

Recognition of Handwritten Devanagari Script Using Soft Computing

A Thesis submitted in partial fulfillment of the requirements
for the award of degree of

Master of Engineering
in
Software Engineering

By:
Divya Sharma
(80731006)

Under the supervision of

Mr. Karun Verma
Lecturer, CSED



COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
THAPAR UNIVERSITY
PATIALA – 147004

JUNE-2009

CERTIFICATE

I hereby certify that the work which is being presented in the thesis entitled, **“Recognition of Handwritten Devanagari Script using Soft Computing”**, in partial fulfillment of the requirements for the award of degree of Master of Engineering in Software Engineering submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of Mr. Karun Verma and refers other researcher’s works which are duly listed in the reference section.

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

(Divya Sharma)

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

(Mr. Karun Verma)

Lecturer,
Computer Science and Engineering Department,
Thapar University, Patiala

Countersigned by

(SEEMA BAWA)

Professor & Head,
Computer Science & Engineering Department,
Thapar University,
Patiala.

(R.K.SHARMA)

Dean (Academic Affairs)
Thapar University,
Patiala.

ACKNOWLEDGEMENT

The real spirit of achieving a goal is through the way of excellence and austere discipline. I would have never succeeded in completing my task without the cooperation, encouragement and help provided to me by various personalities.

First of all, I render my gratitude to the almighty who bestowed self-confidence, ability and strength in me to complete this work. Without his grace this would never come to be today's reality.

With deep sense of gratitude I express my sincere thanks to my esteemed and worthy supervisor **Mr. Karun Verma** in the Department of Computer Science and Engineering for his valuable guidance in carrying out this work under his effective supervision, encouragement, enlightenment and cooperation. Most of the novel ideas and solutions found in this thesis are the result of our numerous stimulating discussions. His feedback and editorial comments were also invaluable for writing of this thesis.

I shall be failing in my duties if I do not express my deep sense of gratitude towards **Dr. Seema Bawa**, Professor and Head of Computer Science and Engineering Department who has been a constant source of inspiration for me throughout this work.

I am grateful to **Dr. R.K. Sharma**, Dean of Academic Affairs for his constant encouragement that was of great importance in the completion of the thesis.

I am also thankful to all the staff members of the Department for their full cooperation and help.

My greatest thanks are to all who wished me success especially my parents and friends whose support and care make me stay on earth.

Place: TU, Patiala

Date:

(Divya Sharma)

ABSTRACT

Development of a Character recognition system for Devnagri is difficult because (i) there are about 350 basic, modified (“*matra*”) and compound character shapes in the script and (ii) the characters in a words are topologically connected. Here focus is on the recognition of offline handwritten Hindi characters that can be used in common applications like bank cheques, commercial forms, government records, bill processing systems, Postcode Recognition, Signature Verification, passport readers, offline document recognition generated by the expanding technological society. Handwriting has continued to persist as a means of communication and recording information in day-to-day life even with the introduction of new technologies. Challenges in handwritten characters recognition lie in the variation and distortion of offline handwritten Hindi characters since different people may use different style of handwriting, and direction to draw the same shape of any Hindi character. This overview describes the nature of handwritten language, how it is translated into electronic data, and the basic concepts behind written language recognition algorithms.

Handwritten Hindi character are imprecise in nature as their corners are not always sharp, lines are not perfectly straight, and curves are not necessarily smooth, unlikely the printed character. Furthermore, Hindi character can be drawn in different sizes and orientation in contrast to handwriting which is often assumed to be written on a baseline in an upright position. Therefore, a robust offline Hindi handwritten recognition system has to account for all of these factors.

An approach using Artificial Neural Network is considered for recognition of Handwritten Hindi Character Recognition.

CONTENTS

Certificate	i
Acknowledgement.....	ii
Abstract.....	iii
Contents.....	iv
List of Figures.....	vi
List of Tables.....	viii
Chapter 1. Introduction.....	1
1.1 Devanagari Script.....	1
1.1.1 Devanagari Script Identification.....	3
1.1.2 Devanagari Script Overview.....	3
1.1.3 Issues in Development Devanagari Character System.....	5
1.2 Optical Character Recognition	6
1.2.1 Application of Character Recognition.....	8
1.2.2 Limitation.....	10
1.2.3 Different Phases of Handwritten Hindi Character Recognition.....	10
1.2.3.1 Pre-processing	11
1.2.3.2 Segmentation	12
1.2.3.3 Feature Extraction and Classification.....	13
1.2.3.4 Post- processing	13
1.3 Types of Character Recognition System	14
1.3.1 Online Character Recognition.....	15
1.3.2 Offline Character Recognition.....	15
Chapter 2. Literature Review.....	17
2.1 Classification	21
2.1.1 Coverage of the Region of the Core Strip.....	21
2.1.2 Vertical Bar Feature.....	22

Chapter 3. Problem Statement	30
3.1 Problems in Character Recognition of Devanagari Script.....	30
3.2 Problem definition.....	30
3.2.1 Background.....	31
3.2.1.1 Graphic Files.....	31
3.2.1.2 Pixel.....	32
3.2.1.3 True Color.....	33
3.2.1.4 Palette/Color Map.....	33
3.2.1.5 Color model.....	33
3.2.1.6 Resolution.....	33
3.2.1.7 Windows Bitmap Format.....	34
3.3 Development of a HDCR System.....	34
3.3.1 Offline Handwritten Image samples.....	35
Chapter 4. Implementation	38
4.1 Neural Network	38
4.1.1 Neuron model.....	39
4.1.1.1 The Biological Neuron.....	39
4.1.1.2 Simple Neuron.....	40
4.2 Neural Network Architecture.....	41
4.3 Working of Neural Networks.....	42
4.3.1 Modes of Learning.....	43
4.3.2 Testing.....	44
4.4 Neural Network Topologies.....	44
4.4.1 Feed-forward Neural Networks.....	44
4.4.2 Recurrent Neural Networks	44
4.5 Advantages of Neural Computing.....	45
4.6 Limitations of Neural Computing.....	45
4.7 Application of Neural Network.....	46
4.8 MATLAB.....	47
4.9 Recognition of Handwritten Hindi Characters.....	48
4.9.1 Load Image.....	48

4.9.2 Selection of Character.....	49
4.9.3 Preprocessing.....	50
4.9.4 Feature Extraction.....	51
4.9.5 Neural Network Training.....	52
4.9.6 Testing.....	53
Chapter 5.Results.....	54
Chapter 6.Conclusion.....	56
6.1 Conclusion.....	56
6.2 Future Scope.....	57
References.....	58
Paper-Published.....	62

LIST OF FIGURES

Figure 1: Different zones of Devanagari text.....	3
Figure 2: Non compound Devanagari Characters.....	4
Figure 3: Devanagari Modifiers.....	4
Figure 4: Vowels and corresponding modifiers	4
Figure5: Consonant.....	4
Figure 6: Half form of Consonant with vertical bar	5
Figure 7: Example of combination of half consonant and consonant	5
Figure 8: Classes of Core Hindi Character	5
Figure 9: same character	6
Figure 10: Different Character that looks similar	6
Figure 11: Basic Devanagari Characters	6
Figure12: Block Diagram of HDCR.....	10
Figure13: correction of Skewness	12
Figure14: Post processing	14
Figure15: Type of character recognition system	14
Figure16: Hierarchy of character recognition problems	18
Figure17: Characters and Symbols of Devanagari Script	19
Figure18: Three strips of a Devanagari word	20
Figure 19: The procedure of Hindi character segmentation	20
Figure 20: classification of character	22
Figure 21: classification of character	22
Figure 22: Complete handwriting recognition system.....	27
Figure 23: Some of the touching characters used in Devanagari script.....	28
Figure 24: Line segmentation using horizontal projection profile.....	28
Figure 25: Baseline Identification.....	29
Figure 26: same character.....	30
Figure 27: Different Character that looks similar.....	30
Figure 28: Samples of Offline Handwritten Devanagari Character Images , , ...	35

Figure 29: Samples of Offline Handwritten Devanagari Character Images , , ...	35
Figure 30: Samples of Offline Handwritten Devanagari Character Images , , ...	35
Figure 31: Samples of Offline Handwritten Devanagari Character Images , , ...	35
Figure 32: Samples of Offline Handwritten Devanagari Character Images , , ...	36
Figure 33: Samples of Offline Handwritten Devanagari Character Images , , ...	36
Figure 34: Samples of Offline Handwritten Devanagari Character Images , , ...	36
Figure 35: Samples of Offline Handwritten Devanagari Character Images , , ...	36
Figure 36: Samples of Offline Handwritten Devanagari Character Images , , ...	37
Figure 37: Samples of Offline Handwritten Devanagari Character Images , , ...	37
Figure38:Samplesof Offline Handwritten Devanagari Character Images , , , ...	37
Figure 39: Block Diagram of Neural Network.....	38
Figure 40: A biological Neuron.....	40
Figure 41: Neuron.....	41
Figure 42: Architecture of Neural network showing different Layers.....	42
Figure 43: Load Button.....	48
Figure 44: Character Image.....	48
Figure 45: Select Button.....	49
Figure 46: Selected Character from Image.....	49
Figure 47: Bounding Box of Characters.....	50
Figure 48: Cropped Character '२'.....	51
Figure 49: Filled character '२'.....	51

Figure 50: Extracted features of character ‘२’	52
--	----

LIST OF TABLES

Table 1: Comparison between online and offline Handwritten characters	16
Table 2: Recognition Accuracy of Handwritten Hindi Characters	54-55

Hindi handwritten character recognition is the one of the major problem in today's world. Typed Hindi characters can be easily recognized by computer machine. But Hindi handwritten characters are not recognized efficiently and accurately by computer machine. Many researches have been done to recognize these characters and many algorithms have been proposed to recognize characters. Many types of software are in the market for optical Hindi character recognition. For recognizing characters, many processes have to be performed. No single process or single machine can perform that recognition. Artificial neural networks can be used for recognition of characters due to the simplicity of their design and their universality. Hindi character recognition is becoming more and more important in the modern world. It helps human ease their jobs and solve more complex problems. The problem of recognition of hand-printed characters is still an active area of research. With ever increasing requirement for office automation, it is imperative to provide practical and effective solutions. It has been observed that all sorts of structural, topological and statistical information about the characters does not lend a helping hand in the recognition process due to different writing styles and moods of persons at the time of writing. Mainly, attention is focused on recognition of hand-printed Hindi characters. Limited variations in shapes of character are considered.

For more than 30 years, researchers have been working on handwritten recognition. Over the few past years, the numbers of companies involved in research on handwritten recognition are increasing continually.

Handwritten recognition is not a new technology, but it has not gained public attention until recently. The ultimate goal of designing a handwritten recognition system with an accuracy rate of 100% is quit illusionary, because even human beings are not able to recognize every handwritten text without any confusion. It can be seen that most of the people can not even read their own notes. Therefore there is an obligation for a writer to write clearly.

1.1 Devanagari Script

Hindi is world's third most commonly used language after Chinese and English, and there are approximately 500 billion people all over the world who speak and write in Hindi. It is the basic script of many languages in India, such as Hindi and Sanskrit.

Many other languages use close variants of this script. Although Sanskrit is an ancient language and no longer spoken, written material still exist. It is very expressive language, which has been influenced and enriched by Dravidian, Turkish, Farsi, Arabic, Portuguese and English. Thus research on Devanagari script mainly Hindi language attracts a lot of interest.

The script Devanagari originally belongs to Brahmi which is considered purely Indian in nature. In the fourth century, from the northern branch of Brahmi Gupta script was developed. Subsequent Kutil came out of Gupta script and Nagri script developed out of Kutil in 8Th and 9Th century. The ancient Nagri script gave birth to modern Nagri, Gujrati, Rajasthani, Mathil and Bangla scripts. Later on this modern Nagri script came to be known as Devanagari. Acharya Vinoba Bhave calls it LokNagri. According to him this script is used not only by one religion, caste or creed, rather it has become the script of whole nation and of the common people. Some people call it Hindi dialect but it does not seem fair because not only Hindi but many other languages are written in this script. Hindi is a language whereas Devanagari is a script.

There are a few opinions about the name “Devnagari”:

1. It was called Nagri for being prevalent in Nagars and Sanskrit was called voice of Devas, so Nagri was called Devanagari.
2. It was called Devanagari due to its excessive use in Brahmins of Gujrat.
3. Another viewpoint is that it was prevalent in Devnagar area of Kashi, hence it was named as Devanagari.

It is indisputable that Devanagari has the most accurate scientific basis. For a long time, it has been script of Indian Aryan languages. It is even now used by Sanskrit, Hindi, Marathi and Nepali languages. Hindi is the world’s widely spoken language and since it’s script is Devanagari, so it’s the most popular script. As Hindi has been declared the national language by constitution of Indian, Devanagari has got the status of national dialect.

In the beginning, Hindi was declared as the state language and Devanagari as the state script of other major states such as Himachal, Haryana, Rajasthan, Madhya Pradesh, Bihar, Uttaranchal, etc. Presently, it is found that Devnagari script is the most scientific script. Since every script is developed from Brahmi script so, Devnagari has connection with almost every other script. In Devanagari, all letters are equal, i.e.

there is no concept of capital or small letters. Devanagari is half syllabic in nature [37].

1.1.1 Devanagari Script Identification

Instead of describing what type of features can be used to identify Devanagari script words from document images, we examine the appearance of Devanagari script. Regular Hindi word can be divided into three Zones i.e. Upper, Middle, Lower Zone. Example is shown below where three Zones are illustrated. The Upper Zone and middle Zone are always separated by the header line called shirorekha. The Upper Zone contains the modifiers, and Lower Zone contains lower modifiers. In Hindi word, Upper and Lower Zone are not always necessary, but depend on Upper and lower modifiers.

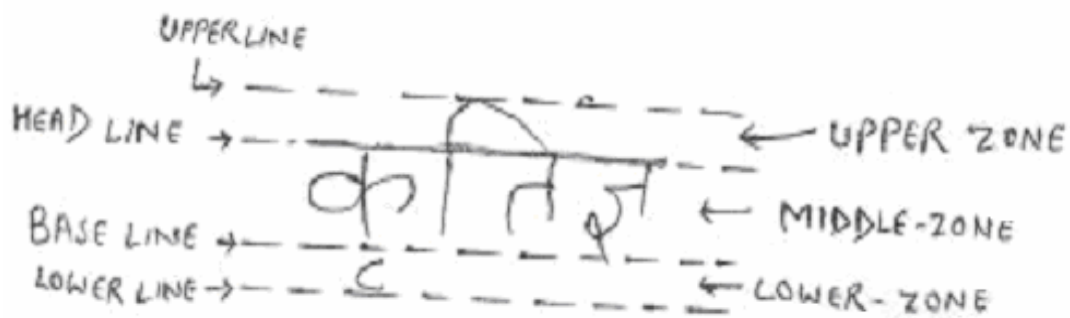


Figure 1: Different zones of Devanagari text

1.1.2 Devanagari Script Overview

Its basic set of symbols consists of 34 consonants and 18 vowels, and though Devanagari has a native set of symbols for numerals, Arabic numbers are now commonly used. Character Recognition for Devanagari is highly complex due to its rich set of conjuncts.

Devanagari is written from left to right along a horizontal line. Its basic set of symbols consists of 34 consonants or ('*vyanjan*') and 18 vowels ('*svar*'). Characters are joined by a horizontal bar that creates an imaginary line by which Devanagari text is suspended, and no spaces are used between words. A single or double vertical line called '*Purn Viram*' was traditionally used to indicate the end of phrase or sentence. Devanagari also has a native set of symbols for numerals, though Arabic numbers are typically used.

Here is a sample set of non-compound Devanagari characters. [15]

Figure 2: Non compound Devanagari Characters

١٠٠

Obviously, these modifiers make Character Recognition with Devanagari script very challenging. HDCR system is further complicated by compound characters that make character separation and identification very difficult.

Figure 4: Vowels and corresponding modifiers

Figure 5: Consonant

क	ख	ग	घ		ङ		च	छ	ज	झ	
			ट	ठ	ड		ढ	ण	त	थ	द
ध	न	प	फ		ब	भ	म	य	र	ल	व

Figure 6: Half form of Consonant with vertical bar

क कक्क	क लक्ल	घ नघ्न	च जच्च	झ बझ्व	त नत्त	प तप्त	प लप्त
ब वव्व	भ नभ्न	म लम्ल	ल लल्ल	श नश्न	श बश्ब	श लशल	स नस्न

Figure 7: Example of combination of half consonant and consonant

Open Header	अ थ ध भ
One Conjunction End Bar	च ज्ञ अ त न व ख
More Conjunctions End Bar	स घ झ प म य ष स ख
Middle Bar	ऋ क फ कृ क्र
No Bar	इ उ ऊ ए ऋ छ ट ठ ड ढ द र ह ङ ण
Special Case	ग ण श ण

Figure 8: Classes of Core Hindi Character

1.1.3 Issues in Development of Devanagari Character System–

a) Complexity of Devanagari Scripts –

- There is Variability for same characters.
- All the individual characters are joined by a head line called “*Shiro Rekha*” in case of Devanagari Script. This makes it difficult to isolate individual characters from the words.
- There are various isolated dots, which are vowel modifiers, namely, “*Anuswar*”, “*Visarga*” and “*Chandra Bindu*”, which add up to the confusion.
- Ascenders and DESCENDER recognition is also complex, attributed to the complex nature of language.
- It contains Composite characters.
- Minor variations in similar characters.
- It contains large number of character and stroke classes.

Devanagari characters are shown in Fig 11.



Figure 9: same character



Figure 10: Different Character that looks similar

b) Bi-Lingual Nature of Text

- In the scenario of a country like India, where there is a influence of many other European languages, like English, French and Portuguese, having these languages mixed in the text is inevitable.
- Apart from the European languages, India itself has fourteen official languages, which could also be found embedded in the text matter.

अ आ इ ई उ ऊ ऋ ॠ ए ऐ ओ औ अं अँ अः ऐँ औँ
क ख ग घ ङ च छ ज झ ञ ट ठ ड ढ ण त थ द ध न
प फ ब भ म य र ल व श ष स ह ल

Figure 11: Basic Devanagari Characters

1.2 Optical Character Recognition (OCR)

OCR is the acronym for Optical Character Recognition. This technology allows a machine to automatically recognize characters through an optical mechanism. Human beings recognize many objects in this manner. Eyes are the “optical mechanism” while the brain “sees” the input, the ability to comprehend these signals varies in each person according to many factors. By reviewing these variables, the challenges faced by the technologist developing an OCR system can be understood easily. Firstly, for reading a page in unknown language, anyone may be unable to recognize the various characters. But on the same page, numerical statements can be easily interpreted

because the symbols for numbers are universally used. This explains why many OCR systems recognize numbers only, while relatively few understand the full alphanumeric character range. Second, there is similarity between many numerical and alphabetical symbol shapes. For example, while examining a string of characters combining letters and numbers, there is very little visible difference between a capital letter “O” and the numeral “0”. Humans can re-read the sentence or entire paragraph to determine the accurate meaning. This procedure, however, is much more difficult for a machine.

Third, contrast helps in recognizing characters. It is very difficult to read text which appears against a very dark background, or is printed over words or graphics. Again, programming a system to interpret only the relevant data and disregard the rest is a difficult task for OCR engineers.

Documents are in the form of papers which the human can read and understand but it is not possible for the computer to understand these documents directly. In order to convert these documents into computer processable form, OCR systems are developed. OCR is the process of converting scanned images of machine printed or handwritten text, numerals, letters and symbols into a computer processable format such as ASCII. OCR is an area of pattern recognition and processing of handwritten character is motivated largely by desire to improve man and machine communication. Few products are currently commercially available for character recognition. Products to perform handwritten recognition are not available, though many approaches have been proposed. In fact, there has recently been a high level of interest in applying artificial neural network architecture to solve this problem. Most of the development in neural network research during the past decade made use of either pattern matching or statistical approaches for feature extraction. 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 may be even consciousness.

Much research on automatic text processing has been done based on OCR. OCR of Indian scripts is in preliminary stage. Much of the research work has been done for developing OCR systems in Roman scripts. Compared to this; extensive research and development activities are required for developing OCR systems for Indian scripts. The popularity of OCR is increasing each year with the advent of fast microprocessors providing the vehicle for vastly improved recognition techniques.

There has been a tremendous improvement in increasing both effective read rates and accuracy. Data Entry through OCR is faster, more accurate, and generally more efficient than keystroke data entry. Desktop OCR scanners can read typewritten data into a computer at rates up to 2400 words per minute!

1.2.1 Application of Character Recognition System

There are number of applications of Character Recognition System:

• Task-specific Readers

Task – specific readers are used primarily for high-volume applications which require high system throughput. Since high throughput rates are desired, handling only the fields of interest helps reduce time constraint. Since similar document possess similar size and layout structure, it is straight forward for the image scanner to focus on those fields where the desired information lies. This approach can considerably reduce the image processing and text recognition time. Some application areas to which task-specific readers have been applied include:

- Assigning ZIP codes to letter mail.
- Reading data entered in forms, e.g. tax forms
- Verification of account numbers and courtesy amounts on bank checks
- Automatic accounting procedure used in processing utility bills
- Automatic accounting of airline passenger tickets
- Automatic validation of passports

• Address Readers

The address reader in a postal mail sorter locates the destination address block on a mail piece and reads the ZIP code in this address block. If additional fields in the address block are read with high confidence the system may generate a 9 digit ZIP code for the piece. The resulting ZIP code is used to generate a bar code which is sprayed on the envelope.

The Multi line Optical Character Reader (MLOCR) used by the United States Postal Services (USPS) locates the address block on a mail piece, reads the whole address, identifies the ZIP+4 code generates 9-digit bar code and sorts the mail to the correct stacker. The character classifier recognizes up to 400 fonts and the system can process up to 45,000 mail pieces per hour.

- **Form Reader**

A form reading system needs to discriminate between pre-printed form instructions and filled-in data. The system is first trained with a blank form. The system registers those areas on the form where the data should be printed. During the form recognition phase, the system uses the spatial information obtained from training to scan the region that should be filled with data. Some readers read hand-printed data as well as various machine written texts. They can read data on a form without being confused with the form instructions. Some systems can process forms at a rate of 5,800 forms per hour.

- **Check Reader**

A check reader captures check image and recognize courtesy amounts and accounts information on the checks and use the information in both fields to cross check the recognition result. An operator can correct misclassified characters by cross-validating the recognition results with the check image that appears on a system console.

- **Bill Processing System**

In general a bill processing system is used to read payment slips, utility bills and inventory documents. The system focuses on certain region on a document where the expected information are located, e.g. account number and payment value.

- **Passport Readers**

An automated passport reader is used to speed up the returning American passengers through custom inspection. The Reader reads a traveller's, date of birth and passport number on the passport and checks these against the database records that contain information on fugitive felons and smugglers.

- **General Purpose Page Readers**

There are two general categories of page reader: high-end page readers and low-end page readers. High-end page readers are more advanced in recognition capability and higher data throughput than the low-end page readers. A low-end page reader usually does not come with a scanner and it is compatible with many flat-bed scanners.

They are mostly used in an office environment with desktop work stations, which are less demanding in system throughput. Since they are designed to handle a broader range of documents, a sacrifice of recognition accuracy has to be made. Some

commercial OCR software allow users to adapt the recognition engine to customer data for improving recognition accuracy.

1.2.2 Limitation

OCR has never achieved a read rate that is 100% perfect. Because of this, a system which permits rapid and accurate correction of rejects is a major requirement. Exception item processing is always a problem because it delays the completion of the job entry, particularly the balancing function. In particular, the system does not accurately balance dollar data. The success of any OCR device to read accurately without substitution is not the sole responsibility of the hardware manufacturer. Much depends on the quality of the items to be processed. The main purpose of OCR from many years is as follows:

- To increase the accuracy of reading, that is , to reduce rejects and substitution
- To eliminate the need for specially designed fonts (character), and to handwritten characters.
- To reduce sensitivity of scanning to read less-control input

The above limits are not objectionable to most application and dedicated users of OCR system are growing each year. But the ability to read a special character is not, by itself, sufficient to create a successful system. In fact OCR is a time saver, but it is not perfect.

- It rarely reaches more than 99.9% level of accuracy.
- It faces problem with early printed books, newspaper, etc.
- It faces problems with heavily bound material.

1.2.3 Different Phases of Handwritten Hindi Character Recognition

The process of a HHCR of Devanagari script can be divided into phases as shown in Figure11. Each phase has been explained below:

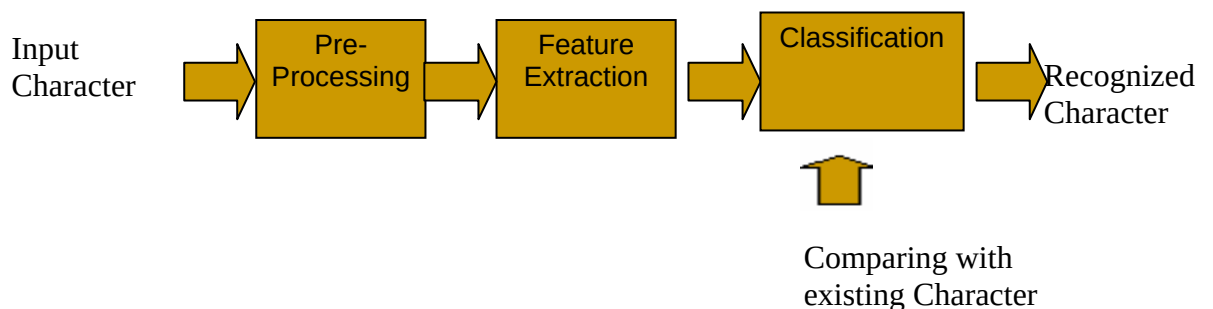


Figure 12: Block Diagram of HDCR

1.2.3.1 Pre-processing

Pre-processing is the name given to a family of procedures for smoothing, enhancing, Filtering, cleaning-up and otherwise massaging a digital image so that subsequent algorithm along the road to final classification can be made simple and more accurate.

Various Pre-processing Methods are explained below:

a) Binarization

Document image binarization (thresholding) refers to the conversion of a gray-scale image into a binary image. Two categories of thresholding:

- Global, picks one threshold value for the entire document image which is often based on an estimation of the background level from the intensity histogram of the image.
- Adaptive (lopcal), uses different values for each pixel according to the local area information

b) Noise Removal

The major objective of noise removal is to remove any unwanted bit-patterns, which do not have any significance in the output.

c) Skeletonization

Skeletonization is also called thinning. Skeletonization refers to the process of reducing the width of a line like object from many pixels wide to just single pixel. This process can remove irregularities in letters and in turn, makes the recognition algorithm simpler because they only have to operate on a character stroke, which is only one pixel wide. It also reduces the memory space required for storing the information about the input characters and no doubt, this process reduces the processing time too.

d) Smoothing

The objective of smoothing is to smooth contours of broken and/or noisy input characters.

e) Contour Smoothing

The objective of contour smoothing is to smooth contours of broken and/or noisy i Skewness input characters.

f) Skewness

Skewness refers to the tilt in the bitmapped image of the scanned paper for character recognition system. It is usually caused if the paper is not fed straight into the scanner.

Most of the character recognition algorithms are sensitive to the orientation (or skew) of the input document image, making it necessary to develop algorithms which can detect and correct the skew automatically. as example of skewness is shown below.

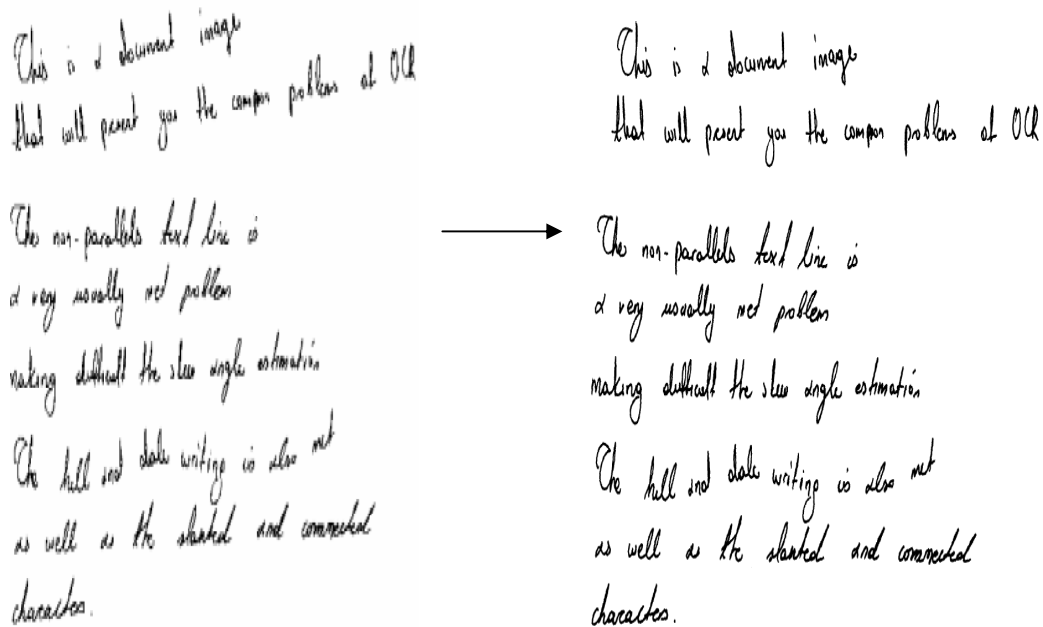


Figure 13: correction of Skewness

After pre-processing phase, a cleaned image is obtained that goes to the segmentation phase.

1.2.3.2 Segmentation

It is an operation that seeks to decompose an image of sequence of characters into sub images of individual symbols. Character segmentation is a key requirement that determines the utility of conventional Character Recognition systems. It includes line, word and character segmentation.

Different methods used can be classified based on the type of text and strategy being followed like recognition-based segmentation and cut classification method.

After scanning the document, the document image is subjected to pre-processing for background noise elimination and skew correction to generate the bit map image of the text. The pre-processed image is then segmented into lines, words and characters.

1.2.3.3 Feature extraction and Classification

Character Recognition system consists of two stages, feature extraction and classification. Feature extraction is the name given to a family of procedures for measuring the relevant shape information contained in a pattern so that the task of classifying the pattern is made easy by a formal procedure.

The feature extraction stage analyses a text segment and selects a set of features that can be used to uniquely identify the text segment. The selection of a stable and representative set of features is the heart of pattern recognition system design.

Classification stage is the main decision making stage of the system and uses the features extracted in the previous stage to identify the text segment according to preset rules. Classification is concerned with making decisions concerning the class membership of a pattern in question. The task in any given situation is to design a decision rule that is easy to compute and will minimize the probability of misclassification relative to the power of feature extraction scheme employed. Patterns are thus transformed by feature extraction process into points in d dimensional feature space. A pattern class can then be represented by a region or subspace of the feature space. Classification then becomes a problem of determining the region of feature space in which an unknown pattern falls.

1.2.3.4 Post-processing

System results usually contain errors because of character classification and segmentation problems. For the correction of recognition errors, OCR systems apply contextual post-processing techniques. The two most common post-processing techniques for error correction are dictionary lookup and statistical approach. The advantage of statistical approach over dictionary-based methods is computational time and memory utilization. Conversely, lexical knowledge about entire words is more accurate when using a dictionary [3].

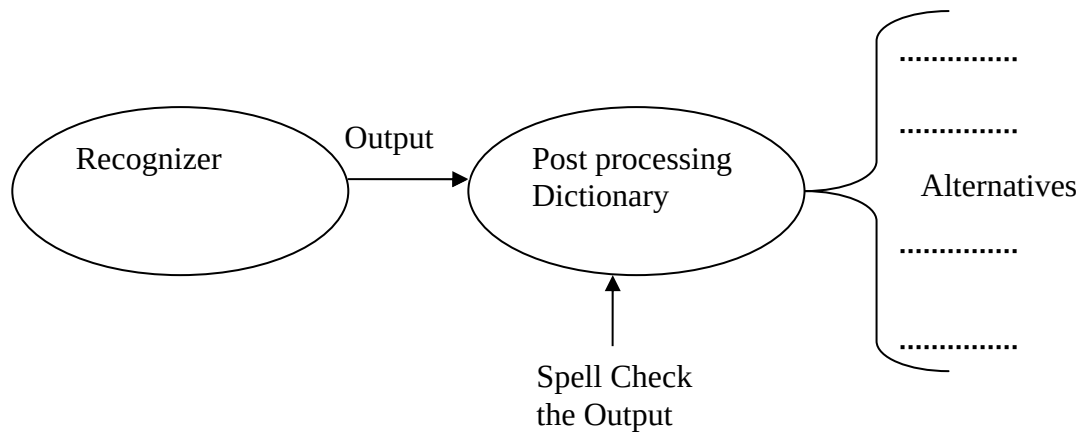


Figure 14: Post processing

The simplest way of incorporating the context information is the utilization of a dictionary for correcting the minor mistakes

1.3 Types of Character Recognition System

The constant development of computer tools leads to a requirement of easier interfaces between the man and the computer. Character Recognition is one way of achieving this. A Character Recognition deal with the problem of reading offline handwritten character i.e. at some point in time (in mins, sec, hrs) after it has been written. However recognition of unconstrained handwritten text can be very difficult because characters cannot be reliably isolated especially when the text is cursive handwriting.

They are classified as the following two types

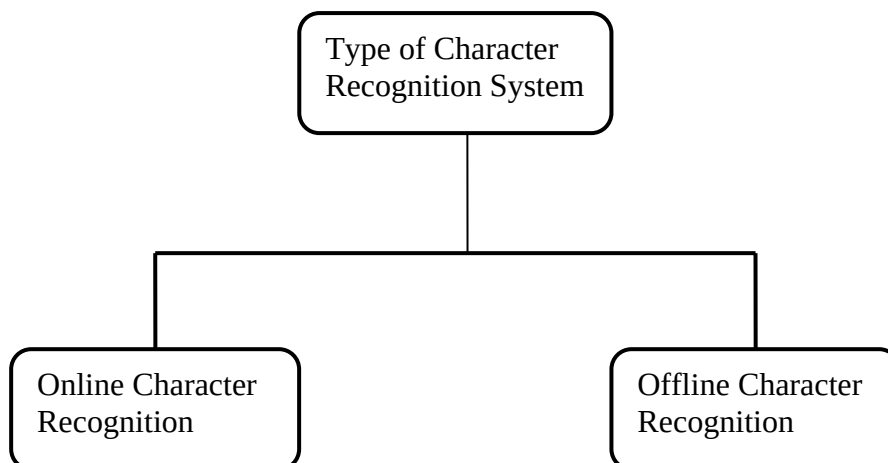


Figure 15: Type of character recognition system

1.3.1 Online Character Recognition

In case of online character recognition, there is real time recognition of characters. Online systems have better information for doing recognition since they have timing information and since they avoid the initial search step of locating the character as in the case of their offline counterpart. Online systems obtain the position of the pen as a function of time directly from the interface. Offline recognition of characters is known as a challenging problem because of the complex character shapes and great variation of character symbols written in different modes.

1.3.2 Offline Character Recognition

In case of offline character recognition, the typewritten/handwritten character is typically scanned in form of a paper document and made available in the form of a binary or gray scale image to the recognition algorithm. Offline character recognition is a more challenging and difficult task as there is no control over the medium and instrument used. The artifacts of the complex interaction between the instrument medium and subsequent operations such as scanning and binarization present additional challenges to the algorithm for the offline character recognition. Therefore offline character recognition is considered as a more challenging task than its online counterpart.

The steps involved in character recognition after an image scanner optically captures text images to be recognized is given to the recognition algorithm.

The major difference between Online and Offline Character Recognition is that Online Character Recognition has real time contextual information but offline data does not [38]. This difference generates a significant divergence in processing architectures and methods.

Table 1: Comparison between online and offline handwritten characters

Sr No.	Comparisons	On-line characters	Off-line characters
1.	Availability of no. of pen-strokes	Yes	No
2.	Raw Data Requirement	# samples/second(e.g. 100)	# dots/inch(e.g. 300)
3.	Way of writing	Using digital pen on LCD surface	Paper document
4.	Recognition Rates	Higher	Lower
5.	Accuracy	Higher	Lower

In 1929, Gustav Tauschek obtained a patent on OCR in Germany, followed by Handel who obtained a US Patent on OCR in USA in 1933 (U.S. Patent 1,915,993). In 1935 Tauschek was also granted a US patent on his method (U.S. Patent 2,026,329). In 1950, David Shepard, a cryptanalyst at the Armed Force Security Agency in the United States, with the help of Harvey Cook founded Intelligent Machines Corporation (IMR), which went on to deliver the world's first several OCR systems used in commercial operation. IBM and others were later licensed on Shepard's OCR patents. The United States Postal Services has been using OCR machines to sort mail since 1965 based on technology devised primarily by the prolific inventor Jacob Rainbow. In 1965 it began planning an entire banking system, National Giro, using OCR technology, a process that revolutionized bill payment systems in the UK. Canada Post has been using OCR systems since 1971. OCR systems read the name and address of the addressee at the first mechanized sorting centre, and print a routing bar code on the envelope based on the Postal Code. After that the letters need only be sorted at later centres by less expensive sorters which need only read the code. To avoid interference with the human-readable address field which can be located anywhere on the letter, special ink is used that is clearly visible under ultraviolet light. This ink looks orange in normal lighting conditions. Envelopes marked with the machine readable bar code may then be processed.

During these days Handwriting recognition, including recognition of hand printing, cursive handwriting, is still the subject of active research, as is recognition of printed text in other scripts. Recognition of cursive text is an active area of research, with recognition rates even lower than that of hand-printed text. Higher rates of recognition of general cursive script will likely not be possible without the use of contextual or grammatical information. For example, recognizing entire words from a dictionary is easier than trying to parse individual characters from script. Reading the *Amount* line of a cheque (which is always a written-out number) is an example where using a smaller dictionary can increase recognition rates greatly. Knowledge of the grammar of the language being scanned can also help determine if a word is likely to be a verb or a noun, for example, allowing greater accuracy. The shapes of individual cursive

characters themselves simply do not contain enough information to accurately (greater than 98%) recognize all handwritten cursive script.

The field of Document Analysis and Recognition is vast and it contains many applications. Character recognition is one of the branches of DAR. As shown in Figure 16, the problem of character recognition can be divided into printed and handwritten character recognition. Handwritten character recognition has been further divided into off-line and online handwritten character recognition [1]. Off-line handwriting recognition refers to the process of recognising words that have been scanned from a surface (such as a sheet of paper) and are stored digitally in grey scale format. After being stored, it is conventional to perform further processing to allow superior recognition. In the on-line case, the handwriting is captured and stored in digital form via different means. Usually, a special pen is used in conjunction with an electronic surface. As the pen moves across the surface, the two dimensional coordinates of successive points are represented as a function of time and are stored in order [1]. It is generally accepted that the on-line method of recognising handwritten text has achieved better results than its 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. On the other side machine-printed character recognition can be on good quality documents or degraded printed documents.

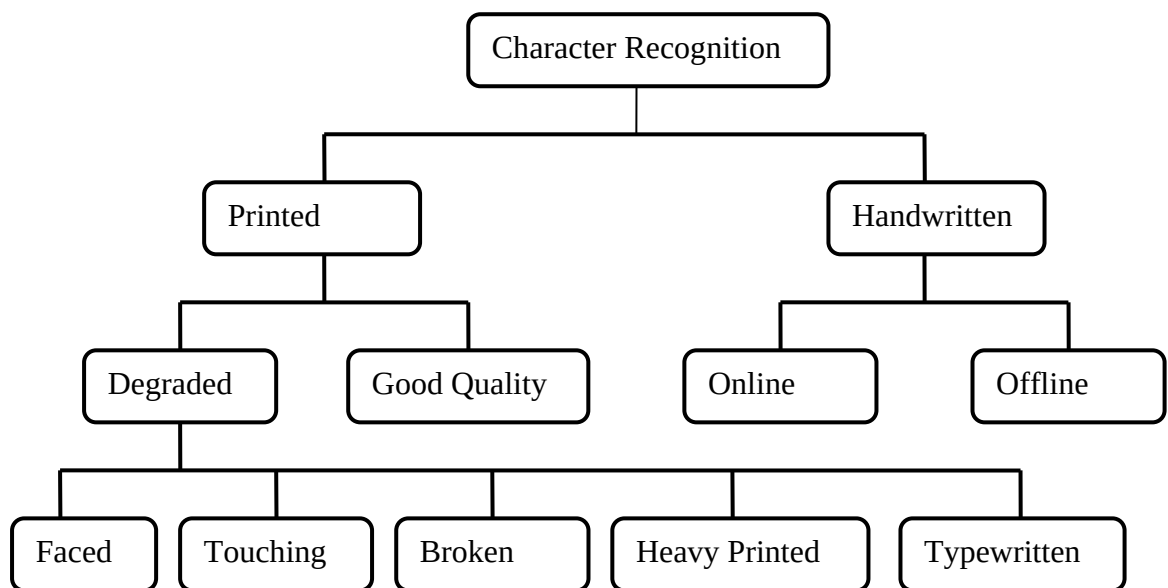


Figure 16: Hierarchy of character recognition problems

Some relevant features of Devanagari script from OCR viewpoint Devanagari script have about 11 vowels and 33 consonants. Some of the vowels and the consonants are shown in figure 17(a) and figure 17(d) respectively. An appropriate modifier is attached from symbols shown in figure 17(b) in an appropriate manner to the consonant. Each modifier has been attached to the first consonant of the script क (see figure 17(c)). A visual inspection of figure 17(c) reveals that some of the modifier symbols are placed next to the consonant (core modifiers), some above (top modifiers) and some are placed below (lower modifiers) the consonant. Some of the modifiers contain a core modifier and a top modifier, the core modifier is placed before or next to the consonant; the top modifier is placed above the core modifier. Devanagari script has a pure form for most of the consonants. A consonant in pure form always touches the next character, yielding conjuncts, touching characters, or fused characters. Figure 17(e) shows some of the conjuncts formed by writing pure form consonants followed by consonant य. We can use almost any consonant in place of 'य and write over 100 conjuncts.

- (a) अ आ इ ई उ ऊ
- (b) ा ि ी ु ू
- (c) क का कि की कु कू
- (d) क ख ग घ ङ
च छ ज झ ञ
ट ठ ड ढ ण
त थ द ध न
प फ ब भ म
य र ल व
श ष स ह
- (e) क्य ख्य च्य ज्य त्य ध्य

Figure 17: Characters and Symbols of Devanagari Script; [8]

- **Composition of Characters and Symbols for Writing Word**

A horizontal line is drawn on top of all characters of a word that is referred to as the header line or *shirorekha*. It is convenient to visualize a Devanagari word in terms of three strips: a core strip, a top strip and a bottom strip. The core and top strips are separated by the header line. Figure 18 shows the image of a word that contains five characters, two lower modifiers and a top modifier. The three strips and the header line have been marked.

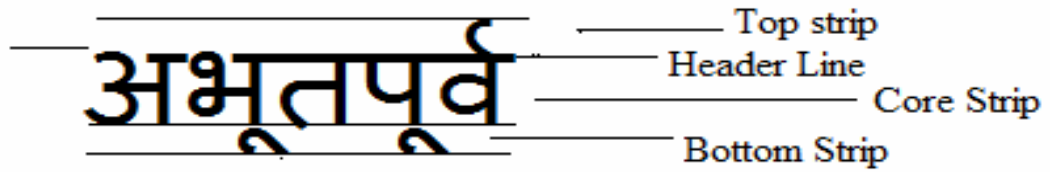


Figure 18: Three strips of a Devanagari word

Huanfeng Ma, David Doermann [2] proposed an algorithm to recognize CHP (collapsed horizontal projection) characters. The procedure to segment a Hindi word into characters (including core characters, and top and bottom modifiers) is illustrated in Figure 19 using the segmentation of the Hindi word. The numbered arrow in Figure 19 represents the step of segmentation, and the characters with solid bounding boxes are the final segmentation results. The procedure to do character segmentation can be described as follows:

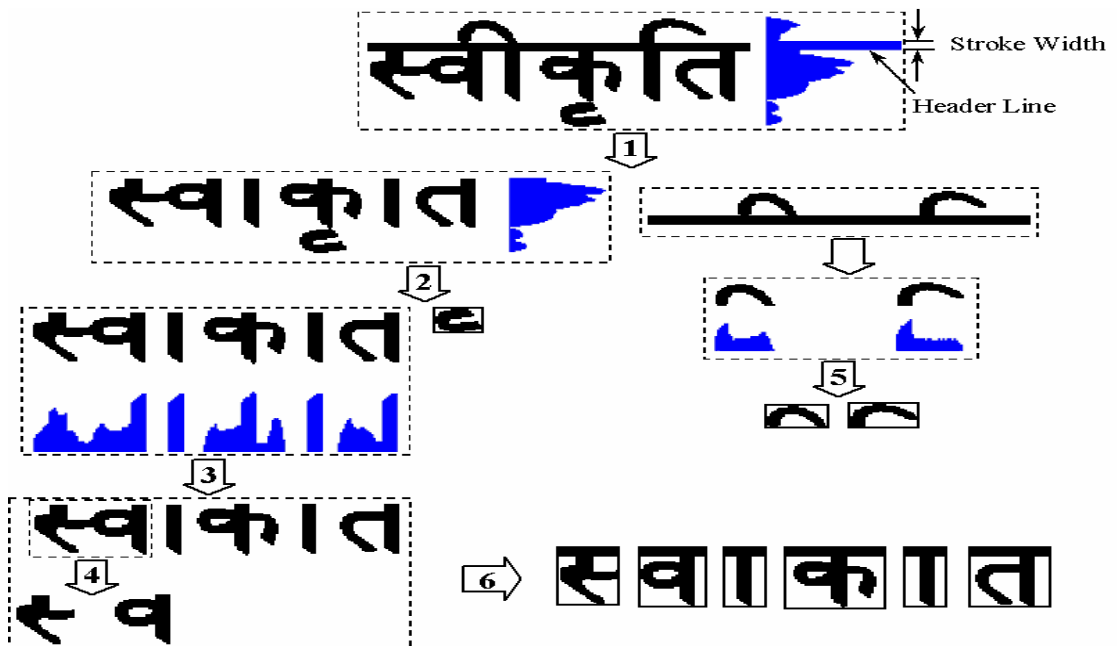


Figure 19: The procedure of Hindi character segmentation [2]

Step 1: Locate the header line and separate the core-bottom strip which contains the core strip and bottom strip, and a top strip which contains the header line and top modifiers.

Step 2: Identify core strip and bottom strip from the core-bottom strip, and extract low modifiers.

Step 3: Separate core strip into characters which may contain conjunct/shadow characters.

Step 4: Segment conjunct/shadow characters into single characters.

Step 5: Remove the header line from the top strip and extract top modifiers.

Step 6: Put header line back to the segmented core character

- **Segmentation of the conjunct character**

The location of the segmentation column contains two steps. First, segmentation is located by examining the right part of the conjunct image. Then second segmentation is located by examining the left part of the conjunct image. The final segmentation is determined by co-relating both segmentations. It can be done by two types i.e. one is by horizontal segmentation and other is vertical segmentation. In vertical segmentation, upper part and lower part of character are recognized. In Devnagari, we recognize the character by segmenting the upper layer of line and lower part of line. In vertical segmentation, character is recognized by recognition based technique [9].

2.1 Classification

2.1.1 Coverage of the Region of the Core Strip

Character set of Devanagari script is divided into three groups based on the coverage of the region of the core strip. The characters which cover most of the core region are referred to as Full box characters. The characters which cover upper region of the core strip are referred to as Upper Half Box characters. Lower Half Box characters are the characters which cover lower region of the core strip. These sets are shown in figure 20.

Full box characters

कखगघङचछजझञटठडढणतथदधनपफबभमयरलवशषसहख्वङ्फप्

Upper half box characters

गद्यध्वज्य

Lower half box characters

चरुतल

Figure: 20 classification of character

2.1.2 Vertical Bar Feature

The Full Box characters are divided into three groups based on the presence and position of vertical bars, namely: no bar characters, end bar characters and middle bar characters. Some of the characters belonging to each of these classes are shown in figure 21. A vertical bar does not occur at the left end of a character. The position of the vertical bar is the left most column where number of black pixels is 80 percent or more of the character height. Character image is divided into three equal vertical zone and compute a vertical bar does not occur at the left end of a character.

End bar characters

अ ख घ च ज झ ञ त थ ध न प ब भ म य ल व श ष स

Middle bar characters

ऋ क फ

No bar characters

इ उ ऊ ए छ ट ठ द ड ढ र ह

Figure: 21 classification of character

OCR work on printed Devanagari script started in early 1970s. Among the earlier pieces of work, some of the efforts on Devanagari character recognition are done by Sinha and Mahabala [3]. A syntactic pattern analysis system and its application to Devanagari script recognition is discussed in his doctoral thesis.

Among the other pieces of work on Devanagari character recognition, Sinha and Mahabala [3] presented a syntactic pattern analysis system with an embedded picture language for the recognition of handwritten and machine printed Devanagari characters. The system stores structural description for each symbol of the Devanagari script in terms of primitives and their relationships. For recognition, an input character

is labelled and compared it with stored description. To increase the accuracy of the system and reduce the computational costs, contextual information regarding the occurrences of certain primitives and their combinations and restrictions are used.

Sinha [3] also demonstrated how the spatial relationship among the constituent symbols of Devanagari script plays an important role in the interpretation of Devanagari words. There are a number of constraints on these spatial relationships which characterize Devanagari script composition syntax. When the word composition is not found to be syntactically correct, the symbols are substituted with their resembling counterparts. The symbol substitution rules are mostly heuristic in nature.

Sethi and Chatterjee [4] also have done some earlier studies on Devanagari script. On the basis of presence or absence of some basic primitives, namely, horizontal line segment, vertical line segment, left and right slant, D-curve, C-curve, etc. and their positions and interconnections, they presented a Devanagari hand-printed numeral recognition system based on binary decision tree classifier. They also used a similar technique for constrained hand-printed Devanagari character recognition. Here, a set of very simple primitives is used, and all the Devanagari characters are looked upon as a concatenation of these primitives. A multi-stage decision process is used where most of the decisions are based on the presence/absence or positional relationship of the primitives.

The systems stated above deal with recognizing characters in isolation. They did not show results of scanning on real document pages. For the purpose some standard techniques have been used and some new ones have been proposed by them. When two or more characters are combined to form a word in Devanagari, the characters in the word normally generate a long line, called head-line. Segmentation of characters from words becomes troublesome because of this head-line. Here, a simple head-line deletion approach is used to segment the characters for the word. Also, a simple approach for dividing a text line into three horizontal zones is used for easier recognition procedure. From zonal information and shape characteristics, the basic, modified and compound characters are separated for the convenience of classification. Modified and basic characters are recognized by a structural feature based binary tree classifier while the compound characters are recognized by a hybrid approach combined with structural and run based template features. The method proposed by Pal and Chaudhuri gives about 96% accuracy.

Recently, a system for hand-written numeral recognition of Devanagari characters is proposed [4]. Here the numerals have been represented using two types of features. The first type provides coarse shape classification of the numeral and is relatively insensitive to minor changes in character shapes. The second class of features tries to provide qualitative descriptions of the characters. These descriptions encode intrinsic properties of the characters expected to be invariant across writing styles and fonts. Multilayer perceptron is used for the categorization of the numerals.

Most Indian languages are very inflectional in nature. Because of this inflectional behaviour, development of OCR error detection and correction technique is not an easy task. The complex character grapheme structure of some Indian scripts also creates difficulty in recognition error detection and correction. An OCR error correction scheme for the Devanagari text is proposed by Bansal and Sinha [9]. They used a partitioned word dictionary to reduce the search space besides preventing forced match to incorrect word. The envelope information of words consisting of number of top, lower, core modifiers along with the number of core characters form the second level partitioning feature for short words partition. The remaining words are further partitioned using a string of fixed length associated with each partition. A distance matrix for assigning penalty for a mismatch is incorporated in the search process.

The ability to identify machine printed characters in an automated or a semi-automated manner has obvious applications in numerous fields. Since creating an algorithm with a one hundred percent correct recognition rate is quite probably impossible in our world of noise and different font styles, it is important to design character recognition algorithms with these failures in mind so that when mistakes are inevitably made, they will at least be understandable and predictable to the person working with the program. Eric W. Brown [5] explores one such algorithm and tests it on two different fonts using a third font as a reference. The results are discussed and several improvements are suggested.

He describes an algorithm that attempts to work with a subset of the features in a character that a human would typically see for the identification of machine-printed English characters. Its recognition rate is currently not as high as the recognition rates of the older, more developed character recognition algorithms, but it is expected that if it were expanded to work with a larger set of features this problem would be removed. If it were expanded to use more features, it would be made correspondingly

slower; with the advent of faster microprocessors this fact is not viewed as a crippling problem. The procedure for extracting these feature points utilized by this algorithm is fairly straightforward. Since an eight by eight character consists of only sixty-four pixels, it is viable to simply loop through the entire character and examine each pixel in turn. If a pixel is on, its eight neighbours are checked. Since each neighbour can also only be on or off, there are merely 256 possible combinations of neighbourhoods. Of these 256, fifty-eight were found to represent significant feature points in a fairly unambiguous way. Extracting feature points thus reduced to calculating a number between zero and 256 to describe a pixel's neighbourhood and then comparing that number against a table of known feature points. While it is true that this method does not always catch every feature point (some can only be seen in a larger context) it catches the majority. Missing feature points is certainly not a limiting factor in the algorithm's accuracy. It also does not suffer from labelling too many uninteresting points as being feature points. It has virtually no false positives. The feature point extractor is thus fast and reliable.

A system for recognition of online handwritten characters has been presented for Indian writing systems. A handwritten character is represented as a sequence of strokes whose features are extracted and classified. Support vector machines have been used for constructing the stroke recognition engine. The results have been presented after testing the system on Devanagari and Telugu scripts [14]. The objective of SVM support vector machine capable of learning is to achieve good generalization performance, given a finite amount of training data, by striking a balance between the goodness of fit attained on a given training dataset and the ability of the machine to achieve error-free recognition on other datasets. With this concept as the basis, support vector machines have proved to achieve good generalization performance with no prior knowledge of the data. The principle of an SVM is to map the input data onto a higher dimensional feature space nonlinearly related to the input space and determine a separating hyperplane with maximum margin between the two classes in the feature space[33]. This results in a nonlinear boundary in the input space. The optimal separating hyperplane can be determined without any computations in the higher dimensional feature space by using kernel functions in the input space.

An SVM in its elementary form can be used for binary classification. It may, however, be extended to multi class problems using the one-against-the-rest approach or by using the one-against-one approach.

Sandhya Arora [15] presents a two stage classification approach for handwritten Devanagari characters. The first stage is using structural properties like shirorekha, spine in character and second stage exploits some intersection features of characters which are fed to a feed forward neural network. Simple histogram based method does not work for finding shirorekha, vertical bar (Spine) in handwritten Devnagari characters. So a new technique, differential distance based technique to find a near straight line for shirorekha and spine. This approach has been tested for 50000 samples and got 89.12% success.

A system for recognizing hand written Indian Devnagari script is presented by K. Y. Rajput and Sangeeta Mishra. The system considers a handwritten image as an input, separates the lines, words and then characters step by step and then recognizes the character using artificial neural network approach, in which Creating a Character Matrix and a corresponding Suitable Network Structure is key. In addition, knowledge of how one is Deriving the Input from a Character Matrix must first be obtained before one may proceed. Afterwards, the Feed Forward Algorithm gives insight into the entire working of a neural network; followed by the Back Propagation Algorithm which comprises Training, Calculation of Error, and Modifying Weights. Once the characters are recognized they can be replaced by the standard fonts to integrate information from diverse sources [17].

A new intelligent segmentation technique is proposed that may be used in conjunction with a neural classifier and a simple lexicon for the recognition of difficult handwritten words. A heuristic segmentation algorithm is initially used to over-segment each word. An Artificial Neural Network (ANN) trained with 32,034 segmentation points is then used to verify the validity of the segmentation points found. Following segmentation, character matrices from each word are extracted, normalised and then passed through a global feature extractor after which a second ANN trained with segmented characters is used for classification. These recognised characters are grouped into words and presented to a variable-length lexicon that utilises a string processing algorithm to compare and retrieve words with highest confidences. This research provides promising results for segmentation, character and word recognition [13]. In the proposed word recognition system, heuristic and

intelligent methods are used for the segmentation of real world, handwritten words. Following segmentation, character matrices are extracted from the words and classified. Finally, to show how the segmentation technique may possibly be used in the context of an overall system, a lexicon is used to match each set of recognised characters (each set represents a single word) to potential correct words. The entire system is shown in following figure.

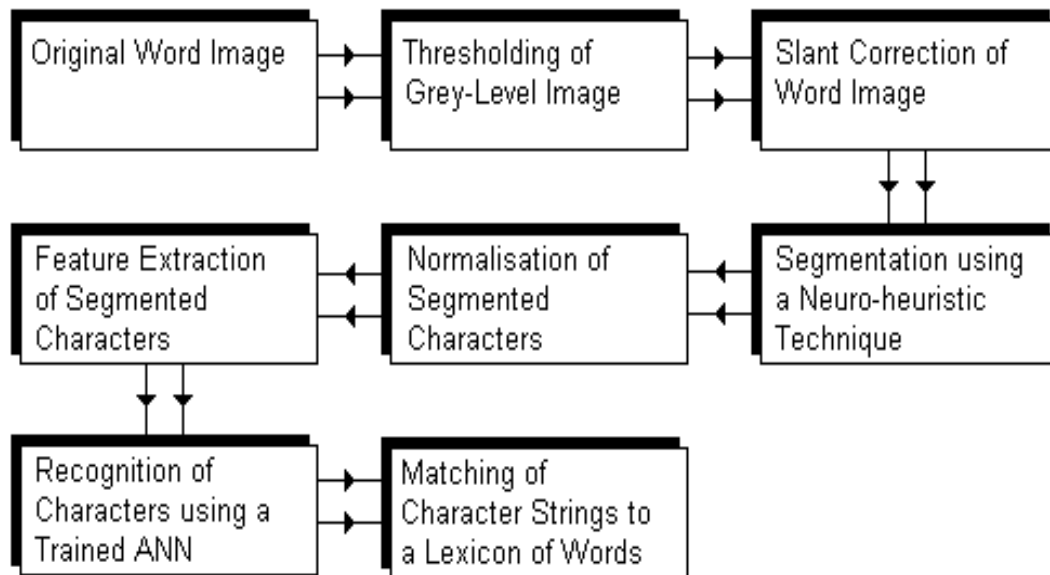


Figure 22: Complete handwriting recognition system [13]

Artificial Neural Networks (ANNs) have been successfully applied to Optical Character Recognition (OCR) yielding excellent results. A method is used for segmentation of difficult handwriting with the use of conventional algorithms in conjunction with ANNs. The segmentation algorithm is heuristic in nature detecting important features which may represent a prospective segmentation point. An Artificial Neural Network is subsequently used to verify the authenticity of the segmentation points found by the algorithm. The C programming language, the SP2 supercomputer and a SUN workstation were used for the experiments. The algorithm has been tested on real-world handwriting obtained from the CEDAR database [16].

An algorithm for segmentation of touching Devanagari characters (also referred to as conjuncts) into its constituent symbols and characters proposed by Veena Bansal and R. M. K. Sinha [9] Proposed algorithm extensively uses structural properties of the script. Statistical information about the height and width of character boxes, which are vertically separate from their neighbours, is used to hypothesize character boxes to be

touching character boxes. The recognition rate of 85% has been achieved on the segmented touching characters.

Touching characters	Constituent characters	
ष	ब	ए
न्ह	न	ह
स्व	स	व
स्न	स्	न
न्त	न्	त
न्म	न्	म

Figure 23: Some of the touching characters used in Devanagari script.

S. Hewavitharana [34] describes a system to recognize handwritten Tamil characters using a two-stage classification approach, for a subset of the Tamil alphabet. In the first stage, an unknown character is pre-classified into one of the three groups: core, ascending and descending characters. Then, in the second stage, members of the pre-classified group are further analyzed using a statistical classifier for final recognition. A recognition rate of 80% was achieved for the 1st choice and 97% for the top 3 choices. Two-stage classification approach is used, which is a hybrid of structural and statistical techniques. In the first stage, known as preliminary classification, the unknown character is classified into one of three groups of Tamil characters. Structural properties of the text line are used for this classification. In the second stage, a statistical classifier recognizes the unknown character as one of the members of the pre-classified group.

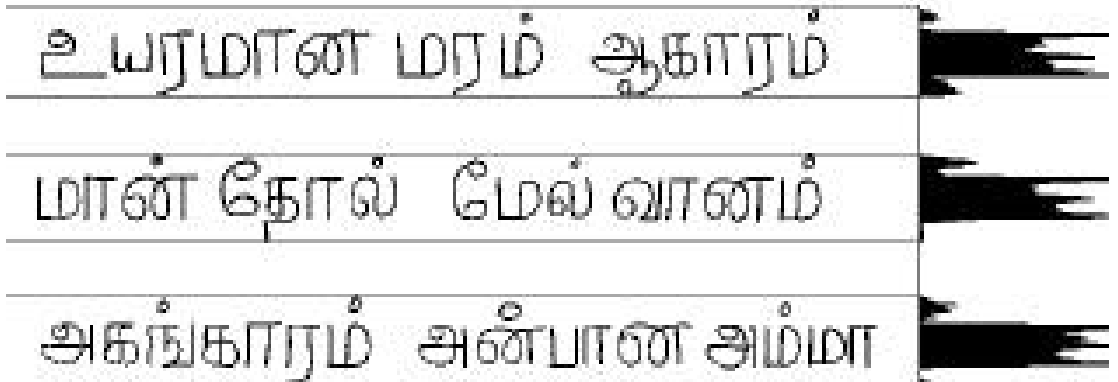


Figure 24: Line segmentation using horizontal projection profile [34]



Figure 25: Baseline Identification [34]

Devanagari is a script used for several major languages such as Hindi, Sanskrit, Marathi and Nepali, and is used by more than 500 million people. Unconstrained Devanagari writing is more complex than English cursive due to the possible variations in the order, number, direction and shape of the constituent strokes. An on-line pen computing environment has numerous applications in providing an easy human interface for a complex script like Devanagari. A Devanagari character recognition experiment with 20 different writers with each writer writing 5 samples of each character in a totally unconstrained way, has been conducted. An accuracy of 86.5% with no rejects is achieved through the combination of multiple classifiers that focus on either local on-line properties, or global off-line properties. Further improvements in performance are expected by using word-level contextual information. We are also exploring the use of writer dependent models to improve the recognition accuracy [25].

Online handwriting recognition is gaining renewed interest owing to the increase of pen computing applications and new pen input devices. The recognition of Chinese characters is different from western handwriting recognition and poses a special challenge. To provide an overview of the technical status and inspire future research, Cheng-Lin Liu reviews the advances in online Chinese character recognition (OLCCR), with emphasis on the research works from the 1990s. The target of recognition has shifted from regular script to fluent script in order to better meet the requirements of practical applications. The research works are reviewed in terms of pattern representation, character classification, learning/adaptation, and contextual processing. We compare important results and discuss possible directions of future research [21].

3.1 Problems in Character Recognition of Devanagari script:

a) Complexity of Devanagari Scripts –

- There are various isolated dots, which are vowel modifiers, namely, “Anuswar”, “Visarga” and “Chandra Bindu”, which add up to the confusion.
- Composite characters
- Variability for same character
- Minor variations in similar characters
- Large number of character and stroke classes
- Ascenders and Descenders recognition is also complex, attributed to the complex nature of language.



Figure 26: same character



Figure 27: Different Character that looks similar

b) Bi-Lingual Nature of Text –

- In the scenario of a country like India, where there is a influence of many other European languages, like English, French and Portuguese, having these languages mixed in the text is inevitable.
- Apart from the European languages, India itself has fourteen official languages, which could also be found embedded in the text matter.

3.2 Problem Definition

Here the problem of handwritten Hindi characters is dealt. This corresponds to the ability of human beings to recognize such characters, which they are able to do little

or no difficulty. Following are main objectives to produce a system that can recognize Hindi Characters:-

- To classify a given input as belonging to a certain class rather than to identify them uniquely, as every input pattern.
- To develop algorithms and data structures for handwritten Hindi characters.
- To develop software, based on the above algorithms, which will convert any scanned and pre-processed document of handwritten Devanagari Script to machine readable form.

In order to accomplish these objectives, a comprehensive study of various methods used in the development of OCR systems for different Indian and other scripts has been carried out.

The system performs character recognition by exploring a neural network paradigm for its ability to recognize handwritten Hindi characters.

The scope of the proposed system is limited to the recognition of a single character. In the next section some background concepts are introduced that are necessary to understand the proposed system.

3.2.1 Background

3.2.1.1 Graphic Files

A Graphic file is a file containing a picture that may be a line or scanned photograph. Any program that displays or manipulates stored images needs to be able to store image for a later use.

Data in graphic files can be encoded in two different ways.

- **ASCII Text**

This is a readable text which is easy for humans to read and to some extent to edit and easy for programs to read and write. But it is bulky and slow to read and write from programs.

- **Compressed Format (Binary Formats)**

They are very compact but incomprehensible to human and require complex reading and writing routines. They vary a lot in terms of the flexibility they offer for the image size, shape, colours and their attributes. At one end is the TIFF (Tagged Input File Format) with so many different options and features that not TIFF

implementation can read them all and at other end is Mac Paint which allows storing the image in exactly one size, two colours and one way.

The graphic files are further classified as of two types in terms of the manner in which they store the image

- **Bitmapped Format**

Here the picture is represented as rectangular array of dots. It stores complete digitally encoded images. They are also called as raster or dot-matrix description. It is used when the images are, in large part, created by hand or scanned from an original document or photograph using some type of scanner.

A few types of bitmapped graphic files formats are:

TIFF (Tagged Input File Format)

GIF (Graphics Interchange Format)

BMP (Bit map Format)

Mac Paint

IMG

TGA (Targa)

JPEG (Joint Photographic Expert Group).

- **Vector Formats**

They represent a picture as a series of lines and arcs i.e. it stores the individual graphics that make up the image. These images are also called as line images. As most of the lines that are needed could be represented by relatively simple mathematical equations hence, images could be stored economically. For example, to specify a straight line all that is needed is a knowledge of the positions of the two end points of the line and for display purposes the line can then be reconstructed knowing the geometrical properties. Similarly to draw a circle all that is needed is knowledge of its centre and its radius. The advantages of vector formats are:

- They require less size.
- Their quality is not affected when the images are magnified as contrasting to the pixel images.

3.2.1.2 Pixel

A pixel (picture-element) is a dot or the most fundamental unit that makes up the image. All pixels have a value associated with them called as the pixel value representing the color for that point/pixel. For the simplest pictures, each point is

black or white so the pixel value is either 0 or 1, a single bit. However commonly, the picture is in grayscale or color, in which case there has to be a large range of pixel values. For a grayscale image, each pixel might be 8 bits, so the value could range from 0 for black to 255 for white.

3.2.1.3 True Color

24 bit color represents the limit of the human eye's ability to differentiate colors, Thus to human eye, there is no perceptible difference between a 24 bit color image of an object and the object viewed directly. Hence it is referred to as the true color.

3.2.1.4 Palette / Color Map

Full color images can be very large. A 600* 800 image may contain 4, 80,000 pixels. If each of the pixel we stored as 24-bit value than the image would consume 1.4 MB. To decrease the amount of space needed to store the image, the concept of color map or palette is used. Rather than storing the actual color of each pixel in the file, the color maps contains a list of all colors used in the image and the individual pixel values are stored as entry numbers in the color map/palette. A typical color map has 16 or 256 entries, so each pixel value is only 4 or 8 bits, an enormous savings from 24 bits per pixel. Programs can create various screen effects by changing the color map. The advantage of using the color map is that

- The amount of RAM and memory needed to store the image is considerably reduced.
- The image definition is virtualized. The value of the latter can be demonstrated by considering the task of changing one color in the image instead of changing all pixels of the color in image, we need to change only the palette entry for that color.

3.2.1.5 Color Model

A color model is a formal way for representing and defining colors. A synonymous term is photometric interpretation. There are different types of color models.

3.2.1.6 Resolution

Graphic images on the screen are made up of tiny dots called pixels or picture elements. The display resolution is defined by the no. of rows (called scan line) from top to bottom and no. of pixels from left to right on each scan line. Each mode uses a particular resolution, higher the resolution more pleasing is the picture. Higher

resolution means a sharper, clearer picture with less pronounced staircase effect on drawing lines diagonally and better looking text characters. High resolution requires more memory requirement to display the pictures.

3.2.1.7 Windows Bitmap Format (BMP)

The windows BMP format is a general purpose format for storing Device Independent Bitmaps (DIB's). By DIB we mean that the physical interpretation of the image and its palette are fixed without regard to the requirements of any potential display device. It is most often used to store screen and scanner generated imagery.

The BMP file only supports single line bitmaps of 1, 4, 8 or 24 bits per pixel. One annoying aspect of BMP is that image is stored by scan line proceeding from the bottom row to the top. All other formats use the reverse order or at least support top-to-bottom order as an option. Top to bottom is a defacto standard.

BMP breaks the file into four separate components

- File Header
- An image header.
- An array of palette entries.
- Actual bitmap

When dealing with BMP it is recommended to use a palette unless we are dealing with a 24-bit image. BMP supports image compression by RLE (run length encoding) only images with 4 bit and 8 bit per pixel sizes can be encoded. The interpretation of encoded image data slightly depends on which pixel size is present. Scanned file in the BMP format are padded with unused bits in the end so that their length is an integral number of double words i.e. the number of bytes is evenly divisible by 4. Despite the fact that the format supports compression, it's rare to find an application that actually bothers to encode image data in this format thus, only a few BMP files are compressed.

3.3 Development of a HDCR System

The problem defines in the acquisition process of an HDCR system can be justified by training of neural networks in reconstruction of Hindi characters. First of all, system is taught by offline handwritten different shapes of Hindi characters. On the basis of this image model database, patterns are matched and classify the reconstructed image. The HDCR system is developing as follows:

3.3.1 Offline Handwritten Image Samples

These are the original image drawn by user by free handwriting that stores in a file databases. This file database makes an image model library in which we are storing different types of binary images drawn by different users using different styles of handwriting. The following are the image samples of Hindi characters:-



Figure 28: Samples of Offline Handwritten Devanagari Character Images क, ख, ग.

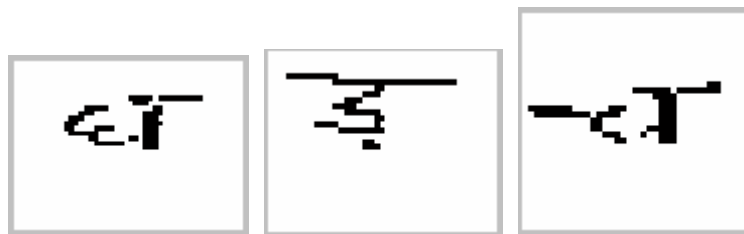


Figure 29: Samples of Offline Handwritten Devanagari Character Images घ, ङ, च.



Figure 30: Samples of Offline Handwritten Devanagari Character Images छ, ज, झ.



Figure 31: Samples of Offline Handwritten Devanagari Character Images ट, ठ, ड.



Figure 32: Samples of Offline Handwritten Devanagari Character Images ड, ढ, ण.



Figure 33: Samples of Offline Handwritten Devanagari Character Images त, थ, द.



Figure 34: Samples of Offline Handwritten Devanagari Character Images ध, न, प.



Figure 35: Samples of Offline Handwritten Devanagari Character Images फ, ब, म.

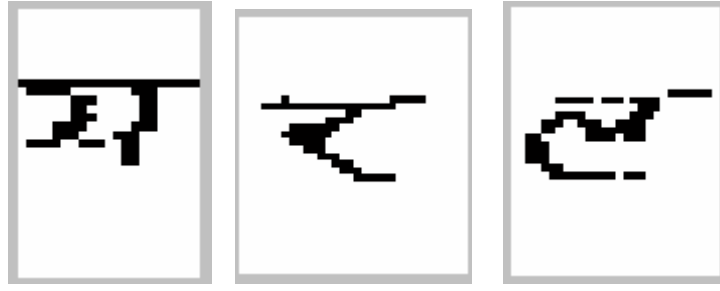


Figure 36: Samples of Offline Handwritten Devanagari Character Images य, र, ल.



Figure 37: Samples of Offline Handwritten Devanagari Character Images श, ष, स.



Figure 38: Samples of Offline Handwritten Devanagari Character Images ह, क्ष, त्र, ज.

4.1 Neural Network

Neural networks are composed of simple elements operating in parallel. These elements are inspired by biological nervous systems. As in nature, the network function is determined largely by the connections between elements. We can train a neural network to perform a particular function by adjusting the values of the connections (weights) between elements. Commonly neural networks are adjusted, or trained, so that a particular input leads to a specific target output. Such a situation is shown in fig 29. There, the network is adjusted, based on a comparison of the output and the target, until the network output matches the target. Typically many such input/target pairs are used, in this supervised learning, to train a network.

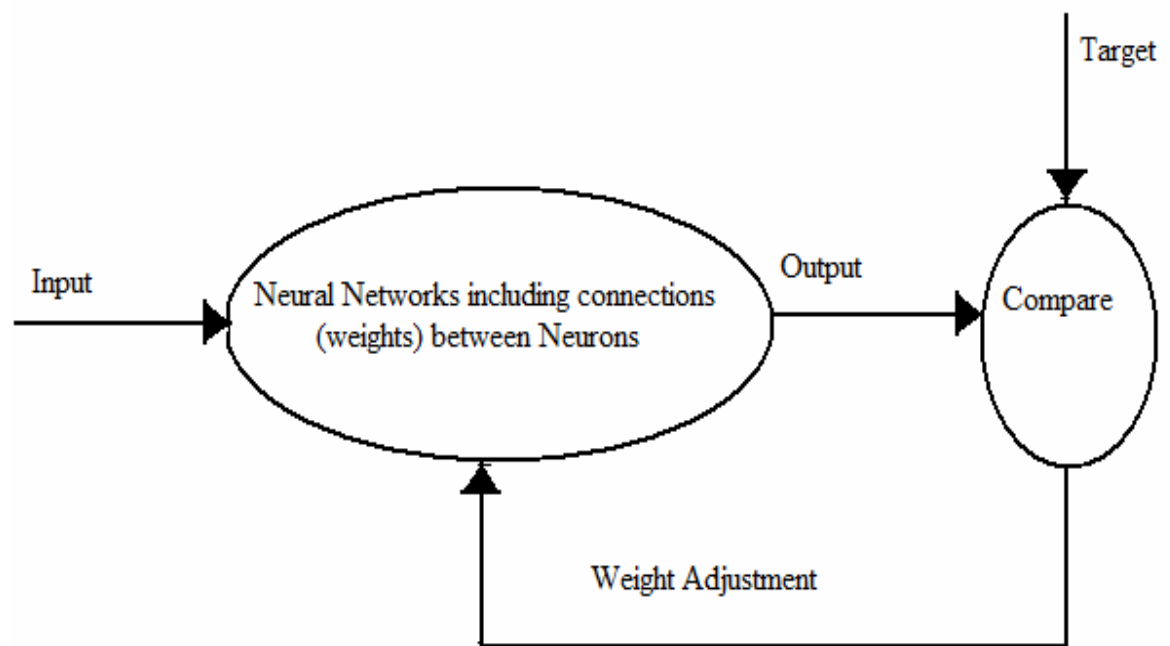


Figure 39: Block Diagram of Neural Network

Neural networks have been trained to perform complex functions in various fields of application including pattern recognition, identification, classification, speech, vision and control systems.

Today neural networks can be trained to solve problems that are difficult for conventional computers or human beings. The supervised training methods are commonly used, but other networks can be obtained from unsupervised training techniques or from direct design methods. Unsupervised networks can be used, for instance, to identify groups of data. Certain kinds of linear networks and Hopfield networks are designed directly. In summary, there are a variety of kinds of design and learning techniques that enrich the choices that a user can make.

The field of neural networks has a history of some five decades but it has been found that solid application only in the past fifteen years and the field is still developing rapidly. Thus, it is distinctly different from the fields of control systems or optimization where the terminology, basic mathematics, and design procedures have been firmly established and applied for many years.

4.1.1 Neuron Model

4.1.1.1 The Biological Neuron

Neural networks resemble the human brain in the following two ways:

1. A neural network acquires knowledge through learning
2. A neural network's knowledge is stored within inter-neuron connection strengths known as synaptic weights.

The most basic element of the human brain is a specific type of cell, which provides us with the abilities to remember, think, and apply previous experiences to our every action. These cells are known as neurons, each of these neurons can connect with up to 200000 other neurons. The power of the brain comes from the number of these basic components and the multiple connections between them. All natural neurons have four basic components, which are dendrites, soma, axon, and synapses. Basically, a biological neuron receives inputs from other sources, combines them in some way, performs a generally nonlinear operation on the result, and then output the final result. The figure below shows a simplified biological neuron and the relationship of its four components.

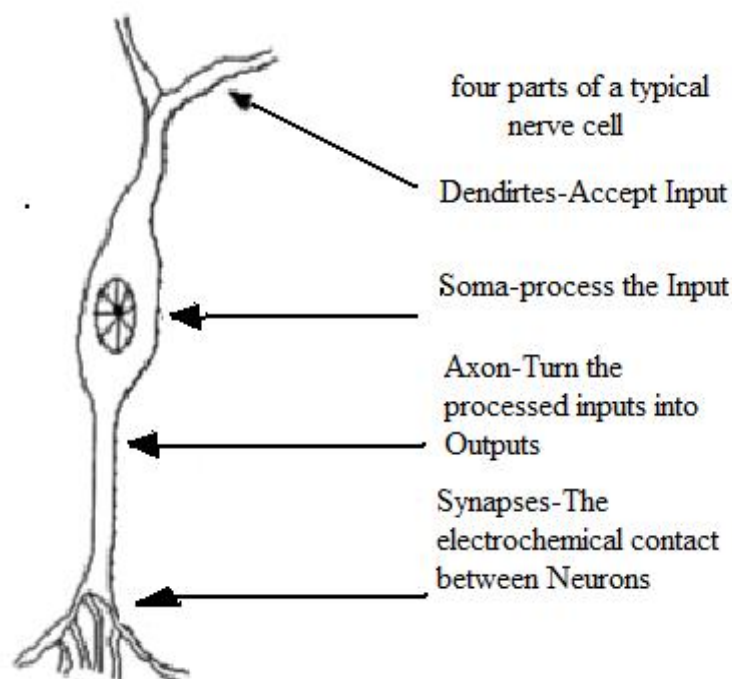


Figure 40: A biological Neuron

In the human brain, a typical neuron collects signals from others through a host of fine structures called dendrites. The neuron sends out spikes of electrical activity through a long, thin strand known as an axon, which splits into thousands of branches. At the end of each branch, a structure called a synapse converts the activity from the axon into electrical effects that inhibit or excite activity from the axon into electrical effects that inhibit and excite activity in the connected neurones. When a neuron receives excitatory input that is sufficiently large compared with its inhibitory input, it sends a spike of electrical activity down its axon. Learning occurs by changing the effectiveness of the synapses so that the influence of the neuron on another changes [36].

4.1.1.2 Simple Neuron

The basic unit of neural networks, the artificial neurons, simulates the four basic functions of natural neurons. Artificial neurons are much simpler than the biological neuron.

A neuron with a single scalar input and no bias is shown below.

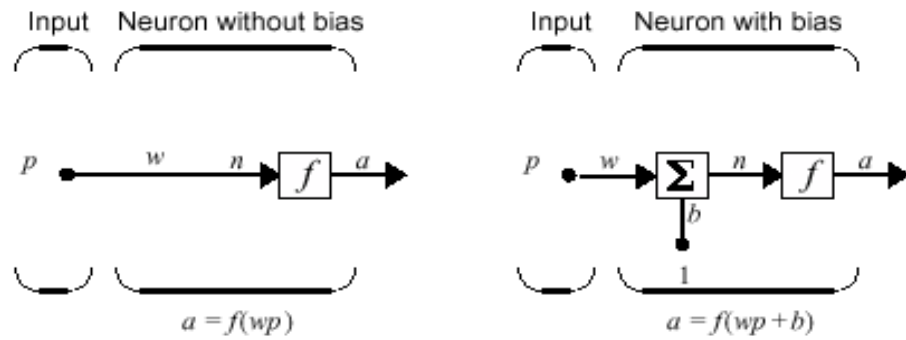


Figure 41: Neuron

The scalar input p is transmitted through a connection that multiplies its strength by the scalar weight w , to form the product wp , again a scalar. Here the weighted input wp is the only argument of the transfer function f , which produces the scalar output a . The neuron on the right has a scalar bias, b . Bias as simply being added to the product wp as shown by the summing junction or as shifting the function f to the left by an amount b . The bias is much like a weight, except that it has a constant input of 1. The transfer function net input n , again a scalar, is the sum of the weighted input wp and the bias b . This sum is the argument of the transfer function f . Here f is a transfer function, typically a step function or a sigmoid function, that takes the argument n and produces the output a . Note that w and b are both adjustable scalar parameters of the neuron. The central idea of neural networks is that such parameters can be adjusted so that the network exhibits some desired or interesting behaviour.

4.2 Neural Network Architecture

Basically the neural network consists of:

- Neurons
- Interconnections among neurons (weights).

Neurons: Similar to the human brain neurons, the neurons in the neural network transports the incoming information on their outgoing connections to the other neurons.

Weights: In neural net the outgoing connections from a neuron to the other neurons are called weights. The following information is simulated with specific values stored in those weights.

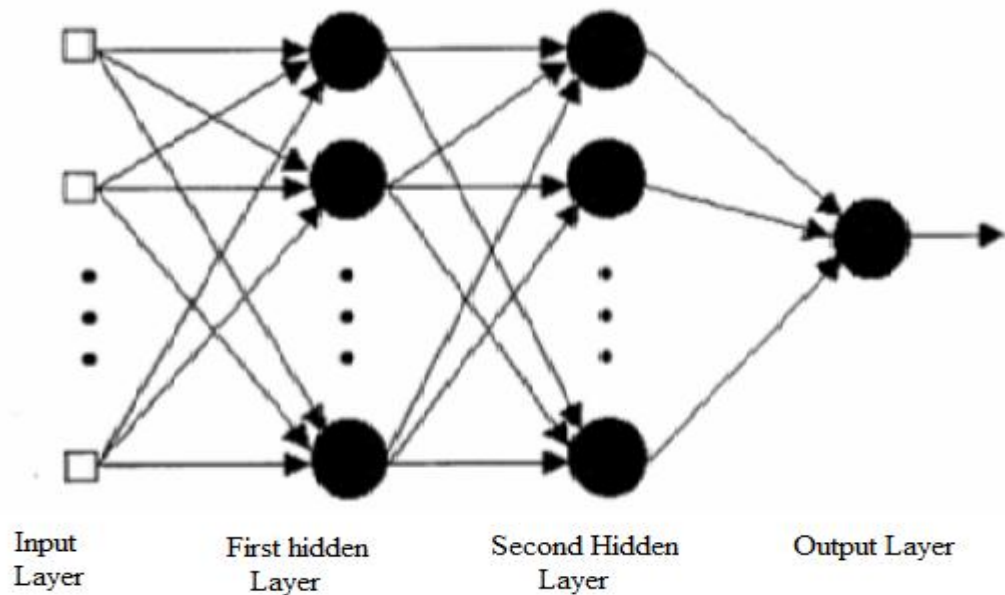


Figure 42: Architecture of Neural network showing different Layers

4.3 Working of Neural Networks

Working of a neural network can be thought of as consisting of two phases:

- Testing
- Learning

Neural network maps a set of inputs to a set of output values. This non-linear mapping can be thought of as a multidimensional mapping surface. Therefore “the object of learning is to model the mapping surface according to a desired response, either with or without an explicit training process”

The brain basically learns from experience. Neural networks are sometimes called machine learning algorithms, because changing of its connection weights (training) causes the network to learn the solution to a problem [35]. The strength of connection between the neurons is stored as a weight-value for the specific connection. The system learns new knowledge by adjusting these connection weights. The learning ability of a neural network is determined by its architecture and by the algorithmic methods chosen for training.

4.3.1 Modes of Learning

The training method usually consists of this schema. There are two modes of learning **Supervised** and **Unsupervised**. Below there is a brief description of each one to determine the best one for our problem

a) Supervised Learning

Supervised learning is based on the system trying to predict outcomes for known examples and is a commonly used training method. It compares its predictions to the target answer and "learns" from its mistakes. The data start as inputs to the input layer neurons. The neurons pass the inputs along to the next nodes. As inputs are passed along, the weighting, or connection, is applied and when the inputs reach the next node, the weightings are summed and either intensified or weakened. This continues until the data reach the output layer where the model predicts an outcome. In a supervised learning system, the predicted output is compared to the actual output for that case. If the predicted output is equal to the actual output, no change is made to the weights in the system. But, if the predicted output is higher or lower than the actual outcome in the data, the error is propagated back through the system and the weights are adjusted accordingly.

This feeding error backwards through the network is called "back-propagation." Both the Multi-Layer Perceptron and the Radial Basis Function are supervised learning techniques. The Multi-Layer Perceptron uses the back-propagation while the Radial Basis Function is a feed-forward approach which trains on a single pass of the data.

b) Unsupervised Learning

Neural networks which use unsupervised learning are most effective for describing data rather than predicting it. The neural network is not shown any outputs or answers as part of the training process-in fact, there is no concept of output fields in this type of system. The primary unsupervised technique is the Kohonen network. The main uses of Kohonen and other unsupervised neural systems are in cluster analysis where the goal is to group "like" cases together. The advantage of the neural network for this type of analysis is that it requires no initial assumptions about what constitutes a group or how many groups there are. The system starts with a clean slate and is not biased about which factors should be most important.

c) Reinforcement Learning

This type of learning may be considered as an intermediate form of the above two types of learning. Here the learning machine does some action on the environment and gets a feedback response from the environment. The learning system grades its action good (rewarding) or bad (punishable) based on the environmental response and accordingly adjusts its parameters. Generally, parameter adjustment is continued until an equilibrium state occurs, following which there will be no more changes in its parameters. The self organizing neural learning may be categorized under this type of learning.

c) Back Propagation

This method is proven highly successful in training of multilayered neural nets. The network is not just given reinforcement for how it is done on a task. Information about errors is also filtered back through the system and is used to adjust the connections between the layers, thus improving performance.

4.3.2 Testing

After training is complete, a test pattern is given to the neural network and the results are compared with the desired result. Difference between the two values gives the error. Percentage accuracy is found as follows:

$$\% \text{ Accuracy} = \frac{\text{No of characters found correctly}}{\text{Total no of patterns.}} * 100$$

4.4. Neural Network Topologies

4.4.1 Feed-forward Neural Networks

Feed-forward neural networks, where the data taken from input to output units is strictly feed forward. The data processing can extend over multiple (layers of) units, but no feedback connections are present, that is, connections extending from outputs of units to inputs of units in the same layer or previous layers.

Classical examples of feed-forward neural networks are: the Perceptron and Adaline.

4.4.2 Recurrent Neural Networks

Recurrent neural networks contain feedback connections. Contrary to feed-forward networks, the dynamical properties of the network are important. In some cases, the

activation values of the units undergo a relaxation process such that the neural network will evolve to stable state in which these activations do not change anymore. In other applications, the changes of the activation values of the output neurons are significant, such that the dynamical behaviour constitutes the output of the neural network.

4.5 Advantages of Neural Computing

There are a variety of benefits that an analyst realizes from using neural networks in their work.

- The system is developed through learning rather than programming. Programming is much more time consuming for the analyst and requires the Analyst to specify the exact behaviour of the model. Neural nets teach themselves the patterns in the data freeing the analyst for more interesting work.
- Pattern recognition is a powerful technique for harnessing the information in the data and generalizing about it. Neural nets learn to recognize the patterns which exist in the data set.
- Neural networks are flexible in a changing environment. Rule based systems or programmed systems are limited to the situation for which they were designed--when conditions change, they are no longer valid. Although neural networks may take some time to learn a sudden drastic change, they are excellent at adapting to constantly changing information.
- Neural networks are flexible in a changing environment. Rule based systems or programmed systems are limited to the situation for which they were designed--when conditions change, they are no longer valid. Although neural networks may take some time to learn a sudden drastic change, they are excellent at adapting to constantly changing information.

4.6 Limitations of Neural Computing

There are some limitations to neural computing. The key limitation is the neural network's inability to explain the model it has built in a useful way. Analysts often want to know why the model is behaving as it is. Neural networks get better answers but they have a hard time explaining how they got there. There are a few other limitations that should be understood. First, it is difficult to extract rules from neural

networks. This is sometimes important to people who have to explain their answer to others and to people who have been involved with artificial intelligence, particularly expert systems which are rule-based.

4.7 Application of Neural Network

Neural network have been successful applied to broad spectrum of data-intensive application, such as:

- **Data association** – Like classification but it also recognizes data that contains errors. For example its not only identifies the characters that are scanned but identify when the scanner is not working properly.
- **Data Conceptualization** – Analyze the inputs so that grouping relationships can be inferred. e.g. extract from a database the names of those most likely to by a particular product.
- **Data Filtering** – Smooth an input signal. e.g. it takes the noise out of a telephone signal.
- **Process Modelling and Control** - Creating a neural network model for a physical plant then using that model to determine the best control settings for the plant.
- **Machine Diagnostics** - Detect when a machine has failed so that the system can automatically shut down the machine when this occurs.
- **Portfolio Management** - Allocate the assists in a portfolio in a way that maximizes return and minimizes risk.
- **Target Recognition** – Military application which uses video and infrared image data to determine if any enemy target is present.
- **Medical Diagnosis** – Assisting doctors with their diagnosis by analyzing the reported symptoms and image data such as MRI or X-rays.
- **Credit Rating** – Automatically assigning a company's or individual's credit rating based on their financial condition.
- **Targeted Market** – Finding the sets of demographics which have the highest response rate for a particular marketing campaign.
- **Voice Recognition** – Transcribing spoken words into ASCII text.
- **Financial Forecasting** – Using the historical data of a security to predict the future movement of that security.

- **Quality Control** – Attaching a camera or sensor to the end of a production process to automatically inspect for defects.
- **Intelligent Searching** – An internet search engine that provides the most relevant content and banner ads based on the users past behaviour.
- **Fraud Detection** – Detect fraudulent credit card transaction and automatically decline the charge.
- **Classification** – Uses input values to determine the classification. e.g. is the input the letter A, is the blob of the video data a plane and what kind of plane is it.
- **Prediction** – Uses input values to predict some output, e.g. pick the best stocks in the market, predict weather, and identify people with cancer risk.

4.8 MATLAB

The name MATLAB stands for matrix laboratory. MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include:

- Math and computation
- Algorithm development
- Modelling, simulation, and prototyping
- Data analysis, exploration, and visualization
- Scientific and engineering graphics
- Application development, including Graphical User Interface building

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows, solving many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar non-interactive language such as C or FORTRAN. MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, MATLAB is the tool of choice for high-productivity research, development, and analysis. The reason that I have decided to use MATLAB for the development of this project is its toolboxes. Toolboxes allow you to learn and apply specialized technology. Toolboxes are comprehensive.

4.9 Recognition of Handwritten Hindi Characters

As discussed previously, the problem of recognition of characters can be solved using neural networks. A scheme is proposed to recognize characters from a given input image. Recognition of Handwritten Hindi characters is performed through following steps by using Neural Network:-

4.9.1 Load Image

Firstly, we load Scanned image as an input, and then by using the input image we recognize characters. Figure 33 and figure 34 represents the step of loading of image.

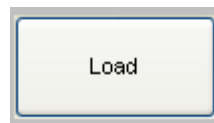


Figure 43: Load Button



Figure 44: Character Image

When load button (shown in figure 43) is pressed, a window opens. This window is used to specify the path where the character image is located. After this process, the image is shown (figure 44). After loading of image, selection of character is performed.

4.9.2 Selection of Character

After inputting the character image, select a particular character for the recognition. When we select a particular character from character image, that character is displayed in another window. A separate window is also shown in which bounding box of all characters is created. The advantage of creating bounding box is that area of a particular character can be calculated easily. In this, there is no limitation of number of characters. Any number of characters can be boxed which are included in the given character image. An example of character २ is shown below, i.e. selection of character २. The following figures represent the selection and bounded boxes of character:-

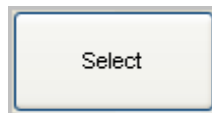


Figure 45: Select Button

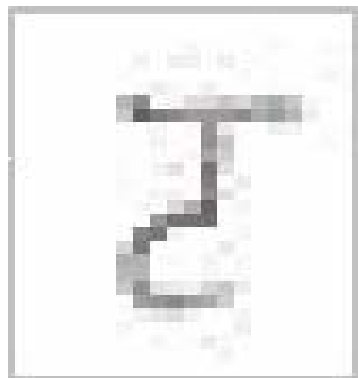


Figure 46: Selected Character from Image

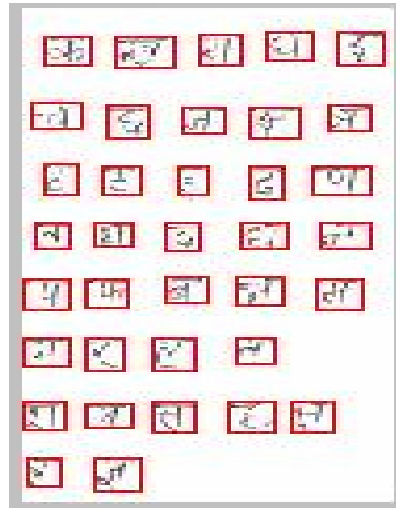


Figure 47: Bounding Box of Characters

When select button (shown in figure 45) is pressed, the selected character is shown in window. This character is processed for further recognition. Another window is also displayed the bounding box of all characters. After selection of image, next step (preprocessing) is performed of the selected image.

4.9.3 Preprocessing

After selection of a particular character, that character is pre-processed. It deals with technique for enhancing contrast; removing noise and isolating regions whose texture indicate a likelihood of character information. In preprocessing stage it is being normalized and removing all redundancy errors from the image and sends it to next stage.

The following are main preprocessing steps:-

- a) Firstly, that character is cropped i.e. extra pixels are removed from the character image.
- b) Then, that RGB image is converted into Gray scale image.
- c) After that, edges are finding out of that character.
- d) Extra holes fill up from that character.
- e) Bounding Boxes are made up of all characters. These boxes represent the area of whole character.

The following example represents the preprocessing steps of a character:-



Figure 48: Cropped Character ‘ट’

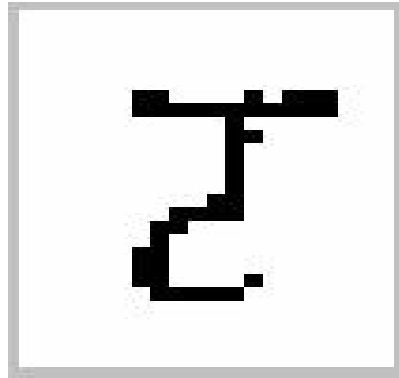


Figure 49: Filled character ‘ट’

4.9.4 Feature Extraction

After preprocessing of character, features of character are extracted. This step is heart of the system. This step helps in classifying the characters based on their features. In fact, the main problem in HDCR system is the large variation in shapes within a class of character. Handwritten Hindi character are imprecise in nature as their corners are not always sharp, lines are not perfectly straight, and curves are not necessarily smooth, unlikely the printed character. This variation depends on font styles, document noise, photometric effect, document skew and poor image quality. The large variation in shapes makes it difficult to determine the number of features that are convenient prior to model building. Though many kinds of features have been developed and their test performances on standard database have been reported. The following figure represents feature extraction of character ‘ट’:-

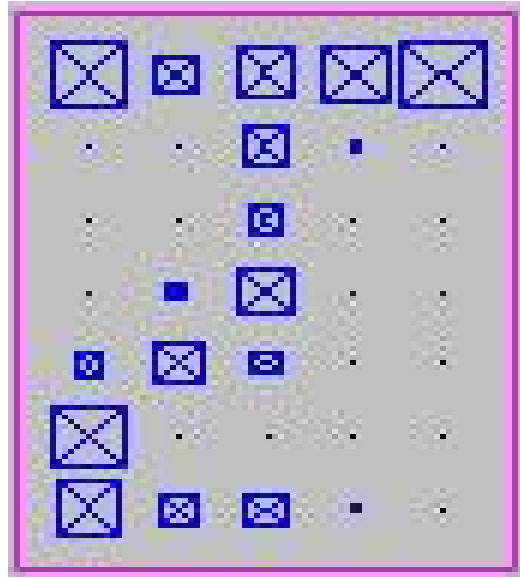


Figure 50: Extracted features of character '२'

Structural features should be chosen keeping in mind that the shape variations should affect feature set minimally. It was not an easy task to decide which structural features should be chosen to extract the structural features from characters of devnagri script due to large shape variations in characters of the same class.

4.9.5 Neural Network Training

The program trains the network to recognize the characters. This network takes input-output vector pairs during training. The network trains its weight array to minimize the selected performance measure, i.e., error using back propagation algorithm.

The following are taken as inputs from the user:

- a) The input pattern file
- b) No. of neurons in each hidden layer
- c) Value of learning rate
- d) Value of momentum constant
- e) Error value for convergence

The output of training program is a file which contains modified weights of different connection of the network. This file is used as the input to testing program. This file also contains the values of numbers of neurons in input layer, Hidden layers, output layer, value of learning rate and momentum factor so that used is no require to enter the values during testing.

Following parameters are used for training of Neural Networks:

No. of neurons in Input Layer: 7

No of neurons in Hidden Layer: 10

No of epochs: 800

Transfer Function Used for Layer 1: “Logsig”

Transfer Function Used for Layer 2: “Tansig”

Adaption Learning Function: “Learngdm”

Performance Function: “MSE”

4.9.6 Testing

After training is complete, a test pattern is given to the neural network and the results are compared with the desired result. Difference between the two values gives the error. Percentage accuracy is found as follows:

$$\% \text{ Accuracy} = \frac{\text{No of characters found correctly}}{\text{Total no of patterns.}} * 100$$

Handwritten Devanagari Character sets are taken. These steps are followed to obtain best accuracy of input handwritten Hindi character image from the HDCR system. First of all, training of system is done by using different data set or sample. And then system is tested for few of the given sample, and accuracy is measured. The data set was partitioned into two parts. The first part is used for training the system and the second was for testing purpose. For each character, feature were computed and stored for training the network. The table given below display the results obtained from the program. The variance is very small but it is there. Following are main results of Hindi character recognition: -

Table 2: Recognition Accuracy of Handwritten Hindi Characters

Character	Confusing Recognized Characters	# Samples for training	# Samples for testing	% Recognition Accuracy
क		180	20	90%
ख		130	30	85%
ग	म	85	20	75%
घ		170	25	71%
ङ		110	15	69%
च		140	20	81%
छ		75	15	79%
ज		90	20	84%
झ		170	20	80%
ञ		100	30	90%
ट	ढ ठ	166	30	70%
ठ		165	25	86%
ड		170	20	89%
ढ	ट	167	20	70%
ण		149	20	95%
त		174	30	80%
थ	भ	168	30	75%
द	ढ	146	20	72%
ध	घ	176	20	69%
न		164	25	88%

प	य	148	20	75%
फ		158	20	84%
ब	व	176	20	65%
भ	म	151	20	70%
म		162	20	88%
य	प	164	20	67%
र		149	20	89%
ल		155	25	76%
व		150	20	78%
श		173	20	71%
ष		165	20	82%
स		146	30	70%
ह		147	20	92%
क्ष		132	25	85%
त्र		158	20	87%
ज्ञ		153	30	89%

Three network layer one input layer, one hidden layer and one output layer are taken. If number of neurons in the hidden layer is increased, then a problem of allocation of required memory is occurred. Also, if the value of error tolerance is high, desired results are not obtained, so changing the value of error tolerance i.e. say, high accuracy rate is obtained. Also the network takes more number of cycles to learn when the error tolerance value is less rather than in the case of high value of error tolerance in which network learns in less number of cycles and so the learning is not very fine.

6.1 Conclusion

Offline handwritten Hindi character recognition is a difficult problem, not only because of the great amount of variations in human handwriting, but also, because of the overlapped and joined characters. Recognition approaches heavily depend on the nature of the data to be recognized. Since handwritten Hindi characters could be of various shapes and size, the recognition process needs to be much efficient and accurate to recognize the characters written by different users. As neural network is used here for recognition of offline Hindi character. There are few reasons that create problem in Hindi handwritten character recognition.

- Some characters are similar in shape (for example  and ).
- Sometimes characters are overlapped and joined.
- Large numbers of character and stroke classes are present there.
- Different, or even the same user can write differently at different times, depending on the pen or pencil, the width of the line, the slight rotation of the paper, the type of paper and the mood and stress level of the person.
- The character can be written at different location on paper or in window
- Characters can be written in different fonts.

These reasons are considered over here. A small set of Hindi characters using back propagation neural network is trained, then testing was performed on other character set. The accuracy of network was very low. Then, some other character images in the old character set are added and trained the network using new sets. Then again testing was performed on some new image sets written by different people, and it was found that accuracy of the network increases slightly in some cases. Again some new character images into old character set are added (on which network was trained) and trained the network using this new set. The network is presented new character

images and it has been seen that recognition increases, although at a slow rate. It can be concluded that as the network is trained with more number of sets, the accuracy of recognition of characters will increase definitely.

6.2 Future scope

- This can be implemented for recognition of online Hindi characters.
- It can be extended for the recognition of words, sentence and documents.
- Another research interest will be on the character images degraded or blurred by various reasons.
- This approach can be used in multilingual character recognition as well.

REFERENCES

- [1] R. Plamondon and S. N. Srihari, "On-line and off-line handwritten recognition: a comprehensive survey", IEEE Transactions on PAMI, Vol. 22(1), pp. 63–84, 2000.
- [2] Huanfeng Ma David Doermann, "Adaptive Hindi OCR Using generalized Hausdorff Image comparison".
- [3] R.M.K.Sinha, "Rule based contextual post-processing for Devanagari text Recognition", Pattern Recognition", 20(5), pp. 475-485, 1987.
- [4] I.K. Sethi and B. Chatterjee, "Machine recognition of handprinted Devanagari Numerals, Journal of Institution of Electronics and Telecommunication Engineers", India, vol. 22, pp 532-535, 1976.
- [5] ERIC W. BROWN, "Character Recognition by Feature Point Extraction".
- [6] Manish,"Degraded Text Recognition of Gurmukhi Script".
- [7] R Sanjeev Kunte and R D Sudhaker Samuel, "A simple and efficient optical character recognition system".
- [8] Veena Bansal and R.M.K. Sinha, "A Complete OCR for printed Hindi text Devnagiri script".
- [9] Veena Bansal an R.M.K. Sinha, "Segmentation of touching and fused devanagari characters", Technical Report, TRCS-97-247, I.I.T. Kanpur, India, 1997.
- [10] Veena Bansal and R.M.K. Sinha, "Partitioning and Searching Dictionary for Correction of Optically-Read Devanagari Character Strings", Technical Report, TRCS-97-246, I.I.T. Kanpur, India, (1997).
- [11] Avinash Tiwari, Veena Bansal , "Content Based Image Retrieval System For Patent Database" Industrial and Management Engineering Department", IIT Kanpur India.
- [12] Aditya Gokhale,"BI-Lingual Optical Character Recognition System for Devanagari and English", Centre For Development Of Advanced Computing, GIST R&D Pune, Maharashtra, INDIA.
- [13] M. Blumenstein and B. Verma , "A New Segmentation Algorithm for Handwritten Word Recognition", IJCNN '99, Washington, U.S.A., 1999.
- [14] H. Swethalakshmi1, Anitha Jayaraman1, V. Srinivasa Chakravarthy2, C. Chandra Sekhar, "Online Handwritten Character Recognition of Devanagari and

Telugu Characters using Support Vector Machines”, Department of Computer Science and Engineering, 2Department of Biotechnology, Indian Institute of Technology Madras, India.

- [15] Sandhya Arora , Debotosh Bhattacharjee , Mita Nasipuri , ”A Two Stage Classification Approach for Handwritten Devanagari characters”, International Conference on Computational Intelligence and Multimedia Applications 2007.
- [16] M. Blumenstein, and B. Verma, “An Artificial Neural Network Based Segmentation Algorithm for Off-line Handwriting Recognition”, International Conference on Computational Intelligence and Multimedia Applications fICCAL4 '98), Melbourne, Australia, 1997.
- [17] K. Y. Rajput and Sangeeta Mishra , ”Recognition and Editing of Devnagari Handwriting Using Neural Network”, Proceedings of SPIT-IEEE Colloquium and International Conference, Mumbai, India Vol. 1, 66.
- [18] Vivek Pandey and Veena Bansal, “A Decision-Making Framework for IT Outsourcing using the Analytic Hierarchy Process”.
- [19] Tripti Negi and Veena Bansal, “Time Series:Similarity Search and its Applications”.
- [20] Veena Bansal and R.M.K. Sinha, “Integrating Knowledge Sources in Devanagari Text Recognition Recognition System”, Technical Report, TRCS-97-248, I.I.T. Kanpur, India, 1997.
- [21] Cheng-Lin Liu , ”Online Recognition of Chinese Characters: The State-of-the-Art”, IEEE Transactions on Pattern Analysis and Machine Intelligence” , VOL. 26, NO. 2, February 2004.
- [22] Veena Bansal R. M. K. Sinha ,” On Integrating Diverse Knowledge Sources in Optical Reading of Devnagari Script “.
- [23] Veena Bansal R. M. K. Sinha, “Designing a Front End OCR System for Indian Scripts for Machine Translation - A Case Study for Devnagari”.
- [24] M. Blumenstein and B. Verma ,” A Neural Based Segmentation and Recognition Technique for Handwritten Words”.
- [25] S. Connell and A.K. Jain, “Writer Adaptation of Online Handwritten Models,” Proc. 5th Int. Conf. Document Analysis and Recognition, Bangalore, India, pp. 434-437, Sept. 1999.

- [26] V. Bansal and R.M.K. Sinha , “On how to describe shapes of Devanagari characters and use them for recognition”, Proc. 5th Int. Conf. Document Analysis and Recognition, Bangalore, India, pp. 410-413, Sept. 1999.
- [27] Harikesh Singh, Dr. R. K. Sharma,” Moment in Online Handwritten Character Recognition”, Thapar Institute of Engineering and Technology, Patiala (Punjab)-INDIA.
- [28] Shuichi Tsujimoto and Haruo Asada, “Solving Ambiguities in Segmenting Touching Characters, Structured Document Image Analysis,” H. S. Baird et al., eds., Springer-Verlag, U.S.A., 1992.
- [29] Vamsi K. Madasu, Brian C. Lovell, M. Hanmandlu ,”Hand printed Character Recognition using Neural Networks “, School of ITEE, University of Queensland, Australia.
- [30] Deepayan Sarkar , “Optical Character Recognition using Neural Networks” ,(ECE 539 Project Report) Department of Statistics University of Wisconsin.
- [31] Alessandro Vinciarelli , “ A surveyon off-line Cursive Word Recognition “, Institute Dalle Molle d’Intelligence Arti_cielle Perceptive, CP592-Rue du Simplon 4, 1920 Martigny, Switzerland , accepted 14 June 2001.
- [32] Subramanian S. Meiyappan, Shobana Sridharan and Esther T. Ososanya , “Multi-Lingual Character Recognition Using Artificial Neural Networks”, Electrical and Computer Engineering, Tennessee Technological University,Cookeville, Tennessee.
- [33] C. J. C. Burges, “A tutorial on support vector machines for pattern recognition, Data Mining and Knowledge Discovery “, 1998, pp 121-167.
- [34] S. Hewavitharana, H. C. Fernando ,” A Two Stage Classification Approach to Tamil Handwriting Recognition “, Department of Computer Science University of Colombo, Sri Lanka.
- [35] Anil k. Jain, Jianchang Mao, K.M. Mohiuddin, “Artificial Neural Network: A Tutorial”, IEEE 0018-9162/96, March 1996.
- [36] Denial Klerfors, Dr Terry L.haston, “Artificial Neural Network”:St. Louis November, 1998.
- [37] Reena Bjaj, Lipika Dey and Shantanu Chaudhary, “Devanagari numeral recognition by combining decision of multiple connectionist classifiers”, Sadhna Vol.27, Part 1, February 2002, pp.59-72.

- [38] Plamondon, R. and S. N. Srihari (2000), "On-Line and Off- Line Handwriting Recognition: A Comprehensive Survey", IEEE trans. Pattern Analysis and Machine Intelligence, vol. 22 (1), pp. 63-84.
- [39] Y. Lu and M. Shridhar, "Character Segmentation in Handwritten Words – An Overview", Pattern Recognition", Vol. 29, 1996, pp. 77-96.
- [40] R.M.K. Sinha and V. Bansal, "On Devanagari document processing," Proc. Int. Conf. on Systems, Man and Cybernetics" , Vancouver, BC, pp. 1621-1626, Oct. 1995.
- [41] R.M.K.Sinha, "Rule based contextual post-processing for Devanagari text recognition" , Pattern Recognition, 20(5), pp. 475-485, 1987.
- [42] R. G. Casey and E. Lecolinet, "A survey of Methods and Strategies in Character Segmentation ", IEEE Transactions on Pattern Analysis and Machine Intelligence, 18, pp.690-706, 1996.
- [43] S. N. Srihari, "Recognition of Handwritten and Machineprinted Text for Postal Address Interpretation", Pattern Recognition Letters, 14, 1993, pp. 291-302.
- [44] M. Blumenstein and B. Verma , "A Segmentation Algorithm used in Conjunction with Artificial Neural Networks for the Recognition of Real-World Postal Addresses" , ICCIMA'97, Australia, 155-160, 1997.

PAPERS PUBLISHED

- “Recognition of Handwritten Indian Scripts: An Analysis” in National Conference on “EMERGING TRENDS IN SOFTWARE AND NETWORKING TECHNOLOGIES” on April 17-18 held in Amity University, Noida.

- “A survey on Recognition of Handwritten Indian Scripts” in National Conference on “MODERN MANAGEMENT PRACTICES & INFORMATION TECHNOLOGY TRENDS” on April 15-16 held in DAV institute of engg. & tech. Jalandhar.