in Information Systems for Indian Languages, C. Singh, et al., Editors 2011, Springer Berlin Heidelberg. p. 110-116.

[169] Choudhary, A., R. Rishi, and S. Ahlawat, Unconstrained Handwritten Digit OCR Using Projection Profile and Neural Network Approach, in Proceedings of the International Conference on Information Systems Design and Intelligent Applications 2012 (INDIA 2012) held in Visakhapatnam, India, January 2012, S. Satapathy, P.S. Avadhani, and A. Abraham, Editors. 2012, Springer Berlin Heidelberg. p. 119-126.

[170] Li, L., L.-l. Zhang, and J.-f. Su, Handwritten character recognition via direction string and nearest neighbor matching. The Journal of China Universities of Posts and Telecommunications, 2012. 19, Supplement 2(0): p. 160-196.

[171] Nemouchi, S., L. Meslati, and N. Farah, Classifiers Combination for Arabic Words Recognition: Application to Handwritten Algerian City Names, in Image and Signal Processing, A. Elmoataz, et al., Editors. 2012, Springer Berlin Heidelberg. p. 562-570.

[172] Ahmed, I., S. Mahmoud, and M. Parvez, Printed Arabic Text Recognition, in Guide to OCR for Arabic Scripts, V. Märgner and H. El Abed, Editors. 2012, Springer London. p. 147- 168.

[173] Likforman-Sulem, L., et al., Features for HMM-Based Arabic Handwritten Word Recognition Systems, in Guide to OCR for Arabic Scripts, V. Märgner and H. El Abed, Editors. 2012, Springer London. p. 123-143.

[174] Kessentini, Y., T. Paquet, and A. Ben Hamadou, Multi-stream Markov Models for Arabic Handwriting Recognition, in Guide to OCR for Arabic Scripts, V. Märgner and H. El Abed, Editors. 2012, Springer London. p. 335-350.

[175] Bhattacharya, U., et al., Offline recognition of handwritten Bangla characters: an efficient two-stage approach. Pattern Analysis and Applications, 2012. 15(4): p. 445-458.

[176] R**e**ddy, G.S., et al. Combined online and offline assamese handwritten numeral recognizer. in Communications (NCC), 2012 National Conference on. 2012.

[177] Muhammad Naeem Ayyaz, Imran Javed and Waqar Mahmood. Handwritten Character Recognition Using Multiclass SVM Classification with Hybrid Feature Extraction. Pak. J. Engg. & Appl. Sci. Vol. 10, Jan., 2012 (p. 57-67)

[178] Vidya V, Indhu T R, Bhadran V K,R Ravindra Kumar, ”Malayalam Offline Handwritten Recognition using Probabilistic Simplified Fuzzy ARTMAP”, Advances in Intelligent Systems and Computing Volume 182, 2013, pp 273-283.

[179] Primekumar K.P,Sumam Mary Idiculla, ”On-line Malayalam Handwritten Character Recognition using HMM and SVM”, 2013 International Conference on Signal Processing, Image Processing and PatternRecognition[ICSIPR].

[180] S. B. Kotsiantis, Feature selection for machine learning classification problems: a recent Overview, © Springer Science+Business Media B.V. 2011

[181] P.M. Lallican, C. Viard-gaudin, and S. Knerr. From Off-Line To On-Line Hand-writing Recognition. In In Proceedings of the 7th International Workshop on Frontiers in Handwriting Recognition, pages 303–312, Amsterdam, 2000.

[182] https://in.mathworks.com/help/images/ref/bwareaopen.html

[183] Dewi Nasien, Habibollah Haron and Siti S. Yuhaniz. (2010). Metaheuristics Methods (GA& ACO) For Minimizing the Length of Freeman Chain Code from Handwritten Isolated Characters, World Academy of Science Engineering and Technology, Vol. 62, February 2010, ISSN: 2070-3274, Article 41, pp. 230-235

[184] Reza Azmi , Boshra Pishgoo , Narges Norozi , Maryam koohzadi ,Fahimeh baesi,2010. A hybrid GA and SA algorithms for feature selection in recognition of hand-printed Farsi characters.

[185] Nibaran Das, Ram Sarkar, Subhadip Basu, Mahantapas Kundu, Mita Nasipuri, Dipak Kumar Basu.2012. A genetic algorithm based region sampling for selection of local features in handwritten digit recognition application. Applied Soft Computing 12 (2012) 1592–1606.

[186] Abhinaba Roy, Nibaran Das, Ram Sarkar, Subhadip Basu, Mahantapas Kundu, Mita Nasipuri,2012. Region Selection in Handwritten Character Recognition using Artificial Bee Colony Optimization. 2012 Third International Conference on Emerging Applications of **i**nformation Technology (EAIT). pp. 183-186.

[187] Li, L., L.-l. Zhang, and J.-f. Su, Handwritten character recognition via direction string and nearest neighbor matching. The Journal of China Universities of Posts and Telecommunications, 2012. 19, Supplement 2(0): p. 160-196

[188] Nagasundara K B, Guru D S, Manjunath S, 2012. Feature selection and Indexing of Online Signatures.

[189] C. De Stefano, F. Fontanella, C. Marrocco, A. Scotto di Freca,2014. A GA-based feature selection approach with an application to handwritten character recognition. Pattern Recognition Letters 35 (2014) 130-141.

[190] Abhinaba Roy, Nibaran Das, Ram Sarkar, Subhadip Basu, Mahantapas Kundu, Mita Nasipuri,2014. An Axiomatic Fuzzy Set Theory Based Feature Selection Methodology for Handwritten Numeral Recognition. ICT and Critical Infrastructure: Proceedings of the 48th Annual Convention of Computer Society of India- Vol I Advances in Intelligent Systems and Computing Volume 248, 2014, pp 133-140.

[191] F. Ghareh Mohammadi, M.Saniee Abadeh, 2014. Imagesteg analysis using a bee colony based feature selection algorithm. Engineering Applications of Artificial Intelligence 31(2014)35-43.

[192] A. Marcano-Cedeño, J. Quintanilla-Domínguez, M.G. Cortina-Januchs and D. Andina, Feature Selection Using Sequential Forward Selection and classification applying Artificial Metaplasticity Neural Network, 2010.

[193] Dewi Nasien, Habibollah Haron, Siti Sophiayati Yuhaniz. (2011). The Heuristic Extraction Algorithm for Freeman Chain Code of Handwritten Character. International Journal of Experimental Algorithms (IJEA). Publisher: CSC Press, Computer Science Journals, Volume: 1, Issue: 1, pp. 1-20, ISSN: 2180-1282.

[194] Gheith Abandah and Nasser Anssari. Novel Moment Features Extraction for Recognizing Handwritten Arabic Letters Journal of Computer Science 5 (3): 226-232, 2009**,** ISSN 1549-3636. Science Publications

[195] Jyh-Shing Roger Jang,Chuen-Tsai Sun and Eiji Mizutani,*Neuro-Fuzzy and Soft Computing*, Prentice-Hall of India, Private Limited, New Delhi,2002.

[196] M. Riedmiller, H. Braun, A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm, University of Karlsruhe, 1993.

[197] Thornton, J.R., Fachney, J., Blumenstein, M., Nguyen, V & Hine,T. Offline cursive character recognition: A state-of-the-art comparison. Proceedings of the 14th Conference of the International Graphonomics Society IGS 2009, Dijon, France, September 13-16, 2009.

[198] Thornton, J.R., Fachney, J., Blumenstein, M., Nguyen, V & Hine,T. Character Recognition using Hierarchical Vector Quantization and Temporal Pooling. AI 2008 Advances in Artificial Intelligence, 21st Australasian Joint Conference on Artificial Intelligence, Auckland, New Zealand, December 3-5 2008.

[199] Chunpeng Wu, Wei Fan, Yuan He, Jun Sun, Satoshi Naoi, Handwritten Character Recognition by Alternately Trained Relaxation Convolutional Neural Network, 14th International Conference on Frontiers in Handwriting Recognition, 2014.

[200] G. Vamvakas , B. Gatos and S.J. Perantonis, "Hierarchical Classification of Handwritten Characters based on Novel Structural Features", 11th International Conference on Frontiers in Handwriting Recognition(ICFHR'08), pp, 535-539, Montreal,Canada, August 2008.

[201] Abdeljalil Gattal , Chawki Djeddi , Youcef Chibani , Imran Siddiqi, Isolated Handwritten Digit Recognition Using oBIFs and Background Features , 12th IAPR Workshop on Document Analysis Systems (DAS) , pp:305- 310, 11-14 April 2016.

[202] Jomy John, Kannan Balakrishnan, K. V. Pramod, "A system for offline recognition of handwritten characters in Malayalam script", International Journal of Image, Graphics and Signal Processing (IJIGSP), MECS-Press, Volume 5, Issue 4, 2013, Pages 53-59, ISSN 2074-9074, DOI: 10.5815/ijigsp. 2013.04.07.

[203] U. Pal, A. Belaid, B. B. Chaudhuri, A System for Bangla Handwritten Numeral Recognition, IETE Journal of Research, March 2015.

[204] Jomy John, K. V. Pramod, Kannan Balakrishnan, Bidyut B. Chaudhuri, "A two stage approach for handwritten Malayalam character recognition", Frontiers in Handwriting Recognition (ICFHR- - 2014), September, 1-4, 2014, Crete, Greece.

[205] Siddharth, K.S., et al., Handwritten Gurmukhi character recognition using statistical and background directional distribution features. International Journal on Computer Science and Engineering, 2011. 3(6): p. 2332-2345.
[206] Singh, S., A. Aggarwal, and R. Dhir, Use of Gabor Filters for Recognition of Handwritten Gurmukhi Character. International Journal of Advanced Research in Computer Science and Software Engineering, 2012. 2(5).
[207] Pathan, I.K., A.A. Ali, and R. RJ, Recognition of offline handwritten isolated urdu character. Advances in Computational Research, ISSN, 2012. 4(1): p. 117-121.
[208] Soman, S.T., A. Nandigam, and V.S. Chakravarthy. An efficient multiclassifier system based on convolutional neural network for offline handwritten Telugu character recognition. in Communications (NCC), National Conference on. IEEE.
[209] Rajput, G. and R. Horakeri. Shape descriptors based handwritten character recognition engine with application to Kannada characters. in Computer and Communication Technology (ICCCT), 2nd International Conference on. 2011. IEEE
[210] Jangid, M., Devanagari Isolated Character Recognition by using Statistical features.International Journal on Computer Science and Engineering (IJCSE) ISSN, 2011. 3.
[211] Shelke, S. and S. Apte, A multistage handwritten Marathi compound character recognition scheme using neural networks and wavelet features. International Journal of Signal Processing, Image Processing and Pattern Recognition, 2011. 4(1): p. 81-94.
[212] Mohammad Haghighat, Saman Zonouz, and Mohamed Abdel-Mottaleb, Identification Using Encrypted Biometrics, R. Wilson et al. (Eds.): CAIP 2013, Part II, LNCS 8048, pp. 440–448, 2013. © Springer-Verlag Berlin Heidelberg 2013.

List of Publications

International Journals:

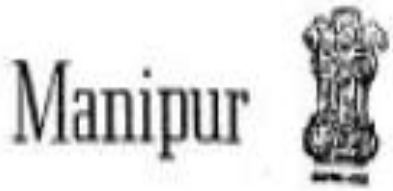
1. Th.Tangkeshwar, P.K.Bansal, Renu Vig and Seema Bawa, "Off-line Handwritten Digit Recognition of Manipuri Script", published in the *International Journal of Computer Sciences and Engineering Systems*, *IJCSES*, ISSN 0973-4406, Vol.4. No.2 April, 2010, a fully refreed, a peer-reviewed, International Journal published quarterly by Serials publications, India.
2. Th.Tangkeshwar, P.K.Bansal, Renu Vig and Seema Bawa , "Recognition of Handwritten Character of Manipuri Script", publised in *Journal of Computers* (*JCP*, ISSN 1796-203X), Vol.5, No.10, 2010, Peer-reviewed open access International Scientific Journal published monthly by Academy Publisher, Finland.

National Journals/Conferences/Workshops:

1. Th.Tangkeshwar, P.K.Bansal, Renu Vig and H.S.Kasana, "A novel approach to off-line Handwritten character recognition of Manipuri Script", published and presented in the *National Conference on Bioinformatics Computing, NCBC'05*, TIET, Patiala, pp365-371, March 18-19, 2005.
2. Th.Tangkeshwar, P.K.Bansal, Renu Vig and H.S.Kasana, "Off-line Handwritten Manipuri character recognition using K-L divergence by Probabilistic Model", published and presented in the *National Workshop on Trends in Advanced Computing*, *NWTAC'06*, Tejpur University, Napaam,Assam,pp163-169, January 23-24, 2006.
3. Th.Tangkeshwar, P.K.Bansal, Renu Vig and Seema Bawa,"Off-line handwritten Character recognition of Meetei (Manipuri) Script", published and presented in the *Manipur Information Technology Expo 2006, MITEX'06*, Main Stadium, Khuman Lampak, Imphal, Dec 7-9, 2006.
4. Th.Tangkeshwar, P.K.Bansal, Renu Vig and Seema Bawa, ,"Recognition of off-line handwritten digits of Manipuri script using artificial Neural networks", published in *the National Conference on Trends in Advanced Computing*, Tejpur University, pp203-207, March 22-23, 2007.

APPENDIX 1

MANIPUR GAZETTE


Manipur Gazette

EXTRAORDINARY
PUBLISHED BY AUTHORITY

No. 33 Imphal, Tuesday, April 22, 1980 (Vaisakha 3, 1902)

GOVERNMENT OF MANIPUR

SECRETARIAT: EDUCATION DEPARTMENT

ORDERS BY THE GOVERNOR OF MANIPUR
Imphal, the 16th April, 1980

No. 1/2/78-SS/E.—The Governor, Manipur is pleased to approve the report submitted by the Meitei Mayek Expert Committee constituted vide Government order of even number dated 16th November, 1978 on the re-introduction of the study of Meitei Scripts numbering 27 (Twentyseven) alphabets and its supplement (uses of Lonsum, Cheitap-Cheikhei, Khudam & Cheising etc) as per annexures as recommended by the committee in the educational institutions in Manipur.

By order and in the name of Governor,

J. K. SANGLURA,
Secretary to the Government of Manipur.

Manipur Gazette

EXTRAORDINARY

PUBLISHED BY AUTHORITY

No. 40 Imphal, Friday, April 25, 1980 (Vaisakha 5, 1902)

GOVERNMENT OF MANIPUR

SECRETARIAT: EDUCATION DEPARTMENT

CORRIGENDUM

Imphal, the 24th April, 1980

No. 1/2/78-SS/E.—Please read "MEETEI" for "MEITEI" occurring in Government Orders of even number dated 16th April, 1980.

By order and in the name of the Governor,

J. K. SANGLURA,

Secretary (Education) to the Government of Manipur.