Handwritten Gurumukhi Character Recognition Using Neural Networks

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

Master of Engineering
in
Software Engineering

By:
Naveen Garg
(80731015)

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, **"Handwritten Gurumukhi Character recognition Using Neural Networks"**, 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.

(Naveen Garg)

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 austerous 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 Affair 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 whose support and care makes me stay on earth.

Place: TU, Patiala

Date: **(NAVEEN GARG)**

ABSTRACT

Today, Handwritten Character recognition is one of the challenging computational processes. Some computational fields like artificial intelligence, expert systems have provided an important role in recognition of these handwritten Gurumukhi characters. There is competition between the speed and efficiency. The human mind can easily decipher these handwritten characters easily, accurately and speedily. The human mind can do it because of the presence of densely neural network in his mind.

In 1948, Wiener proposed an idea of using the non-biological neural network to solve human processes. Using that non-biological neural network, many processes can be done easily and efficiently as human mind can do where the system can take the decisions to solve the problems. In pattern recognition, artificial neural network take decision to recognize the characters. In pattern recognition, first of all the Punjabi characters are digitized using any VLSI technology. After that these characters are pre-processed for removing noisy data.

After preprocessing, feature extraction is performed in which mainly neural network is needed. Feature extraction is used to differentiate between the characters. The character is written on page and scanned. These characters are stored as an image of 32 * 32 matrix size. The image can be of any type i.e. .jpg, .bmp, .gif. The end user can test the additional characters if required.

Contents

List of Figures

List of Tables

Chapter 1 Introduction

Today, many researches have been done to recognize Punjabi characters. But the problem of interchanging data between human beings and computing machines is a challenging one. Even today, many algorithms have been proposed by many researchers so-that these Gurumukhi characters can be easily recognize. But the efficiency of these algorithms is not satisfactory.

Mainly, users do Handwritten Character Recognition for interpretation of data, which describes handwritten drawing. Handwritten character recognition can be differentiated into two categories i.e. Online Handwritten character recognition and Offline Handwritten character recognition. On-line handwritten character recognition deals with automatic conversion of characters, which are written on a special digitizer, tablet PC or PDA where a sensor picks up the pen-tip movements as well as pen-up/pen-down switching. Off-line handwritten character recognition deals with a data set, which is obtained from a scanned handwritten document. The main objective of handwritten Gurumukhi character recognition (HGCR) is to recognize the Gurumukhi characters in desirable format from image format so that they can be easily edited.

Many researches have been done to solve handwritten character recognition problem in related areas such as Image Processing, Pattern Recognition, Artificial Intelligence, and cognitive science etc. Further researches are being done to improve accuracy and efficiency. Recognition of Offline Handwritten Gurumukhi characters is a goal of many research efforts in the pattern recognition field. Many techniques have been applied for recognition of handwritten Gurumukhi characters but still it is the case of less efficiency and accuracy of recognition.

Artificial Intelligence concepts like neural networks are used to perform the work as human mind can do. This explores the idea of how humans recognize text in general and are used to develop machines that simulated this process. Developing these intelligent machines for recognizing characters is not an easy task; this is because a character can be written in different ways. Also there are so many imperfections and variation of handwriting such as alignment, noise and angles, which make handwritten character recognition difficult to implement with a machine. All these imperfections of

handwritten characters can not be removed easily, so this problem is separated in sub-problems. This means that a single process or single machine is not capable of performing the entire process. It can be done by a series of processes that return some desirable result. Then a final process will combine the results from all the sub-process to produce a final result. For recognition of handwritten characters, some sub-processes are followed i.e. scanning a document, pre-processing, feature extraction etc. The output of previous process is used as input in next process.

For example, the first sub-process i.e. scanning of document scans the information and produces some output that is taken by next sub-process i.e. preprocessing which eliminates all the noise such as ink spots that have no meaning. Then next sub-process extracts all the features of characters and produces the pixels according to feature of character. Then output of this process is used to train the neural network and produces the final result. Thus, recognition of handwritten characters is not a single process; it is the combination of many processes such as scanning of characters called as digitization, pre-processing, segmentation, feature extraction, classification and post-processing. By the combination of these processes, the desired output is obtained.

For recognition of handwritten characters, HGCR machine depends on learning process in which back propagation algorithm takes input from user. In this learning process, training and testing of characters is done. A library is made in which the information of text segments is stored which are used for future comparisons. This library helps in acceptance and rejection of characters. For example: Character  is written 50 times in library in which 40 fonts are used for training of neural network and rest are used for testing the network. The importance of neural networks has grown dramatically during the last fifteen years. A large number of universities and companies are using neural networks and products based on neural network are available in market.

Neural network works as human brain, thus the architecture of neural network machine is same as the architecture of human brain. There are hundreds or even thousands of neurons in custom integrated circuits. In conjunction, the growing interest in learning machines, non-linear dynamics and parallel computation spurred renewed attention in artificial neural networks. There are so many real world applications such as pattern recognition, system identification, noise removal etc. in which neural network is

extensively applied. There are also positive results when it is used in character recognition. Many non-Indian and Indian scripts are recognized accurately and efficiently with the help of neural networks.

There are too many applications, which can be solved easily using neural networks that are difficult to solve using traditional methods. A neural network is consisted of three layers i.e. input layer, hidden layer and the output layer. Each layer consists of small-interconnected processing units. These units are interconnected with each other via a weighted link. Every unit has individual function but the combination of these units display complex behaviors. A neural network is a massively parallel-distributed processor that has a natural propensity for storing experimental knowledge and making it available for use. Neural network behaves like human brain. Neural network acquires knowledge as human brain acquires knowledge from learning process. Neural network has many advantages over traditional systems. For example: - It is insensible to noise and easier to handle, because it involves less human work than other traditional statistical analysis. Neural network can solve problems that do not have an algorithmic solution or problems whose algorithmic solutions are too complex to find. It has fewer errors because it can respond to anything and small changes in the input normally do not cause a change in the output. This behavior of neural network shows its importance.

1.1 Background Information

1.1.1 Character Recognition

Character recognition is a process, which associates a symbolic meaning with objects (letters, symbols and numbers) drawn on an image, *i.e.,* character recognition techniques associate a symbolic identity with the image of a character. Mainly, character recognition machine takes the raw data that further implements the process of preprocessing of any recognition system. On the basis of that data acquisition process, character recognition system can be classified into following categories: -

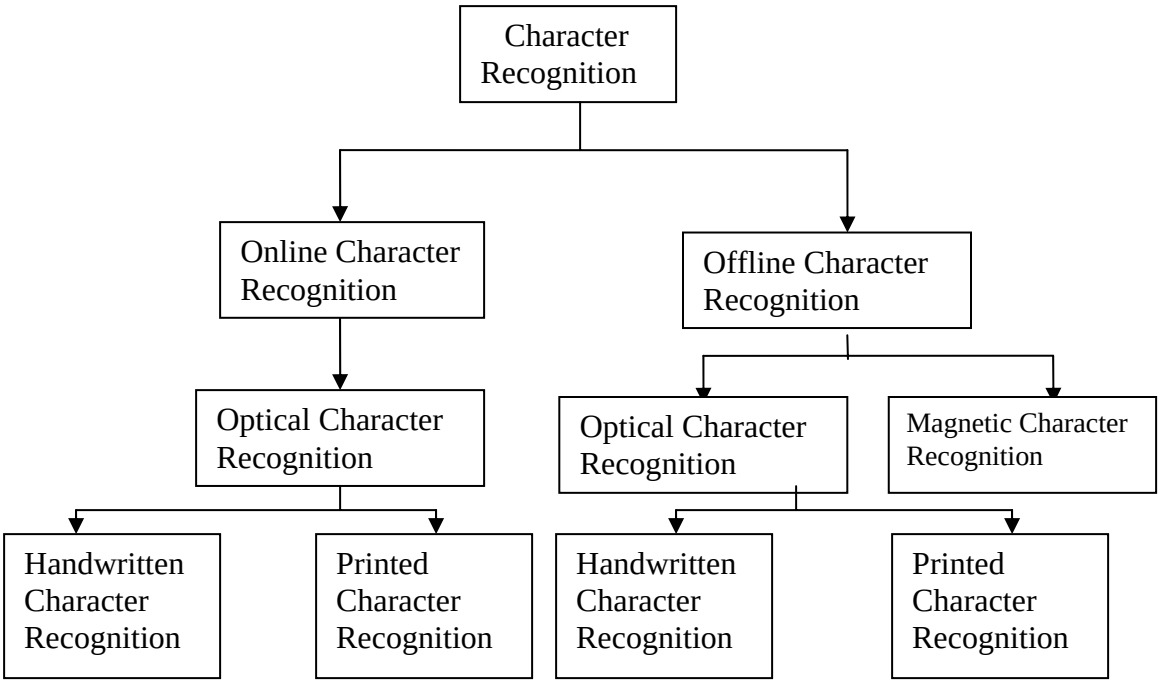

Figure 1: Classification of Character Recognition

Character recognition can be categorized into following two parts: -

1. Online Character Recognition
2. Offline Character Recognition

Off-line handwriting recognition refers to the process of recognizing 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 case of online handwritten character recognition, 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 recognizing 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. The major difference between Online and Offline Character Recognition is that Online Character Recognition has real time contextual information but offline data does not

[2]. This difference generates a significant divergence in processing architectures and methods. The offline character recognition can be further grouped into two types:

- Magnetic Character Recognition (MCR)
- Optical Character Recognition (OCR)

In MCR, the characters are printed with magnetic ink. The reading device can recognize the characters according to the unique magnetic field of each character. MCR is mostly used in banks for check authentication. OCR deals with the recognition of characters acquiring by optical means, typically a scanner or a camera. The characters are in the form of pixelized images, and can be either printed or handwritten, of any size, shape, or orientation. The OCR can be subdivided into handwritten character recognition and printed character recognition. Handwritten Character Recognition is more difficult to implement than printed character recognition due to diverse human handwriting styles and customs. In printed character recognition, the images to be processed are in the forms of standard fonts like Times New Roman, Arial, Courier, etc.

1.1.2 Handwritten Gurumukhi Character Recognition

Handwritten Gurumukhi Character Recognition (HGCR) is an area of pattern recognition that has been the subject of considerable research since last some decades. There are too many applications (i.e. Indian offices such as bank, sales-tax, railway, embassy, etc.) the both English and regional languages are used. Many forms and applications are filled in regional languages and sometimes those forms have to be scanned directly. If there is no HGCR system, then image is directly captured and there is no option for editing those documents. Handwritten character recognition (HCR) is a process of automatic computer recognition of characters in optically scanned and digitized pages of text. The main objective of an HGCR system is to recognize alphabetic Gurumukhi characters, which are in the form of digital images, without any human intervention. This is done by searching a match between the features extracted from the given character's image and the library of image models. The library helps in distinction of features between the character images; this eliminates the confusion for correct character recognition . Firstly HGCR system takes data input from user and leads to preprocessing stage, feature extraction and this extracted image search matching from

image model library and then classifying the characters. The basic process for Handwritten Gurumukhi character recognition is as follows: -

1.1.2.1 Digitization

The handwritten data is converted into digital form either by scanning the writing on paper (i.e. offline characters) or by writing with a special pen on an electronic surface such as digitizer combined with a LCD (i.e. online characters). For online handwritten characters, numbers of strokes made by the writer are available but in case of off-line, the document is scanned and gets only the image of that document.

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 this case, no information regarding pen-strokes etc. is available. The comparison between offline and online characters is shown in table 1. Digitization produces the digital image, which is fed to the pre-processing phase.

1.1.2.2 Pre-processing

In HGCR, typical preprocessing operations include binarization, contour smoothing, noise reduction, skew detection, and skeletonization of a digital image so that subsequent algorithms along the road to final classification can be made simple and more accurate. The corresponding objectives of Pre-processing methods are as follows: -

- Noise Removal - The major objective of noise removal is to remove any unwanted bit-patterns, which do not have any significance in the output.
- Contour Smoothing - The objective of contour smoothing is to smooth contours of broken and/or noisy input characters.
- Skew Detection – Skew Detection refers to the tilt in the bitmapped image of the scanned paper for OCR. It is usually caused if the paper is not fed straight into the scanner. Most of the 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.
- Skeletonization - 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.

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

The raw data, depending on the data acquisition type, is subjected to a number of preliminary processing steps to make it usable in the descriptive stages of character analysis. Preprocessing aims to produce data that are easy for the HGCR system to operate accurately.

1.1.2.3 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 systems. Different methods used can be classified based on the type of text and strategy being followed like straight segmentation method, recognition-based segmentation and cut classification method. In order to achieve broad utility, it is important that a segmentation method have the following properties:

1. Capture perceptually important groupings or regions, which often reflect global aspects of the image. Two central issues are to provide precise characterizations of what are perceptually important, and to be able to specify what a given segmentation technique does. There should be precise definitions of the properties of a resulting segmentation, in order to better understand the method as well as to facilitate the comparison of different approaches.

2. In order to be of practical use, segmentation methods should run at speeds similar to edges detection or other low-level visual processing techniques, meaning nearly linear time and with low constant factors. For example, a segmentation technique that runs at several frames per second can be used in video processing applications.

1.1.2.4 Feature extraction

Each character has some features, which play an important role in pattern recognition. Punjabi characters have many particular features. Feature extraction describes the relevant shape information contained in a pattern so that the task of classifying the pattern is made easy by a formal procedure. Feature extraction stage in HGCR system analyses these Punjabi character segment and selects a set of features that can be used to uniquely identify that character segment. Mainly, this stage is heart of HGCR system because output depends on these features. 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. Among the different design issues involved in building a recognizing system, perhaps the most significant one is the selection of set of features.

Feature extraction for exploratory data projection enables high-dimensional data visualization for better data structure understanding and for cluster analysis. In feature extraction for classification, it is desirable to extract high discriminative reduced-dimensionality features, which reduce the classification computational requirements. However, feature extraction criteria for exploratory data projection regularly aim to minimize an error function, such as the mean square error or the inter pattern distance difference whereas feature extraction criteria for classification aim to increase class separability as possible. Hence, the optimum extracted features (regarding a specific

criterion) calculated for exploratory data projections are not necessarily the optimum features regarding class separability and vice versa. In particular, two or more classes may have principal features that are similar. Moreover, feature extraction for exploratory data projection is used for two or three-dimensional data visualization, whereas classification usually needs more than two or three features. Consequently, feature extraction paradigms for exploratory data projection are not generally used for classification and vice versa.

1.1.2.5. Representation of Character Features

After extracting the features, the data should be represented in one of two ways, either as a boundary or as a complete region. When the focus is on external shape characteristics such as corners and inflections then boundary representation is appropriate. While regional representation is appropriate when the focus is on internal properties such as textures or skeletal shape. In some applications like character recognition these representations coexist, which often require algorithm based on boundary shape as well as skeletons and other internal properties. In terms of character recognition descriptors such as holes and bays are powerful features that help differentiate one part of the character from another.This description also called feature selection, deals with extracting features which results in some quantitative information of interest or features that are basic for differentiating one class of objects from another.

1.2. Architecture of HGCR

An HGCR (Handwritten Gurumukhi Character Recognition) is a system which loads a character image, preprocesses the image, extracts proper image features, classify the characters based on the extracted image features and the known features are stored in the image model library, and recognizes the image according to the degree of similarity between the loaded image and the image models. In preprocessing noise is removed from image so that recognition algorithms can be applied on image easily. Some feature extraction algorithms only deal with the contours of the image while some algorithms calculate every pixel of the image [3]. The preprocessing stage, which includes binarization, edge detection, gap filling, and segmentation and so on, can make the initial image more suitable for later computation.

This HGCR system maintains a library, which is created by initial users and stores the data in each and every file in the binary form. Different shapes drawn by initial users maintain the image model library that further uses in recognition process to match the pattern of original image drawn by new users else shows the percentage match with similar character if there any and this is performed using neural network training.

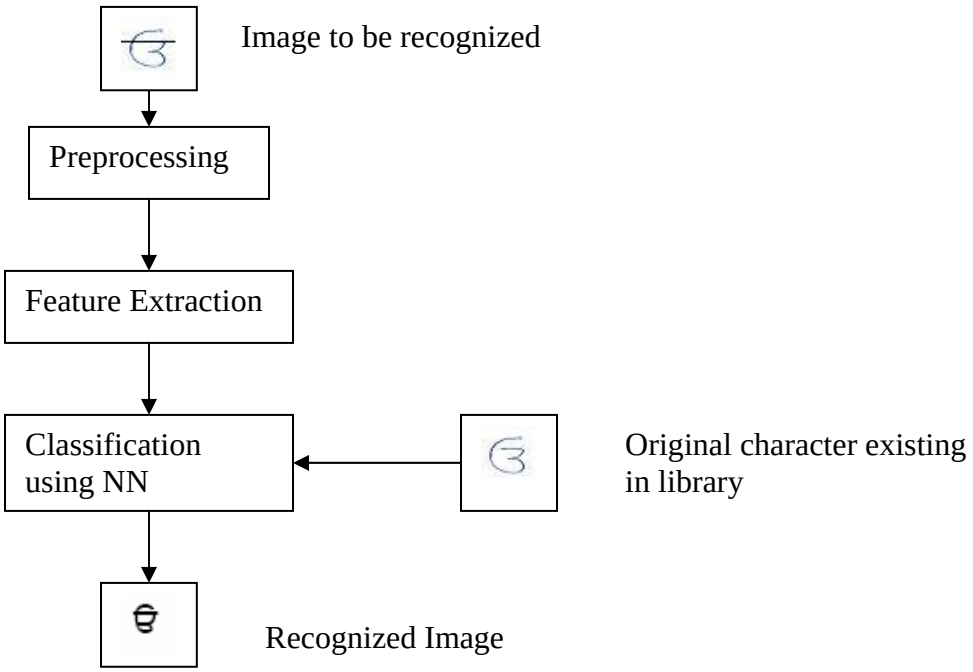

Figure 2: Recognition of Gurumukhi Character

For example- if there is an image of a character, which has small difference from feature of existing characters in the library, then neural network identify the most resembling features and recognize that character accurately. This feature can be shown in figure 2.Here the aim is to develop a machine that can recognize gurumukhi characters based on learning of data, this machine is learned by user using hundred numbers of characters. The learning process can be at the run time also. When user is learning the system it goes to be training for different user inputs by the help of neural networks. Once the system is being trained then testing process is performed and obtains maximum accuracy of recognition of character image.

1.3 Application of Offline Handwritten Recognition

Some of the more important applications of offline handwritten recognition are discussed in the following section:

• **Cheque Reading**: Offline handwritten recognition is basically used for cheque reading in banks. Cheque reading is the very important commercial application of offline handwritten recognition. Handwritten recognition system plays very important role in banks for signature verification and for recognition of amount filled by user.

• **Postcode Recognition:** Handwritten recognition system can be used for reading the handwritten postal address on letters. Offline handwritten recognition system used for recognition handwritten digits of postcode. HCR can be read this code and can sort mail automatically.

• **Form Processing:** HCR can be also used for form processing. Forms are normally used for collecting the public information. Replies of public information can be handwritten in the space provided.

• **Signature Verification:** HCR can be also used for identify the person by signature verification. Signature identification is the specific field of handwritten identification in which the writer is verified by some specific handwritten text. Handwritten recognition system can be used for identify the person by handwriting, because handwriting may be vary from person to person.

1.4 Artificial Neural Network

An Artificial Neural Network (ANN) is an information-processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. ANNs, like people, learn by example. An ANN is configured for a specific application, such as pattern recognition or data classification through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons. This is true of ANNs as well. An Artificial Neural Network is a network of many very simple

processors ‘units’, each possibly having a small amount of local memory. The units are connected by unidirectional communication channels ‘connections’, which carry numeric as opposed to symbolic data. The units operate only on their local data and on the inputs they receive via the connections.

The design motivation is what distinguishes neural networks from other mathematical techniques: A neural network is a processing device, either an algorithm, or actual hardware, whose design was motivated by the design and functioning of human brains and components thereof.

There are many different types of Neural Networks, each of which has different strengths particular to their applications. The abilities of different networks can be related to their structure, dynamics and learning methods.

Neural Networks offer improved performance over conventional technologies in areas which includes: Machine Vision, Robust Pattern Detection**,** Signal Filtering**,** Virtual Reality, Data Segmentation**,** Data Compression**,** Data Mining**,** Text Mining, Artificial Life**,** Adaptive Control**,** Optimization and Scheduling**,** Complex Mapping and more.

1.4.1 Biological Neuron

The brain is principally composed of about 10 billion neurons; each connected to about 10,000 other neurons. Each of the yellow blobs in the picture above is neuronal cell bodies (soma), and the lines are the input and output channels (dendrites and axons), which connect them. Each neuron receives electrochemical inputs from other neurons at the dendrites. If the sum of these electrical inputs is sufficiently powerful to activate the neuron, it transmits an electrochemical signal along the axon, and passes this signal to the other neurons whose dendrites are attached at any of the axon terminals. These attached neurons may then fire. It is important to note that a neuron fires only if the total signal received at the cell body exceeds a certain level. The neuron either fires or it does not, there are not different grades of firing. So, our entire brain is composed of these interconnected electro-chemical transmitting neurons. From a very large number of extremely simple processing units (each performing a weighted sum of its inputs, and then firing a binary signal if the total input exceeds a certain level) the brain manages to perform extremely complex tasks.

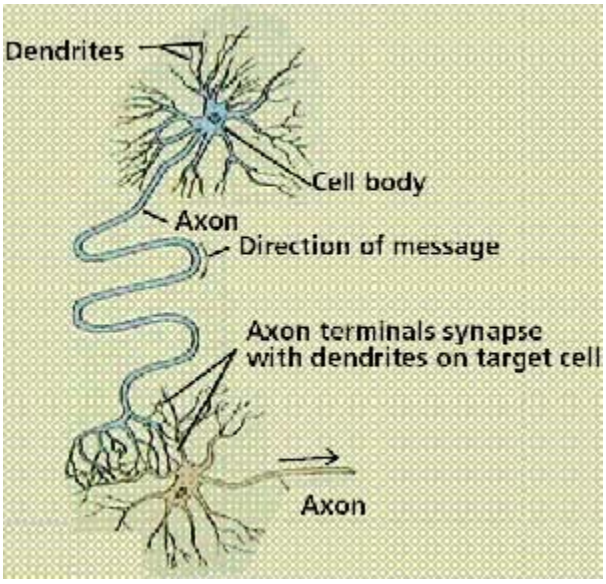

Figure 3: Biological Neuron

This is the model on which artificial neural networks are based. Thus far, artificial neural networks haven't even come close to modeling the complexity of the brain, but they have shown to be good at problems which are easy for a human but difficult for a traditional computer, such as image recognition and predictions based on past knowledge.

- **How the Human Brain Learns?**

Much is still unknown about how the brain trains itself to process information, so theories abound. 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 stand 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 or excite activity in the connected neurons. 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 one neuron on another changes.

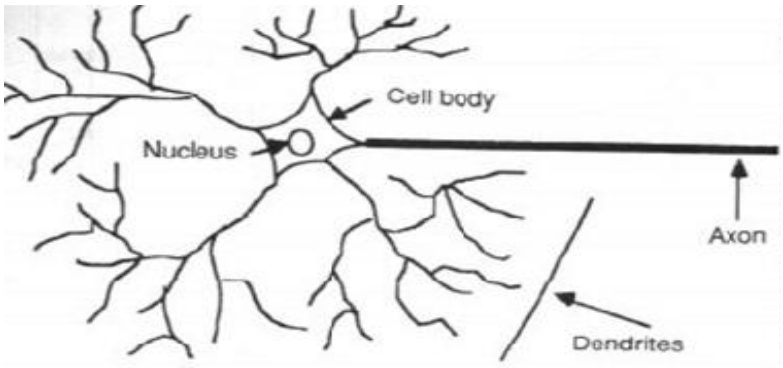

Figure 4: Components of a neuron

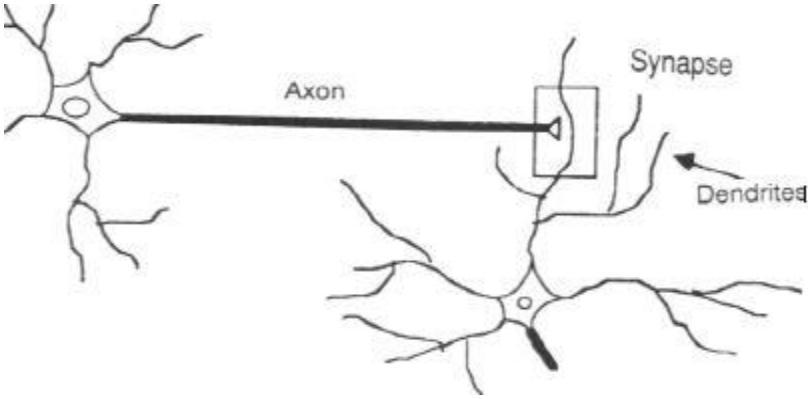

Figure 5: Synapse

An Artificial Neural Network (ANN) is an information-processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. ANNs, like people, learn by example. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons [44].

Artificial neural networks have been studied for many years in the hope of achieving human– like performance in the fields of speech and image recognition. These networks composed of many nonlinear computational elements operating in parallel and arranged in patterns reminiscent of biological neural nets.

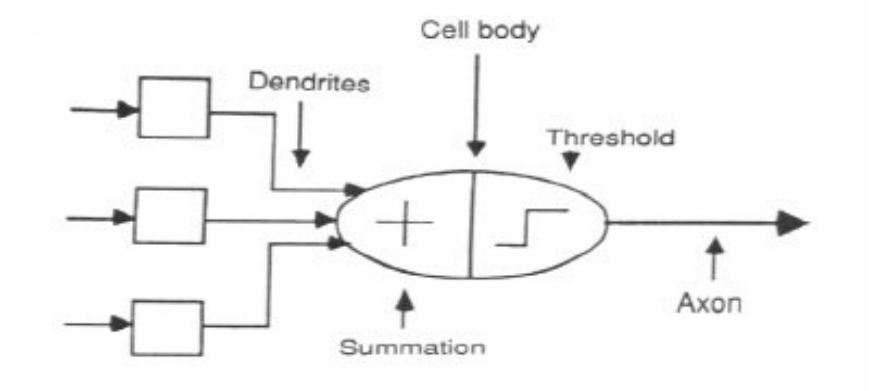

Figure 6: The Neuron Model

Neural Networks (NNs) are simplified imitations of the central nervous system, and obviously therefore, have been motivated by the kind of computing performed by the human brain. The structural constituents of a human brain termed neurons are the entities, which perform computations such as cognition, logical inference, pattern recognition and so on. Hence the technology, which has been built on a simplified imitation of computing by neurons of a brain, has been termed Artificial Neural Systems (ANS) technology or Artificial Neural Networks (ANN) or simply Neural Networks. A human brain develops with time and this, in common parlance is known as experience. An artificial neuron is a device with many inputs and one output.

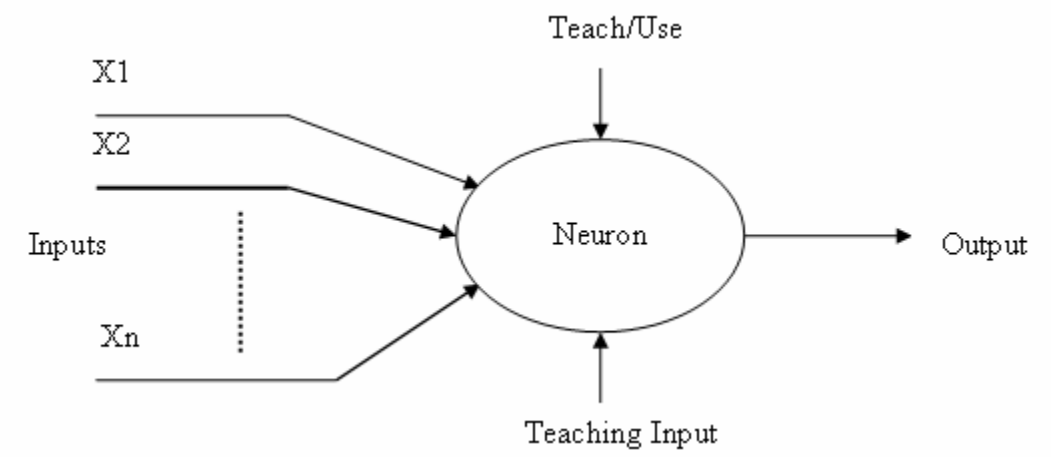

Figure 7: An artificial neuron with n inputs and one output

The stability-plasticity issue is of great importance to NN architectures. The NN needs to remain plastic to significant or useful information but remain stable when presented with irrelevant information.

1.4.2 A More Complicated Neuron

The previous neuron doesn't do anything that conventional computers don't do already. A more sophisticated neuron (figure 5) is the McCulloch and Pitts model (MCP). The difference from the previous model is that the inputs are ‘weighted’; the effect that each input has at decision making is dependent on the weight of the particular input. The weight of an input is a number which when multiplied with the input gives the weighted input. These weighted inputs are then added together and if they exceed a pre-set threshold value, the neuron fires. In any other case the neuron does not fire.

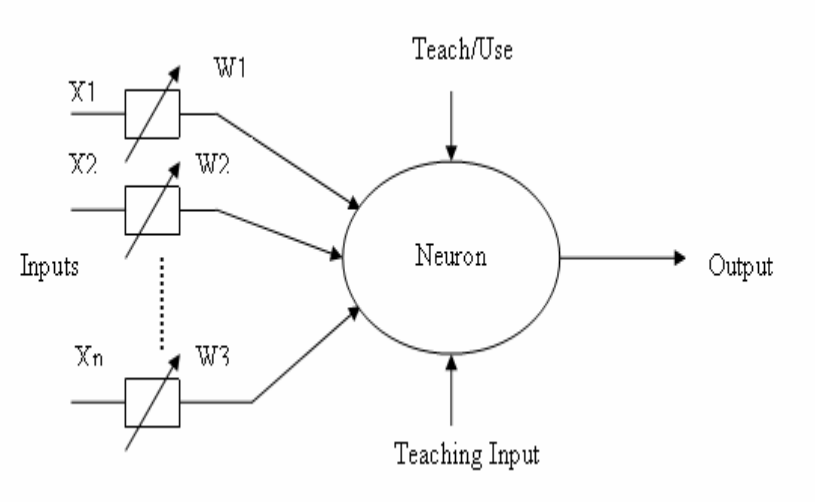

Figure 8: A Complicated Neuron

In mathematical terms, the neuron fires if and only if;

$$X1W1 + X2W2 + X3W3 + ... > T$$

The addition of input weights and of the threshold makes this neuron a very flexible and powerful one. The MCP neuron has the ability to adapt to a particular situation by changing its weights and/or threshold. Various algorithms exist that cause the neuron to 'adapt' the most used ones are the Delta rule and the back error propagation. The former is used in feed-forward networks and the latter in feedback networks.

Chapter 2 Literature Review

OCR is one of the oldest ideas in the history of pattern recognition using computers. In recent time, Punjabi character recognition becomes the field of practical usage. In character recognition, the process starts with reading of a scanned image of a series of characters, determines their meaning, and finally translates the image to a computer written text document. Mainly, this process is done commonly in the post-offices to mechanically read names and addresses on envelopes and by the banks to read amount and number on cheques. Also, companies and civilians can use this method to quickly translate paper documents to computer written documents.

Many researches have been done on character recognition in last 56 years. Some books [6-8] and many surveys [4, 5] have been published on the character recognition. Most of the work on character recognition has been done on Japanese, Latin, Chinese characters in the middle of 1960s. The work by Impedovo *et al*. [9] focuses on commercial OCR systems. Jain *et al.* [10] summarized and compared some of the well-known methods used in various stages of a pattern recognition system. They have tried to identify research topics and applications, which are at the forefront in this field. Pal and Chaudhuri [8] in their report summarized different systems for Indian language scripts recognition. They have described some commercial systems like Bangla and Devnagiri OCRs. Manish [11] in his survey report summarized a system for the recognition of Punjabi characters. They reported the scope of future work to be extended in several directions such as OCR for poor quality documents, for multi font OCR and bi-script/multi-script OC*R* development *etc.* A bibliography of the fields of OCR and document analysis is given in [12]. Tappet *et al*. [13] and Wakahara *et al*. [14] worked on-line handwriting recognition and described a distortion-tolerant shape matching method. Noubound and Plamondon [15] and Suen *et al*. [16] proposed methods used for on-line recognition of hand-printed characters while Connell *et al*. [17, 18] described on-line character recognition for Devanagari characters and alphanumeric characters. Bortolozzi *et al.* [19] have published a very useful study on recent advances in handwriting recognition. Lee *et al*. [20] described off-line recognition of totally unconstrained handwritten numerals using multiplayer cluster neural network. The

character regions are determined by using projection profiles and topographic features extracted from the gray-scale images. Then, a nonlinear character segmentation path in each character region is found by using multi-stage graph search algorithm.

Khaly and Ahmed [21], Amin [22] and Lorigo & Govindraju [23] have produced a bibliography of research on the Arabic optical text recognition. Hildebrandt and Liu [24] have reported the advances in handwritten Chinese character recognition and Liu *et al.* [25] have discussed various techniques used for on-line Chinese character recognition.

2.1 Indian Script Recognition

As compared to English and Chinese languages, the research on OCR of Indian language scripts has not achieved that perfection. Few attempts have been carried out on the recognition of Indian character sets on Devanagari, Bangla, Tamil, Telugu, Oriya, Gurmukhi, Gujarati and Kannada. These attempts are briefly described in the following sub-sections.

2.1.1 Recognition of Handwritten Devnagari Scripts

Devnagari is the most popular script in India. Devnagari script is used to write many Indian languages such as Hindi, Marathi, Rajasthani, Sanskrit and Nepali. The characters of Hindi Language are shown in figure 9.

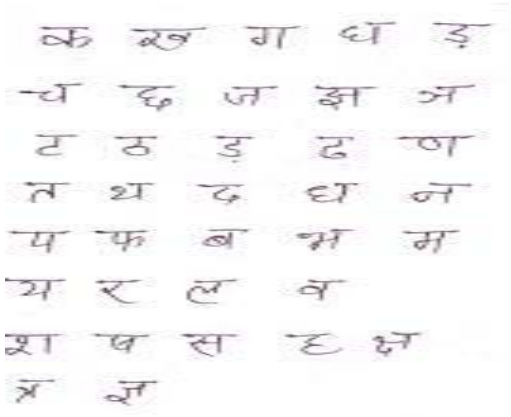

Figure 9: Handwritten Hindi Characters

The work on Handwritten Devnagari character recognition started early in 1977. Firstly in 1977, I. K. Sethi and B. Chatterjee [26] presented a system for handwritten Devnagari characters. In this system, sets of very simple primitives were used. Most of the decisions were taken on the basis of the presence/absence or positional relationship of these

primitives. A multistage process was used for taking these decisions. By completion of each stage, the options for making decision regarding the class membership of the input token decreases. In 1979, Sinha and Mahabala [27] presented a syntactic pattern analysis system with an embedded picture language for the recognition of handwritten and machine printed Devnagari characters. In this system, mainly feature extraction technique was used. Sethi and Chatterjee [28] also have done some studies on hand-printed Devnagari numerals which is based upon binary decision tree classifier and that binary decision tree was made 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. That decision process was also based on multistage process. Brijesh K. Verma [29] presented a system for HCR using Multi-Layer Perceptron (MLP) networks and the Radial Basis Function (RBF) networks in the task of handwritten Hindi Character Recognition (HCR). The error back propagation algorithm was used to train the MLP networks.

2.1.2. Recognition of Bangla Characters

Among all the Indian scripts, the maximum work for recognition of handwritten characters has been done on Bangla characters. Handwritten Bangla characters are shown in figure 10. For offline handwritten Bangla numerals and characters recognition, some OCR systems are available in market. In 1982, S. K. Parui and B.B. Chaudhuri et al. [30] proposed a recognition scheme using a syntactic method for connected Bangla handwritten numerals. By using some automation some sub-patterns are made on the basis of these one-dimensional strings of eight direction codes. In 1998, A.F.R. Rahman and M. Kaykobad [31] proposed a complete Bangali OCR system in which they used hybrid approach for recognition of handwritten Bangla characters. Everybody have different writing style. For this purpose, Pal and Chaudhuri [32] proposed a robust scheme for the recognition of isolated Bangla off-line handwritten numeral. In this scheme, the direction of numeral, height and position of numeral with respect to the character bounding box, shape of the reservoir etc. are used for recognition. Dutta and Chaudhuri [34] reported a work on recognition of isolated Bangla alphanumeric handwritten characters using neural networks. In this method, the primitives are used for representing the characters and structural constraints between the primitives imposed by

the junctions present in the characters. Neural network approach is also used by Bhattacharya et al. [33] for the recognition of Bangla handwritten numeral. In this, certain features like loops, junctions, etc. present in the graph are considered to classify a numeral into a smaller group. Sural and Das [35] defined fuzzy sets on Hough transform of character pattern pixels from which additional fuzzy sets are synthesized using t-norms. Garain et al. [36] proposed an online handwriting recognition system for Bangla. A low complexity classifier has been designed and the proposed similarity measure appears to be quite robust against wide variations in writing styles.

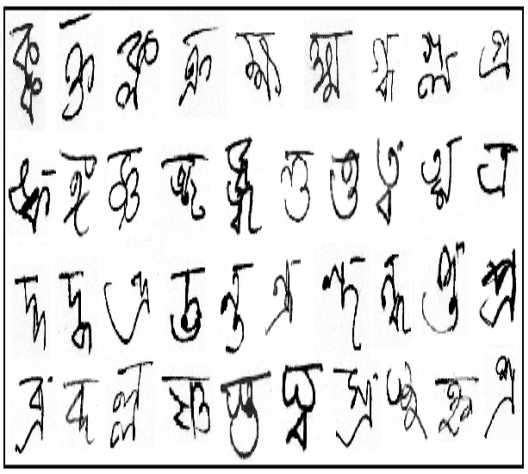

Figure 10: Handwritten Bangla Characters [37]

U. Pal, Wakabayashi and F. Kimura [37] proposed a recognition system for handwritten offline compound Bangla characters using Modified Quadratic Discriminate Function (MQDF). The features used for recognition purpose are mainly based on directional information obtained from the arc tangent of the gradient. To get the feature, at first, a 2 X 2 mean filtering is applied 4 times on the gray level image and non-linear size normalization is done on the image.

2.1.3 Recognition of Tamil Characters

The work on recognition of Tamil characters started in 1978 by Siromony *et al*. [38]. They described a method for recognition of machine-printed Tamil characters using an encoded character string dictionary. The scheme employs string features extracted by row- and column- wise scanning of character matrix. Features in each row (column) are encoded suitably depending upon the complexity of the script to be recognised.

Chandrasekaran *et al.* [39] used similar approach for constrained hand-printed Tamil character recognition. Chinnuswamy and Krishnamoorthy [40] presented an approach for hand-printed Tamil character recognition employing labeled graphs to describe structural composition of characters in terms of line-like primitives. Recognition is carried out by correlation matching of the labeled graph of the unknown character with that of the prototypes.

A piece of work on on-line Tamil character recognition is reported by Aparna *et al.* [41]. They used shape- based features including dot, line terminal, bumps and cusp. Comparing an unknown stroke with a database of strokes does stroke identification. Finite state automation has been used for character recognition with an accuracy of 71.32-91.5%.

2.1.4 Recognition of Telugu Characters

A two-stage recognition system for printed Telugu alphabets has been described by Rajasekaran and Deekshatulu [42]. In the first stage a directed curve tracing method is employed to recognize primitives and to extract basic character from the actual character pattern. In the second stage, the basic character is coded, and on the basis of the knowledge of the primitives and the basic character present in the input pattern, the classification is achieved by means of a decision tree. Lakshmi and Patvardhan [43] presented a Telugu OCR system for printed text of multiple sizes and multiple fonts. After pre-processing, connected component approach is used for segmentation characters. Real valued direction features have been used for neural network based recognition system. The authors have claimed an accuracy of 98.6%. Negi *et al*. [2] presented a system for printed Telugu character recognition, using connected components and fringe distance based template matching for recognition. Fringe distances compare only the black pixels and their positions between the templates and the input images.

2.1.5 Recognition of Gurmukhi Characters

Gurmukhi script is used primarily for writing Punjabi language. Punjabi Language is spoken by eighty four million native speakers and is the world's 14th most widely spoken language. Lehal and Singh [30] developed a complete OCR system for printed Gurmukhi

script where connected components are first segmented using thinning based approach. They started work with discussing useful pre-processing techniques. Lehal and Singh [30] have discussed in detail the segmentation problems for Gurmukhi script. They have observed that horizontal projection method, which is the most commonly used method employed to extract the lines from the document, fails in many cases when applied to Gurmukhi text and results in over segmentation or under segmentation. The text image is broken into horizontal text strips using horizontal projection in each row. The gaps on the horizontal projection profiles are taken as separators between the text strips. Each text strip could represent: a) Core zone of one text line consisting of upper, middle zone and optionally lower zone (core strip), b) upper zone of a text line (upper strip), c) lower zone of a text line (lower strip), d) core zone of more than one text line (multi strip). Then using estimated average height of the core strip and its percentage they identify the type of each strip. The classification process is carried out in three stages. In the first stage, the characters are grouped into three sets depending on their zonal position, *i.e.*, upper zone, middle zone and lower zone. In the second stage, the characters in middle zone set are further distributed into smaller sub-sets by a binary decision tree using a set of robust and font independent features. In the third stage, the nearest neighbor classifier is used and the special features distinguishing the characters in each subset are used. This enhances the computational efficiency. The system has an accuracy of about 97.34%. An OCR postprocessor of Gurmukhi script is also developed. In last, Lehal and Singh and Lehal *et al.* Proposed a post-processor for Gurmukhi OCR where statistical information of Punjabi language syllable combinations and certain heuristics based on Punjabi grammar rules have been considered. There is also some literature dealing with segmentation of Gurmukhi Script. Lehal and Singh have performed segmentation of Gurmukhi script by connected component analysis of a word assuming the headline not being a part of the word. Goyal *et al.* have suggested a dissection based Gurmukhi character segmentation method, which segments the characters in the different zones of a word by examining the vertical white space. Manish [11] proposed an algorithm for recognizing Gurumukhi scripts. In his work he recognized Punjabi characters with the efficiency of 92.56 %. In Chinese, Latin the efficiency of recognition of words is over 99%.

Chapter 3 Problem Statement

3.1 Introduction

On the basis of literature survey of character recognition algorithms, one question to be answered is: which algorithm is the best choice for a given application? This question leads the thesis to characterize the available algorithms so that the most efficient methods can be sorted out for different applications. An experimental approach needs to be developed to compare and evaluate the performance of different invariants of shape-based Gurmukhi Characters.

An image with 1280×800 pixels will certainly take much longer time to compute than a 32×32 image. The investigations of the reconstruction of the region-based Gurmukhi character's image are a major motivation for the work. Using neural networks, recognition of handwritten Gurmukhi character is a good idea. However, in the practical image acquisition systems and conditions, shape distortion is common processes in HGCR system because of different people handwriting have different shape of characters.

The observed character image is being representing only a degraded version of the original character image. Recognition of Gurmukhi characters that are of various shapes is a goal of recent research. Gurumukhi characters using neural networks having Back propagation algorithms are recognized.

3.2 Development of an HGCR System

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

3.2.1 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 different types of

binary images drawn by different users using different styles of handwriting are stored. The following are the image samples of Gurumukhi characters:-

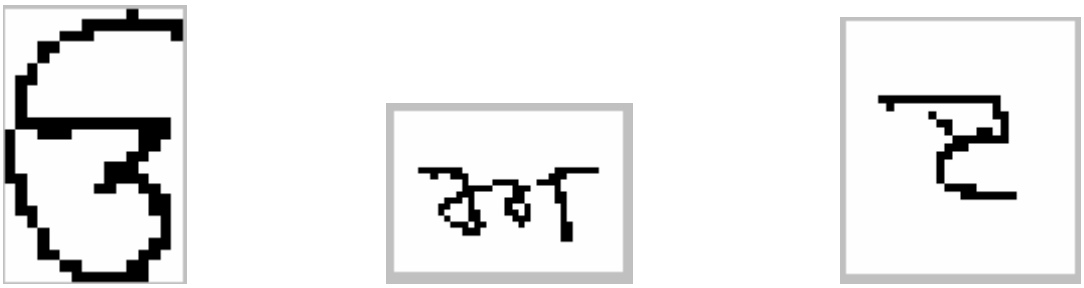

Figure 11: Samples of Offline Handwritten Gurumukhi Character Images ਉ, ਅ, ਏ

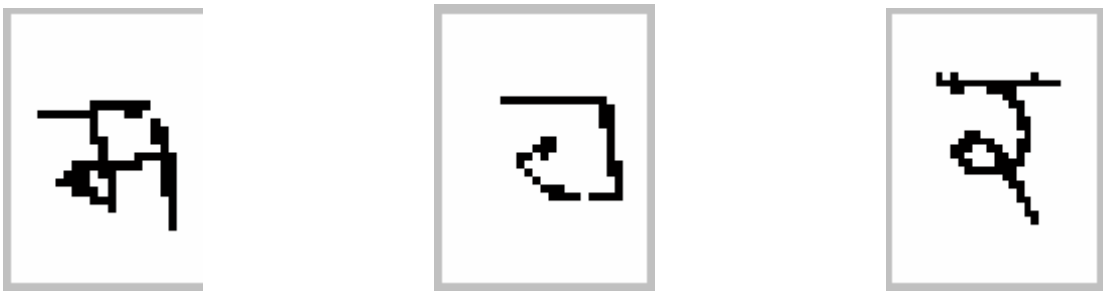

Figure 12: Samples of Offline Handwritten Gurumukhi Character Images ਮ, ਹ, ਰ

Figure 13: Samples of Offline Handwritten Gurumukhi Character Images ਖ, ਗ, ਯ

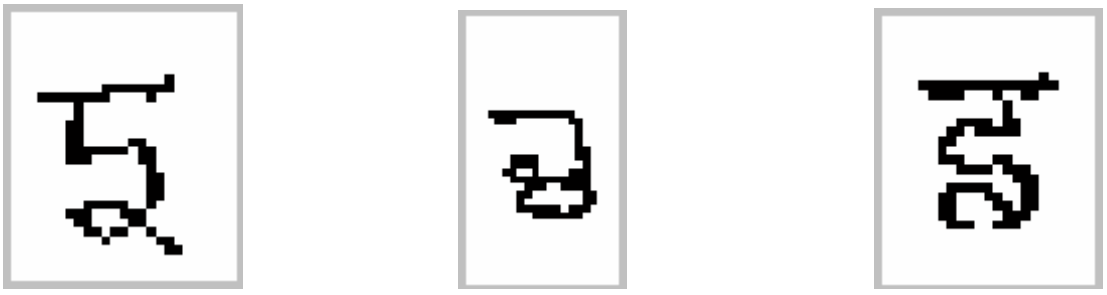

Figure 14: Samples of Offline Handwritten Gurumukhi Character Images ਨ, ਚ, ਙ

Figure 15: Samples of Offline Handwritten Gurumukhi Character Images ਜ, ਝ, ਞ

Figure 16: Samples of Offline Handwritten Gurumukhi Character Images ਟ, ਠ, ਡ

Figure 17: Samples of Offline Handwritten Gurumukhi Character Images ਢ, ਣ, ਤ

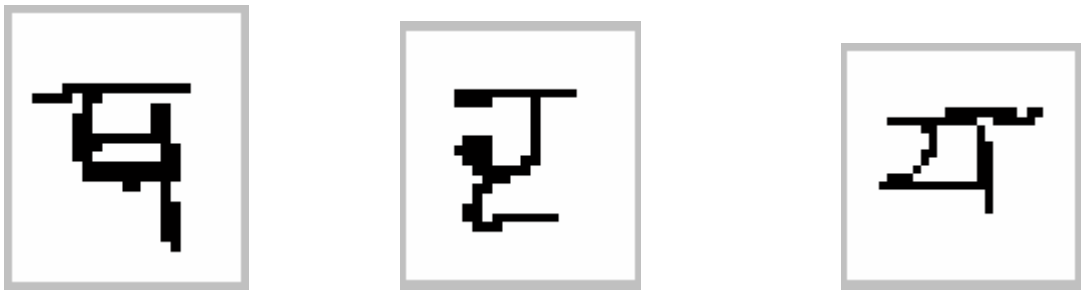

Figure 18: Samples of Offline Handwritten Gurumukhi Character Images ਥ, ਦ, ਧ

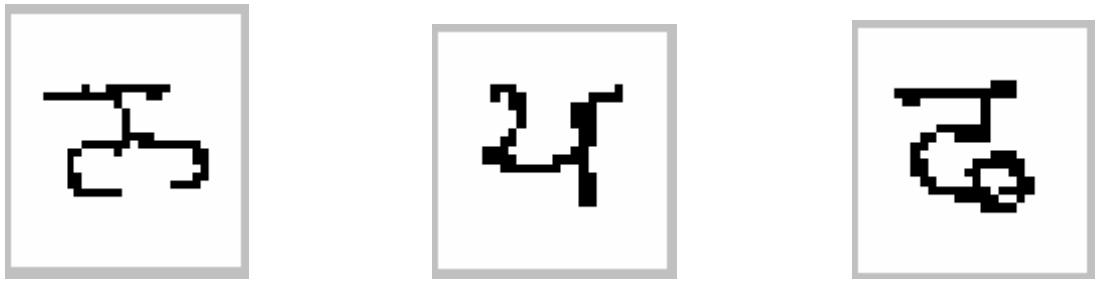

Figure 19: Samples of Offline Handwritten Gurumukhi Character Images ਨ, ਪ, ਫ

Figure 20: Samples of Offline Handwritten Gurumukhi Character Images ਘ, ਤ, ਮ

Figure 21: Samples of Offline Handwritten Gurumukhi Character Images ਜ, ਰ, ਲ

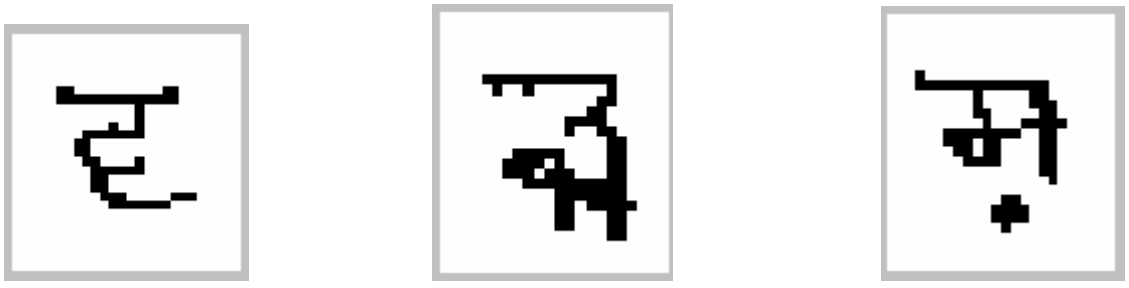

Figure 22: Samples of Offline Handwritten Gurumukhi Character Images ਵ, ਖ, ਸ਼

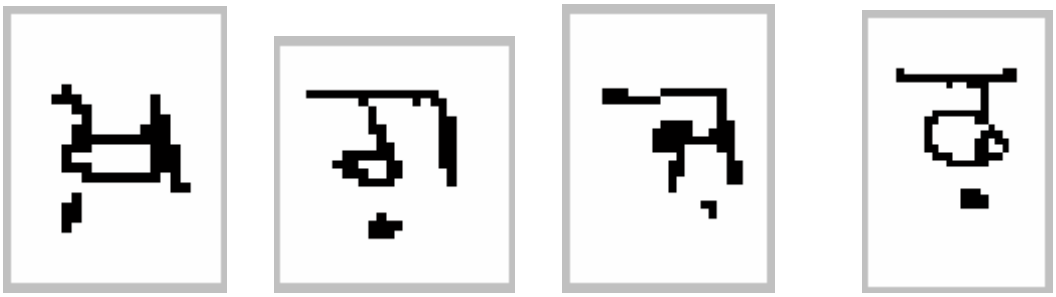

Figure 23: Samples of Offline Handwritten Gurumukhi Character Images ਅ, ਸ਼ , ਜ਼, ਫ਼

3.3 Detailed Description

This work has been carried out to implement multi-input multi layered Neural Network (parallel distributed system) for the purpose of recognition of Punjabi characters, which is trained using back propagation, for the final use of this trained network to recognize the patterns trained for, and classify these under different, distinct output classes which the network was trained to group them under.

This problem is divided into two phases:

1) Reading a windows image format

2) Development of Artificial Neural Network model.

Second phase is further divided into two sub-phases:

a) Training phase of neural network

b) Testing phase of neural network.

3.3.1 Window Image Format

There exist so many popular formats for digital image files, which are usually referred to by the file extension used. The more common formats include BMP, JPG and GIF. Every format has some pros and cons associated with the subject of image files, the window BMP file offers significant advantages because of its simplicity and it also has the advantage of being highly standardized and extremely widespread. Its principal disadvantage is it does not support effective image compression except on specific types of images. However, there is a hidden benefit to this shortcoming. Since other image file formats do offer significant data compression, very few images stored in BMP format are

compressed and, therefore, a simple BMP editor that doesn't support working with compressed BMP files will seldom encounter a file it can't work.

The file is a binary file (as opposed to a text file) that is separated into four sections (though one, the color table, does not always exist). The sections, in order from the start of the file, are the File Header, the Image Header, the Color Table, and finally the pixel data itself. The File Header is used primarily for the accessing software to confirm that the file is (at least probably) a BMP file, to tell exactly how large the file is, and to learn where the actual image is located within the file. The Image Header gives the detailed information about the image and its data format such as the height and width of the image, how many bits are used per pixel, and whether the image data is compressed or not. The color table may or may not be present, depending on the format of the image data. When it does exist, it is either a color palette or a set of bit masks used to extract the color information from the image data. Firstly, the rest of the file is the image data itself. Each of these four sections is discussed in some detail below.

3.3.1.1 The File Header

The File Header has exactly fourteen bytes in it. The first two bytes must be the ASCII codes for the characters 'B' and 'M'. The program should check these values to confirm that the file it is reading is most probably a Windows BMP file. The second field is a four byte integer that contains the size of the file in bytes.

Table 2: Bitmap File Header: BMP file Header contents

Field Name	Size in Bytes	Description
BfType	2	The characters 'BM'
BfSize	4	The size of the file in Bytes
bfReserved1	2	Unused- must be zero
bfResevered2	2	Unused- must be zero
BfOffBits	4	Offset to start of Pixel Data

The third and fourth fields are each two bytes and are for future extensions to the formal definition. The present definition requires that both of these fields be zero. The final four byte field is an integer that gives the offset of the start of the pixel Data section relative to the start of the file.

3.3.1.2 The Image Header

There are actually two distinct options for the image Header- one that was developed for the OS/2 BMP format and that is for the Windows BMP format. The OS/2 Image Header is exactly 12 bytes long while the Windows Image Header is at least 40 bytes long. Fortunately, the first four bytes of each format is the length of the Image Header in bytes and therefore a simple examination of this value tells us which format is being used. While the Window Image Header allows headers longer than 40 bytes and this was supported by Window 95, few applications ever adopted it.

3.3.1.3 The Color Table

On dealing with a 24- bit image, no color Table is Present. While dealing with an 8- bit Windows (as opposed to OS/2). BMP image, the Color Table consists of 256 entries with each entry consisting of four bytes of data exists. The first three bytes are the blue, green and red color values respectively. The fourth byte is unused and must be equal to zero. The only difference between the windows and OS/2 version is that the latter does not have this fourth byte.

3.3.1.4 The Pixel Data

In the 8-bit format, each pixel is represented by a single byte of data, that byte is an index into the Color Table. In the 24- bit format, each pixel is represented by three consecutive bytes of data that specify the blue, green and red component values respectively.

Here, different patterns of Gurumukhi characters on paper are drawn and scan the document.

Table 3: Bitmap PinFo Header : BMP Image Header Contents for Windows Format

Field Name	Size In Bytes	Description
BiSize	4	Header size- Must be at least 40
BiWidth	4	Image width in pixels
BiHeight	4	Image height in pixels
Biplanes	2	Must be 1
BiBitCount	2	Bits per pixel – 1,2,4,8,16,24 or 32
BiCompression	4	Compression type(0=uncompressed)
BiSizeImage	4	Image Size- may be zero for uncompressed images
BiXPelsPerMeter	4	Preferred resolution in pixels per meter
BiYPelsPerMeter	4	Preferred resolution in pixels per meter
BiClrUsed	4	Number Color Map entries that are actually used
BiClrImportant	4	Number of significant colors

3.3.2 Problem In Recognition of Characters

Neural network themselves are not invariant to translation and scale change. The images need to be preprocessed first to achieve the property of translation and scaling invariance. For recognition of character images, firstly the image should be preprocessed so that extra noise can be removed. After removing of noise, the character features are extracted to classify the characters. For this recognition scheme, neural network must be trained. The

advantage of using neural network for recognition is that different character styles can be easily recognized. Using algorithms, the particular characters which are available in database can be recognized. This chapter defines the problem statement and describes the implementation of the neural network in the reconstruction of image. Furthermore, neural network also help to find out the accuracy of the reconstructed Gurumukhi characters drawn by different users.

Chapter 4 Implementation

4.1 Recognition of Characters

As defined in previous chapter, the problem of recognition of characters can be solved using neural networks. A scheme is proposed to recognize characters from image. Using neural network, recognition of characters is done in following steps:-

4.1.1 Load Image

Firstly, input digitized image. Further, this image is used to recognize characters. Figure 24 and figure 25 represents the step of loading of image.

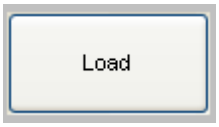

Figure 24: Load Button

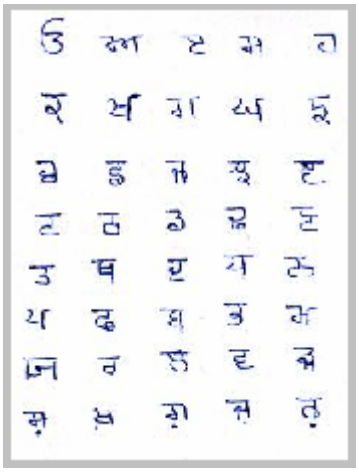

Figure 25: Character Image

When load button (shown in figure 24) 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 25). After loading of image, selection of character is performed.

4.1.2 Selection of Character

After loading image, a particular character is selected for recognition. When a particular character from image is selected, a window represents that character. A separate window is also shown in which bounding box of all characters is created. The advantage of creating bounding box is calculation of area of particular character. In this, there is no limitation of number of characters. Any number of characters can be boxed which are present in image. We take an example of character ऋ, i.e. selection of character ऋ. The following figures represent the selection and bounded boxes of character:-

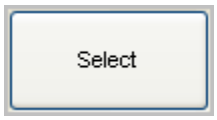

Figure 26: Select Button

Figure 27: Selected Character from Image

When select button (shown in figure 26) is pressed, the selected character is shown in next window. This character is further processed for recognizing that character. As select button is pressed, another window represents the bounding box of all characters. After selection of image, preprocessing is done.

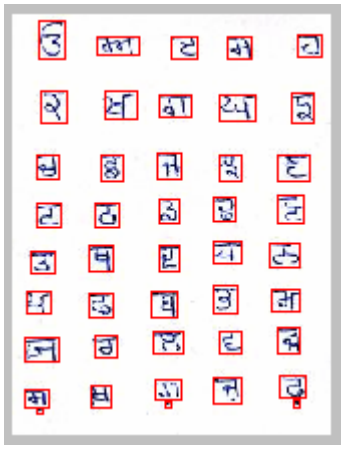

Figure 28: Bounding Box of Characters

4.1.3 Preprocessing

After selection of a particular character, that character is preprocessed. 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 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:-

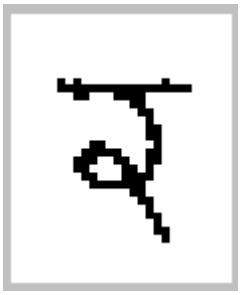

Figure 29: Cropped Character 'ञ'

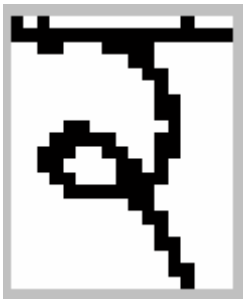

Figure 30: Filled character 'ব'

4.1.4 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 HGCR system is the large variation in shapes within a class of 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 'ব':-

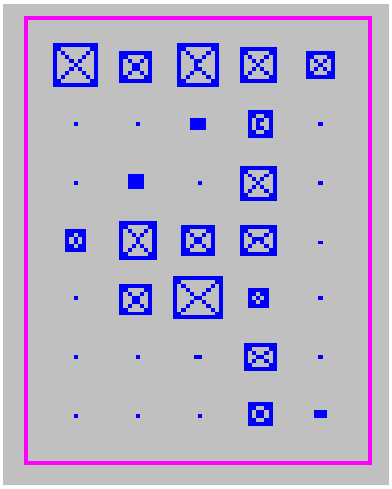

Figure 31: Extracted features of character 'ব'

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 degraded characters of Gurumukhi script due to large shape variations in characters of the same class.

4.2 MATLAB

The name MATLAB stands fo*r matrix laborator*y. 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
- Modeling, 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 learning and applying specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M-files) that extend the MATLAB environment to solve particular classes of problems. It includes among others image processing and neural networks toolboxes.

4.3 Image database

The starting point of the project was the creation of a database with all the character images that would be used for training and testing. The image database can have different formats. Character images are handwritten digitized images. For training of neural networks, characters are written from different people in different handwriting styles and in different fonts. This means that characters on paper have different sizes, different resolutions and some times almost completely different angles. Images belonging to the last case was very few but they were discarded, as there was no chance of classifying them correctly. Two operations were carried out in all of the images. They were converted to grayscale and the background was made uniform. The database itself was constantly changing throughout the completion of the thesis work as it was it that would decide the robustness of the algorithm. Therefore, it had to be done in such way that different situations could be tested and thresholds above which the algorithm didn't classify correct would be decided. The construction of such a database is clearly dependent on the application.

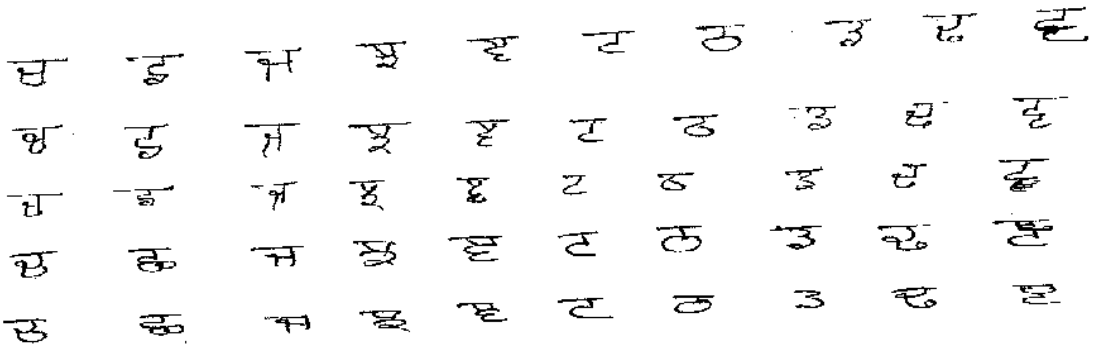

Figure 32: Character image for training and testing

In figure 32, the first four lines are used for training of neural networks. All lines have different writing styles. Last line is used for testing of characters.

The final form of the database is this.

4.4 Perceptron Implementation in MATLAB

4.4.1 Learning Rules

Learning rules as a procedure for modifying the weights and biases of a network are defined. The learning rule is applied to train the network to perform some particular task.

Learning rules in the MATLAB toolbox fall into two broad categories: supervised learning and unsupervised learning. The algorithm is developed using supervised learning. In supervised learning, the learning rule is provided with a set of examples (the training set) of proper network behavior: where is an input to the network, and is the corresponding correct (target) output. As the inputs are applied to the network, the network outputs are compared to the targets. The learning rule is then used to adjust the weights and biases of the network in order to move the network outputs closer to the targets. The Perceptron learning rule falls in this supervised learning category.

4.4.2 Neural Network training Using MATLAB

In MATLAB, training of neural network is done using neural network toolbox. Firstly, a NN Network/data manager screen is opened in this tool.

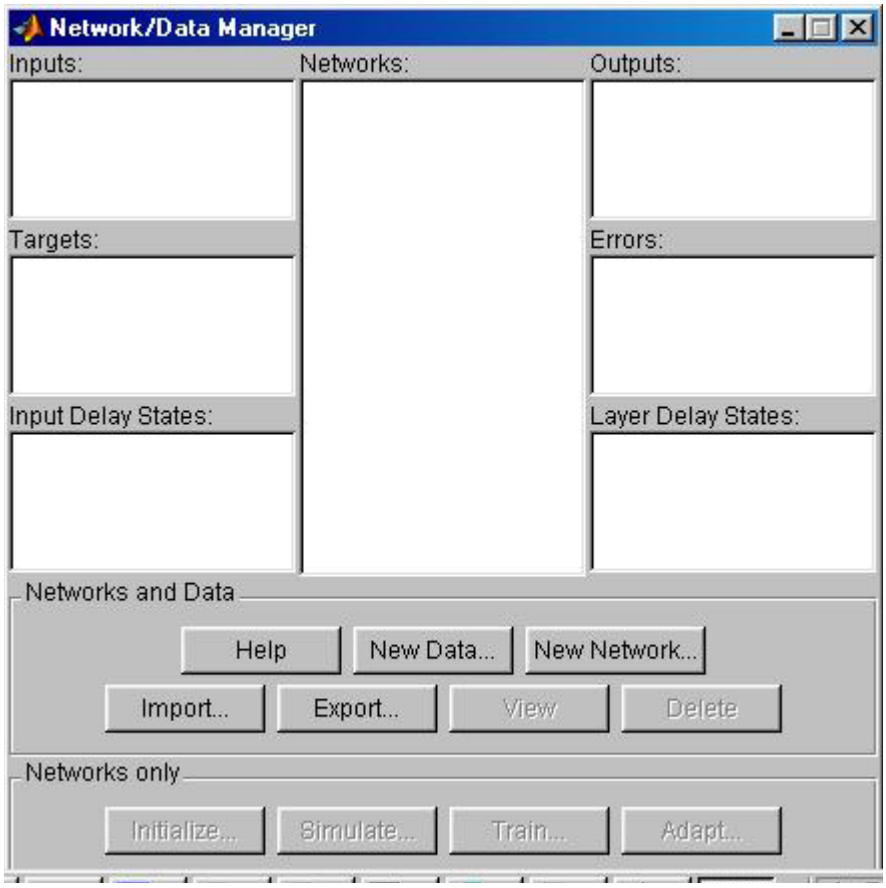

Figure 33: NN Network / Data Manager Screen

Let P denote the input and T denote the target/output. The value of P denotes that character image which has to be recognized and value of T denotes the value of final

character. Each character has its own value of P and T. These values depend on the features of character. In order to use that network firstly the network is designed and then trained it. After this, network is ready for simulations to be performed on it. Following steps are used to design and train a network:-

- **Design Phase**

Step 1: Firstly, the value of P and T is entered to the NN Network Manager. This is done by clicking new data once.

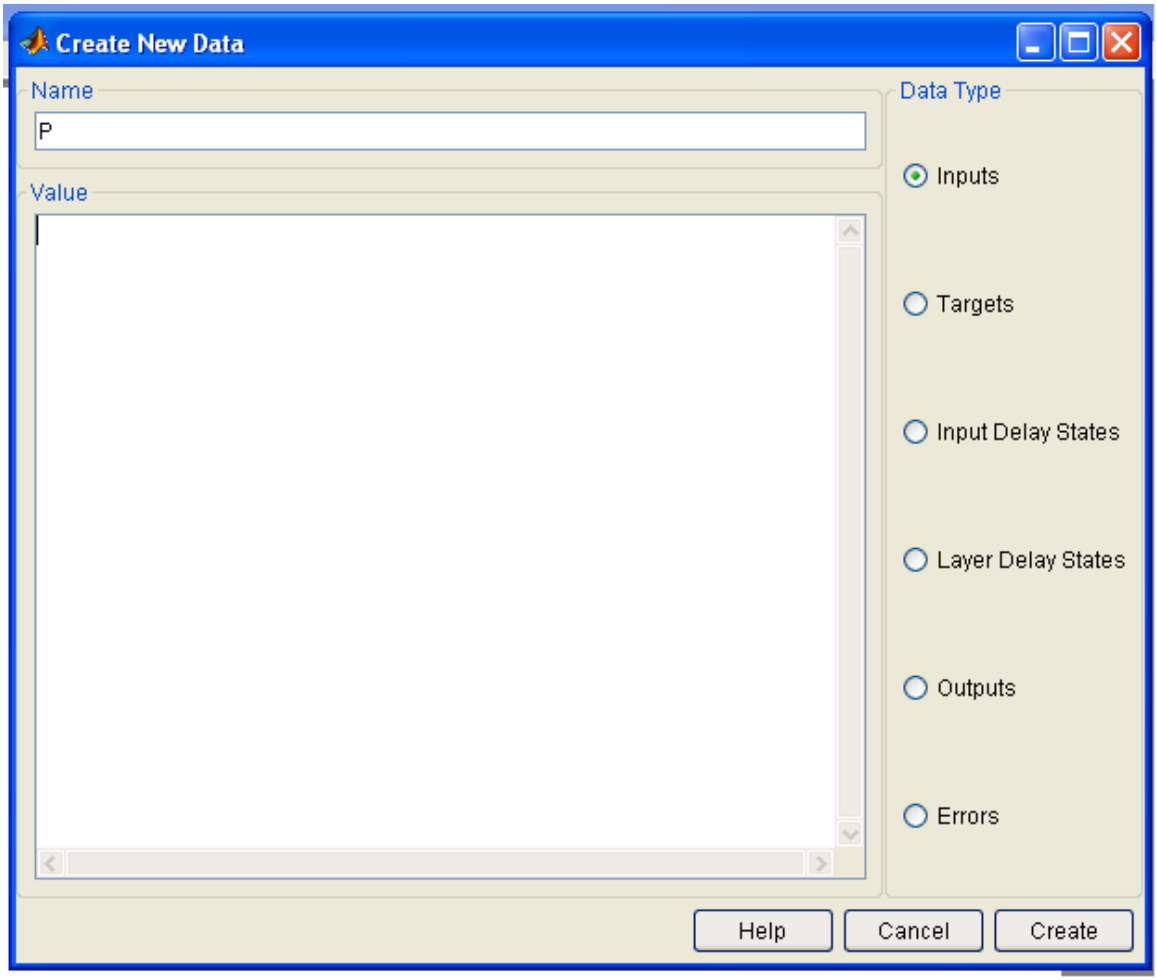

Figure 34: Create New Data

Step 2: Type P as the Name, and corresponding matrix of character as the Value, select Inputs under Data Type, then confirm it by clicking on Create Button.

Step 3: Similarly, type in T as the Name, and corresponding matrix of target character as the Value, select Targets under Data Type, then confirm it by clicking Create Button.

Step 4: Now, a network is created. For this, click on New Network. A screen is opened as shown in figure 36.

Now, all the parameters on the screen are changed according to the values:

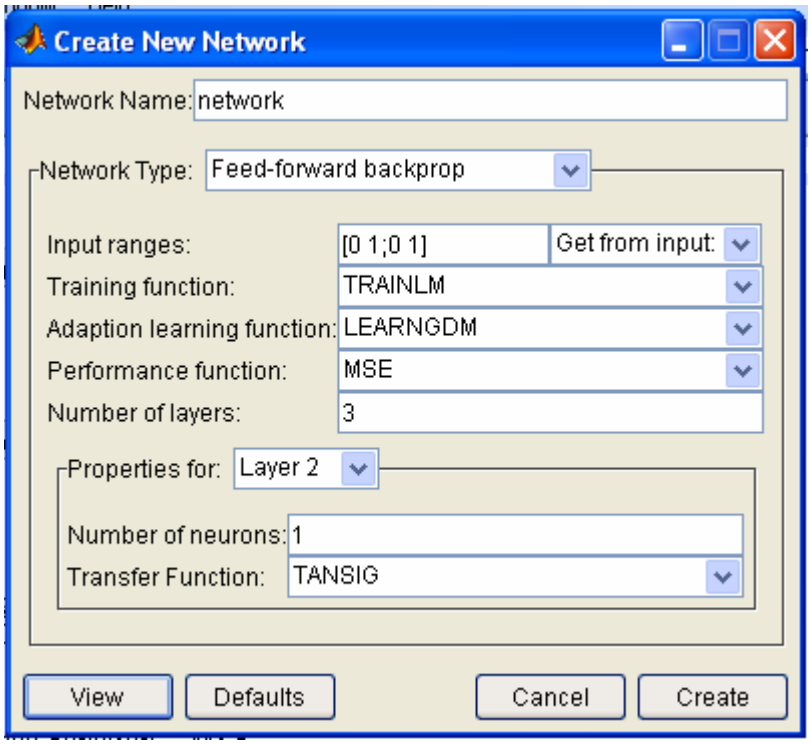

Figure 35: Create New Network

Following parameters are used:

Network Type = Feedforword Backprop

Train Function = TRAINLM

Adaption Learning Function = LEARNGDM

Performance Function = MSE

Numbers of Layers = 3

Step 5: For Layer 1, LOGSIG is selected as Transfer Function and for layer 2; TANSIG is selected as Transfer Function

Step 6: Then, Create button is pressed which concludes the network implementation phase.

- **Network Training:**

Step 7: Now, click on train button.

Step 8: On Training info, P is selected as Inputs, T as Targets.

Following parameters are used for Training:

No. of epochs = 500

Goal = 0.001

max_fail = 50

After specifying all the parameters, Train network is clicked. This gives a training and performance plot.

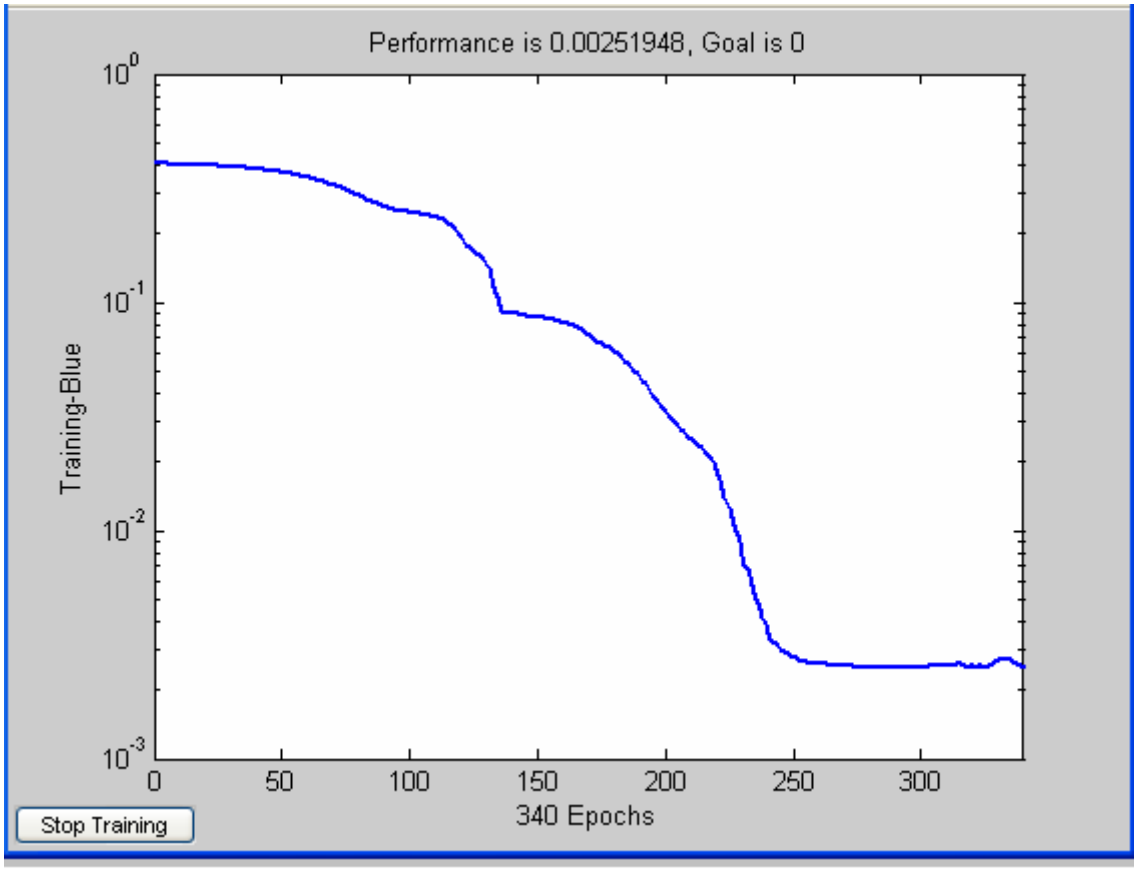

Figure 36: Training of characters using neural networks

4.4.3 Train Set:

Ten training sets of such images, each one containing 40 characters are used for training of neural networks. Each set of character images varies for handwriting styles and fonts. The training of characters is as shown in figure 36.

4.4.4 Test Set:

There is no reason for keeping those on a constant number. Each set of character images contain last line of 10 characters for testing purpose. Beside these character images, other character images can be used for testing.

Chapter 5 Results

The sets of handwritten Gurumukhi characters are made. The data set was partitioned into two parts. The first one is used for training the system and the second one for testing. For each character, features were computed and stored for training the network. Three network layers, i.e. 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, say 0.1, desired results are not obtained, so changing the value of error tolerance i.e. say 0.01, 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. The unit disk is taken for each character by finding the maximum radius of the character (i.e. the maximum distance between the center of the character and the boundary of the character), so that the character could fit on the disk.

Here are some tables displaying the results obtained from the program. Sign images of the same letter are grouped together on every table. The table gives us information about the pre-processing operations that took place (i.e. noise, edge detection, filling of gap) and also if the image belongs to the same database with the training images. The amount of each filter is also recorded so maximum values of noise can be estimated that the network can tolerate. This of course varies from character image to character image. The result also varies for every time the algorithm is executed. The variance is very small but it is there. Following are main results of Gurumukhi character recognition: -

Table 4: Recognition Accuracy of Handwritten Gurumukhi Characters

Character	No. of Samples	Train/Test	% Accuracy
	200	180/20	93%
	196	176/20	87%
	155	130/25	89%
	184	169/15	71%
	192	162/30	69%
	160	140/20	81%
	179	159/20	79%
	168	148/20	84%
	195	170/25	80%
	177	152/25	90%
	191	166/25	88%
	180	165/15	86%
	195	170/25	89%
	187	167/20	96%
	169	149/20	95%
	199	174/25	92%
	188	168/20	94%
	166	146/20	82%
	196	176/20	82%
	189	164/25	88%
	168	148/20	85%
	178	158/20	84%
	196	176/20	87%
	171	151/20	81%
	182	162/20	88%
	184	164/20	80%
	169	149/20	89%
	180	155/25	76%
	170	150/20	78%
	193	173/20	71%
	185	165/20	82%
	176	146/30	70%
	167	147/20	92%
	157	132/25	85%
	178	158/20	87%
	183	153/30	69%
	191	161/30	73%
	185	155/30	70%

Chapter 6 Conclusion and Future Scope

6.1 Conclusion

Handwritten Gurumukhi character recognition using neural networks is discussed here. It has been found that recognition of handwritten Gurumukhi characters is a very difficult task. Following are main reasons for difficulty in recognition of Gurumukhi characters:-

- Some Gurumukhi characters are similar in shape (for example ਹ and ਝ).
- Different, or even the same writer 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 facts are justified by the work done here. A small set of all Gurumukhi 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. The result of the last training by 50 character set and testing with the 10 character set are presented. 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

Over the past three decades, many different methods have been explored by a large number of scientists to recognize characters. A variety of approaches have been proposed

and tested by researchers in different parts of the world, including statistical methods, structural and syntactic methods and neural networks. No OCR in this world is 100% accurate till date. The recognition accuracy of the neural networks proposed here can be further improved. The number of character set used for training is reasonably low and the accuracy of the network can be increased by taking more training character sets. This approach of recognition is used for recognition of Gurumukhi characters only. In future work, this can be implemented for recognition of Gurumukhi words.

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] Negi, C. Bhagvati and B. Krishna, "An OCR system for Telugu", in the Proceedings of the Sixth International Conference on Document Processing, pp.1110-1114, 2001.

[3] Hong, J.I. and Landay, J.A. SATIN: A Toolkit for Informal Ink-based Applications. CHI Letters: ACM Symposium on UIST, 2 (2), 63-72.

[4] S. Mori, C. Y. Suen and K. Yamamoto, "Historical review of OCR research and development", Proceedings of the IEEE, Vol. 80(7), pp. 1029-1058, 1992.

[5] U. Pal and B. B. Chaudhuri, "Indian script character recognition", Pattern Recognition, Vol. 37(9), pp. 1887-1899, 2004.

[6] H. Bunke and P. S. P. Wang, Handbook of Character Recognition and Document Image Analysis, World Scientific Publishing Company, 1997.

[7] Stephen V. Rice, George Nagy and Thomas A. Nartker, Optical Character Recognition: An Illustrated Guide to the Frontier, Kluwer Academic Publications, 1999.

[8] S. Mori, H. Nishida and H. Yamada, Optical Character Recognition, John Wiley & Sons, 1999.

[9] S. Impedovo, L. Ottaviano and S. Occhinegro, "Optical character recognition", International Journal Pattern Recognition and Artificial Intelligence, Vol. 5(1-2), pp. 1-24, 1991.

[10] A. K. Jain, R. P. W. Duin and J. Mao, "Statistical pattern recognition: a review",*IEEE Transactions on PAMI*, Vol. 22(1), pp. 4-37, 2000.

[11] Manish Kumar, "Degraded text recognition of gurmukhi scripts",Dspace, Thapar University, Patiala.

[12] R. Kasturi and L. O'Gorman, "Document image analysis: a bibliography", Machine Vision and Applications, Vol. 5(3), pp. 231-243, 1992.

[13] C. C Tappert, C. Y. Suen and T. Wakahara, "The state of the art in on-line

handwriting recognition", IEEE Transactions on PAMI, Vol. 12(8), pp. 787-808, 1990.

[14] T. Wakahara, H. Murase and K. Odaka, "On-line handwriting recognition", Proceedings of the IEEE, Vol. 80(7), pp. 1181-1194, 1992.

[15] F. Nouboud and R. Plamondon, "On-line recognition handprinted characters: beta tests", Pattern Recognition, Vol. 23(9), pp. 1031-1044, 1990.

[16] C. Y. Suen, M. Berthod and S. Mori, "Automatic recognition of hand printed characters- the state of the art", Proceedings of the IEEE, Vol. 68(4), pp. 469-487, 1980.

[17] S. D. Connell, R. M. K. Sinha and A. K. Jain, "Recognition of unconstrained **o**n-line Devanagari characters", in the Proceedings of 15th International Conference on Pattern Recognition (ICPR), Vol. 2, Spain, pp. 368-371, 2000.

[18] S. D. Connell and A. K. Jain, "Template-based online character recognition", Pattern Recognition, Vol. 34(1), pp. 1-14, 2001.

[19] F. Bortolozzi, A. Britto Jr., L. S.Oliveria and M. Morita, "Recent advances in handwriting recognition", in the Proceedings of International Workshop on Document Analysis (IWDA), India, pp. 1-30, 2005.

[20] S. W. Lee, "Off-line recognition of totally unconstrained handwritten numerals Using multiplayer cluster neural network", IEEE Transactions on PAMI, Vol 18(6), pp.648-652, 1996.

[21] F. El-Khaly and M. A. Sid-Ahmed, "Machine recognition of optically captured machine printed Arabic text", Pattern Recognition, Vol. 23(11), pp. 1207-1214, 1990.

[22] A. Amin, "Off-line Arabic character recognition", in the Proceedings of ICDAR, pp. 596-599, 1997.

[23] L. M. Lorigo and V. Govindaraju, "Offline Arabic handwriting recognition", IEEE Transactions on PAMI, Vol. 28(5), pp. 712-724, 2006.

[24] T. H. Hildebrandt and W. Liu, "Optical recognition of handwritten Chinese characters: Advances since 1980", Pattern Recognition, Vol. 26(2), pp. 205-225, 1993.

[25] C. L. Liu, S. Jaeger and Masaki Nakagawa, "Online recognition of Chinese

characters: the state-of-the-art", IEEE Transactions on PAMI, Vol. 26(2), pp. 198-213, 2004.

[26] I. K. Sethi and B. Chatterjee, "Machine Recognition of constrained Hand-printed Devnagari", Pattern Recognition, Vol. 9, pp. 69-75, 1977.

[27] R.M.K. Sinha, H. Mahabala, Machine recognition of Devnagari script, IEEE Trans. Systems Man Cybern. 9 (1979) 435–441.

[28] K. Sethi, B. Chatterjee, Machine recognition of constrained hand-printed Devnagari, Pattern Recognition 9 (1977) 69–76.

[29] Brijesh k. Verma, "Handwritten Hindi Character recognition Using Multilayer Perceptron and Radial Basis Function Neural Networks," IEEE International conference on Neural Networks,vol. 4,pp. 2111-2115, Nov. 1995.

[30] S.K. Parui, B.B. Chaudhuri, D. Dutta Majumder, A procedure for recognition of connected hand written numerals, Int. J. Systems Sci. 13 (1982) 1019–1029.

[31] A.F.R. Rahman, M. Kaykobad, A complete Bengali OCR: a novel hybridapproach to handwritten Bengali character recognition, J. Comput. Inform. Technol. 6 (1998) 395–413.

[32] U. Pal, B.B. Chaudhuri, Automatic recognition of unconstrainedo8-line Bangla hand- written numerals, in: T. Tan, Y. Shi, W. Gao (Eds.), Advances in Multimodal Interfaces, Springer Verlag Lecture Notes on Computer Science (LNCS-1948), 2000, pp. 371–378.

[33] U. Bhattacharya, T.K. Das, A. Datta, S.K. Parui, B.B. Chaudhuri, A hybrid scheme for hand printed numeral recognition based on a self-organizing network and MLP classifiers, Int. J. Pattern Recognition Artificial Intelligence. 16 (2002) 845–864.

[34] A.K. Dutta, S. Chaudhuri, Bengali alpha-numeric character recognition using curvature features, Pattern Recognition 26(1993) 1757–1770.

[35] Shamik Sural, P.K. Das, An MLP using Hough transform based fuzzy feature extraction for Bengali script recognition, Pattern Recognition Lett. 20 (1999) 771–782.

[36] U. Garain, B.B. Chaudhuri, T.T. Pal, Online handwritten Indian script recognition: a human motor function based framework, in: Proceedings of the 16th International

Conference on Pattern Recognition, Vol. 3, 2002, pp.164–167.

[37] U. Pal, Wakabayashi and F. Kimura "Handwritten Bangla Compound character recognition using Gradient feature, "ICIT 2007,10th international conference on information technology, pp. 208-213, Dec. 2007.

[38] G. Siromoney, R. Chandrasekaran and M. Chandrasekaran, "Machine recognition of printed Tamil characters", Pattern Recognition, Vol. 10, pp. 243-247, 1978.

[39] M. Chandrasekaran, R. Chandrasekaran and G. Siromoney, "Context dependent recognition of handprinted Tamil characters", in the Proceedings of International Conference on Systems Man and Cybernetics, Vol. 2, pp. 786-790, 1983.

[40] P. Chinnuswamy and S. G. Krishnamoorthy, "Recognition of hand-printed Tamil characters", *Pattern Recognition*, Vol. 12(3), pp. 141-152, 1980.

[41] K. H. Aparna, V. Subramaniam, M. Kasirajan, G. V. Prakash, V. S. Chakravarthy and S. Madhvanath, "Online handwrting recognition for Tamil", in the Proceedings of 9th International Workshop on Frontiers in Handwriting Recognition(IWFHR), pp. 438-443, 2004.

[42] S. N. S Rajasekaran and B. L. Deekshatulu, "Recognition of printed Telugu characters", Computer Graphics and Image Processing (CGIP), Vol. 6, pp. 335-360, 1977.

[43] C. V. Lakshmi and C. Patvardhan, "A high accuracy OCR system for printed Telugu text", in the Proceedings of Conference on Convergent Technologies for Asia-Pacific Region (TENCON 2003), Vol. 2, pp. 725-729, 2003.

[44] D. Bersekas and R. Gallager, Data Networks, Prentice-Hall, 1987.

Papers Published

1.Naveen Garg, Divya Sharma, Karun Verma, “Analysis of Recognition of handwritten Indian script”, International IEEE Conference on Innovative Technology, in PDM college of engineering Bahadurgarh Haryana, pp.219, June 18-19,2009

2. Naveen Garg, Divya Sharma, Karun Verma, “Handwritten Gurumukhi Character Recognition using multilayer Perceptron neural networks”, National Conference on advances in computer networks & information technology GJUS&T Hisar, pp.263-265 March 24-25, 2009.