Handwritten Character Recognition of a Vernacular Language: The Odia Script

Ramesh Kumar Mohapatra

Department of Computer Science and Engineering
National Institute of Technology Rourkela

Handwritten Character Recognition of a Vernacular Language: The Odia Script

Dissertation submitted in partial fulfillment

of the requirements of the degree of

Doctor of Philosophy

in

Computer Science and Engineering

by

Ramesh Kumar Mohapatra

(Roll Number: 511CS402)

based on research carried out

under the supervision of

Prof. Banshidhar Majhi

and

Prof. Sanjay Kumar Jena

November, 2016

Department of Computer Science and Engineering

National Institute of Technology Rourkela

Department of Computer Science and Engineering
National Institute of Technology Rourkela

November 28, 2016

Certificate of Examination

Roll Number: *511CS402*
Name: *Ramesh Kumar Mohapatra*
Title of Dissertation: *Handwritten Character Recognition of a Vernacular Language: The Odia Script*

We the below signed, after checking the dissertation mentioned above and the official record book (s) of the student, hereby state our approval of the dissertation submitted in partial fulfillment of the requirements of the degree of Doctor of Philosophy in Department of Computer Science and Engineeringat National Institute of Technology Rourkela. We are satisfied with the volume, quality, correctness, and originality of the work.

Sanjay Kumar Jena
Co-Supervisor

Banshidhar Majhi
Principal Supervisor

Pankaj Kumar Sa
Member, DSC

Bidyadhar Subudhi
Member, DSC

Sarat Kumar Patra
Member, DSC

External Examiner

Santanu Kumar Rath
Chairperson, DSC

Durga Prasad Mohapatra
Head of the Department

Department of Computer Science and Engineering
National Institute of Technology Rourkela

Prof. Banshidhar Majhi
Professor

Prof. Sanjay Kumar Jena
Professor

November 28, 2016

Supervisors' Certificate

This is to certify that the work presented in the dissertation entitled *Handwritten Character Recognition of a Vernacular Language: The Odia Script* submitted by *Ramesh Kumar Mohapatra*, Roll Number 511CS402, is a record of original research carried out by him under our supervision and guidance in partial fulfillment of the requirements of the degree of *Doctor of Philosophy* in *Department of Computer Science and Engineering*. Neither this dissertation nor any part of it has been submitted earlier for any degree or diploma to any institute or university in India or abroad.

Sanjay Kumar Jena	Banshidhar Majhi
Professor	Professor

dedicated to my parents and my beloved wife...

Declaration of Originality

I, *Ramesh Kumar Mohapatra*, Roll Number *511CS402* hereby declare that this dissertation entitled *Handwritten Character Recognition of a Vernacular Language: The Odia Script* presents my original work carried out as a doctoral student of NIT Rourkela and, to the best of my knowledge, contains no material previously published or written by another person, nor any material presented by me for the award of any degree or diploma of NIT Rourkela or any other institution. Any contribution made to this research by others, with whom I have worked at NIT Rourkela or elsewhere, is explicitly acknowledged in the dissertation. Works of other authors cited in this dissertation have been duly acknowledged under the sections "Reference" or "Bibliography". I have also submitted my original research records to the scrutiny committee for evaluation of my dissertation.

I am fully aware that in case of any non-compliance detected in future, the Senate of NIT Rourkela may withdraw the degree awarded to me on the basis of the present dissertation.

November 28, 2016
NIT Rourkela

Ramesh Kumar Mohapatra

Acknowledgment

"*Gratitude needs the honesty to acknowledge that beyond me a many and beyond them is the Lord*."

It gives me immense pleasure to thank publicly the individuals who have been compelling amidst my studies at National Institute of Technology Rourkela, India. I am particularly grateful to God, who gave me this lovely life and consistently showers me with blessings. I am significantly obligated to my supervisor, Dr. Banshidhar Majhi and co-supervisor, Dr. Sanjay Kumar Jena, on account of their calmness, endless support, and profound experiences. I would like to express my gratefulness to my supervisors, for giving me the chance to work on this topic and for their enduring and reassurance all through years of my studies. I feel particularly obligated to the accordant people of the Image Processing and Computer Vision Laboratory (IPCV Lab) for the spirit of coordinated effort being instrumental in the completion my thesis. Special thank goes to Ratnakar Dash for perusing my thesis. Words fail me to express my gratitude to my adored parents who yielded their solace for my advancement. I am perpetually obliged to my parents for their understanding and comprehension, alleviating my family responsibilities and encouraging me to focus on my study.

At last and above all, I would like to express special thanks to my dear spouse *Sarita* for her unconditional love and support when it was most required. Without her assistance and consolation, this study would not have been finished.

November 28, 2016 *Ramesh Kumar Mohapatra*

NIT Rourkela Roll Number: 511CS402

List of Abbreviations

ANN	Artificial Neural Network
BPNN	Back Propagation Neural Network
CCHM	Chain Code Histogram Matrix
CCHV	Chain Code Histogram Vector
CD	Contrastive Divergence
C-DAC	Centre for Development of Advanced Computing
CEDAR	Center of Excellence for Document Analysis and Recognition
DBN	Deep Believe Network
DIA	Document Image Analysis
DOST	Discrete Orthogonal Stockwell Transform
FNR	False Negative Rate
FPR	False Positive Rate
GUI	Graphical User Interface
HMM	Hidden Markov Model
ICA	Independent Component Analysis
kNN	k- Nearest Neighbor Classifier
KLT	Karhunen–Loève transform
LQD	Linear Quadratic Classifier
LSVM	Linear Support Vector Machine
MATLAB	Matrix Laboratory
MLP	Multi Layer Perceptron
MNIST	Modified National Institute of Standards and Technology Database
OCR	Optical Character Recognition
ODDB	Odia Digit Database
OHCS	Odia Handwritten Character Set
OHCS v1.0	Odia Handwritten Character Set version 1.0
PCA	Principal Component Analysis
RAM	Random Access Memory
RBM	Restricted Boltzmann Machine
ROC	Receiver Operating Characteristic
ST	Stockwell Transform
SVM	Support Vector Machine
TNR	True Negative Rate
TPR	True Positive Rate
USPS	United State Postal Service Database

Abstract

"*A person doesn't really understand something until after teaching it to a computer.*"

–D.E. Knuth

Optical Character Recognition, i.e., OCR taking into account the principle of applying electronic or mechanical translation of images from printed, manually written or typewritten sources to editable version. As of late, OCR technology has been utilized in most of the industries for better management of various documents. OCR helps to edit the text, allow us to search for a word or phrase, and store it more compactly in the computer memory for future use and moreover, it can be processed by other applications. In India, a couple of organizations have designed OCR for some mainstream *Indic* dialects, for example, Devanagari, Hindi, Bangla and to some extent Telugu, Tamil, Gurmukhi, Odia, etc. However, it has been observed that the progress for Odia script recognition is quite less when contrasted with different dialects. Any recognition process works on some nearby standard databases. Till now, no such standard database available in the literature for Odia script. Apart from the existing standard databases for other *Indic* languages, in this thesis, we have designed databases on handwritten Odia Digit, and character for the simulation of the proposed schemes. In this thesis, four schemes have been suggested, one for the recognition of Odia digit and other three for atomic Odia character. Various issues of handwritten character recognition have been examined including feature extraction, the grouping of samples based on some characteristics, and designing classifiers. Also, different features such as statistical as well as structural of a character have been studied. It is not necessary that the character written by a person next time would always be of same shape and stroke. Hence, variability in the personal writing of different individual makes the character recognition quite challenging. Standard classifiers have been utilized for the recognition of Odia character set.

An array of Gabor filters has been employed for recognition of Odia digits. In this regard, each image is divided into four blocks of equal size. Gabor filters with various scales and orientations have been applied to these sub-images keeping other filter parameters constant. The average energy is computed for each transformed image to obtain a feature vector for each digit. Further, a Back Propagation Neural Network (BPNN) has been employed to classify the samples taking the feature vector as input. In addition, the proposed scheme has also been tested on standard digit databases like MNIST and USPS. Toward the end of this part, an application has been intended to evaluate simple arithmetic equation.

A multi-resolution scheme has been suggested to extract features from Odia atomic character and recognize them using the back propagation neural network. It has been observed that few Odia characters have a vertical line present toward the end. It helps in dividing the whole dataset into two subgroups, in particular, *Group I* and *Group II* such that all characters in *Group I* have a vertical line and rest are in *Group II*. The two class classification problem has been tackled by a single layer perceptron. Besides, the two-dimensional Discrete Orthogonal S-Transform (DOST) coefficients are extracted from images of each group, subsequently, Principal Component Analysis (PCA) has been applied to find significant features. For each group, a separate BPNN classifier is utilized to recognize the character.

The proposed HOCR-SF scheme works in two phases. In the first phase, the overall Odia character set has been classified into two groups using a Support Vector Machine (SVM) classifier. To accomplish this the sample character is represented as a vector consisting of the number of pixels in each column of the image. The mean value of the lower half and max of the upper half together represents a feature point of the character and used as input to the classifier. In the second phase, the structural features of the character of each group are extracted and fed to a BPNN for recognition. Separate BPNN networks have been designed for classifying the characters in each group.

A semi-supervised learning method using Deep Belief Network (DBN) has been proposed. DBN uses an approximation algorithm namely Contrastive Divergence (CD) to optimize the network parameters. The proposed DBN structure has three hidden layers excluding the input and output layers. The DBN works on an unlabeled dataset. Though the accuracy is not at par with other proposed schemes the advantage is that it requires no prior knowledge about the label of the sample.

Extensive simulations have been carried out in the MATLAB environment to validate the proposed schemes along with other existing schemes. The performance parameters like recognition accuracy, feature length are measured and compared with other techniques. It is observed that the suggested schemes perform better than the state of the art methods.

Keywords: ***Odia Script, Database, Feature Extraction, Chain Code Histogram, Support Vector Machine, Back Propagation Neural Network, Deep Belief Network***.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The human visual system alongside neural structure empowers a man to classify and perceive the objects. It processes the signals and sends it to the most complex human brain for further processing and analyzing the objects. A typical human brain system with 86 billion neurons, is so complex and robust that it is quite impossible to have a replica of that system. Like human visual system, Document Image Analysis (DIA), is the procedure that plays out the overall interpretation of document images [1]. The concept of Optical Character Recognition, popularly known as **OCR** is based on the principle of applying electronic or mechanical translation of images from printed, handwritten, or typewritten sources to editable version. Lately, OCR innovation has been utilized all through the industries helping in the process of document management and has enabled the scanned documents to be viewed more than just image files. It transforms these files into completely searchable text files with the content that is needed by a PC to process and store. The OCR [2–4], has been the subject of intensive research for more than four decades. Specifically, it comes under the category of *pattern recognition*. It is a branch of machine learning [5, 6] that focuses on the recognition of patterns and regularities in data. It is broadly used to put in books and documents into electronic files. For instance, consider a sample paragraph which has been taken from an Odia book and shown in Figure 1.1(a). It's handwritten equivalent text shown in Figure 1.1(b). Figure 1.1(c) shows the expected editable version of the above two samples. OCR is capable of altering the content as well as permit us to hunt down a word and store it more succinctly in the computer memory for future use. These conventional sources are transformed into a machine readable and editable format (usually in text form), to allow its processing by other applications. The OCR handwriting recognition technology formulated from this concept forms the backbone for many new software developments and hardware innovations. The computer industry subjected to various OCR handwriting recognition deployments, in fields ranging from academic to intelligent solution systems. The extensive usage of the OCR handwriting recognition technology has also created an upswing in the manufacturing of various types of scanning devices for every purpose imaginable. The primary process for the OCR handwritten recognition concept is that the scanning device will extract recognizable information from the source document, and subject it to software processing to come up with the digitized file [7].

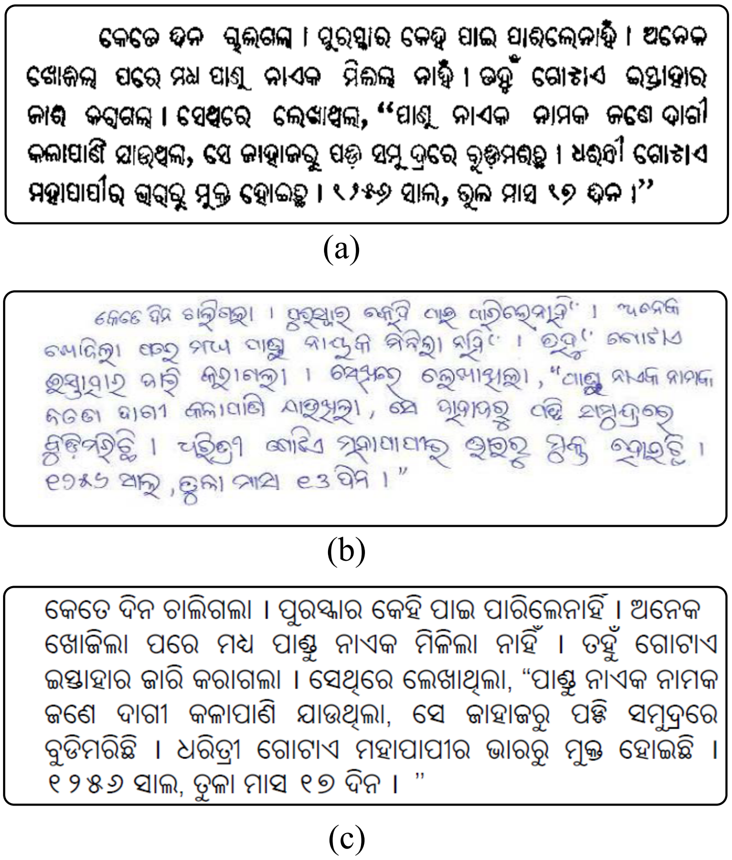

Figure 1.1: Sample paragraph from an Odia book in (a), along with the handwritten text of the same sample in (b). In (c) the expected output of an OCR.

1.1 Phases of OCR and its Applications

Despite the difficulties in analyzing any scanned or digitized document, the overall standard steps followed in an OCR system is shown in Figure 1.2. It has numerous applications in real time e-processing of data [8]. Few applications include bank cheque processing, postal mail sorting, automatic address reading, and much more. In each of the applications, the objective is to extract information about the text being imaged. Depending on the nature of the application, handwritten document image processing can be classified into the following subareas [9].

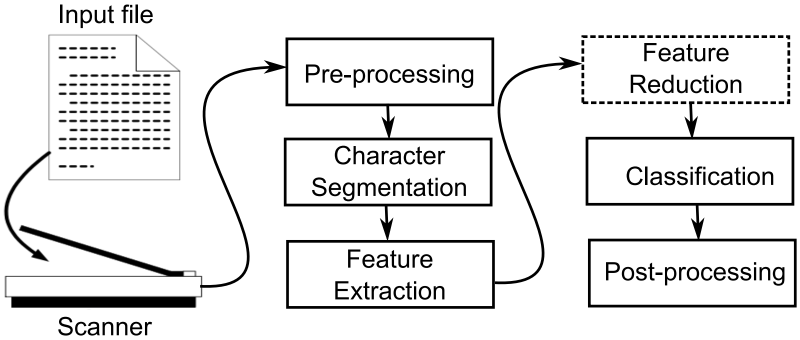

Figure 1.2: Basic steps in any OCR system.

- **Pre-processing** deals with the techniques which are used to process the scanned or captured images to a manageable form so that with less effort it is easy to extract the discriminable features and proceed for classification. It improves the chances of successful recognition. Nowadays with the advancement of technology we are having intelligent scanners, good cameras and imported hand held devices. In these systems, the noise is less likely to be associated with the captured image. However, preprocessing is required for slant and/or skew correction [10], removal of noise, character segmentation [11], etc. During data acquisition, it may so happen that the document may be tilted. The moisture also affects the image quality. Thus, it is necessary to pre-process the raw image to obtain a high quality image.

- **Segmentation** is the process of parceling a digital image into multiple segments. More incisively, image segmentation [12] is the way toward assigning a label to every pixel of that image such that pixels with the same label share certain essential characteristics. To go for character level segmentation [13, 14] first of all we need to find the lines from the whole documents, break each line into words, and further, disunite each word into characters. As far as printed paper is concerned, this process is managed with less exertion whereas for any handwritten document this phase is subtle task for any researcher and he/she must crusade to achieve the goal.

- **Feature Extraction** means to understand the image and pull out the discriminable features [15, 16] from the image. These are effective in recognizing the characters and the most important phase in OCR.

- **Feature Reduction** is an optional step in the process of recognition. Sometimes the amount of feature extracted is huge, and they need to be reduced to an optimal feature set. There may be some redundant information which may not improve the accuracy of the system. Dimensionality reduction algorithms contribute to reducing the classification time and sometimes the miss-classification rate of a classifier [17]. The optimality of a classifier can be enhanced further by the use of dimension reduction techniques. In machine learning, dimensionality reduction techniques can be divided into transform methods (linear and non-linear) and feature (subset) selection methods. Transform methods include Principal Component Analysis (PCA) (also called Karhunen–Loève transform (KLT)), Independent Component Analysis (ICA) (linear) and self-organizing map (SOM, also known as Kohonen map), Locally-Linear Embedding (LLE), and some manifold methods [18]. In some techniques, the goal is to preserve fidelity on the original data using a certain metric like a mean squared error, and in some cases, the goal is to improve the performance of the classifier.

- **Classification** in general refers to the process of assigning a sample to a predefined class. Classification techniques broadly classified into three categories namely, supervised, semi-supervised, and unsupervised learning method [19]. It is the second most challenging phase in any pattern recognition application after feature extraction. Usually in supervised learning we used to predict the class of a new sample by some already computed information. Semi-supervised learning is a type of supervised learning techniques where for modeling the network a small amount of labeled data are used, and many unlabeled data samples processed for prediction. In unsupervised learning, the system has no prior knowledge of the sample for prediction. Conventional classifiers include —

- ☐ Support vector machines (SVM) [20]
- ☐ Neural network [21]
- ☐ Naive Bayes classifier [22]
- ☐ Decision tree [23]
- ☐ Discriminant analysis [24]
- ☐ *k*-Nearest Neighbor (*k*-NN) [25] and many more.

- ■ **Post-processing** in character recognition refers to final error correction in recognization results. That could be either dictionary-based which is also known as lexical error correction or context-based error correction. Dictionary-based relates to detecting and correct misspelled words whereas context-based error correction refers to correcting errors based on their grammatical occurrence in the sentence.

In this thesis, the emphasis is given to investigate some notable features of each character in the Odia language and recognize them utilizing some standard classifier.

1.2 Taxonomy of Character Recognition

In earlier days, documents were made from fallible materials that often fade, rip or degrade over time. So, we should secure those documents for the use of ourselves and our posterity. Possibly, one way is to convert them into a digitized form and then enhance the text in the document to improve its readability. Protecting artifacts against degradation is one of the major challenges in the process of digitization. Many methods proposed in the literature for the warehousing of magazines, historical documents, newspaper, books and so on. But, most organizations come across documents like forms and checks which are hand printed. The classification of character recognition is shown below in Figure 1.3. It is additionally worth seeing that OCR manages off-line recognition while penmanship recognition is for both on-line and off-line samples. On-line implies information caught as composed whereas in off-line the information gathered before processing starts. Off-line processing utilizes just depictions of the handwritten documents without time information.

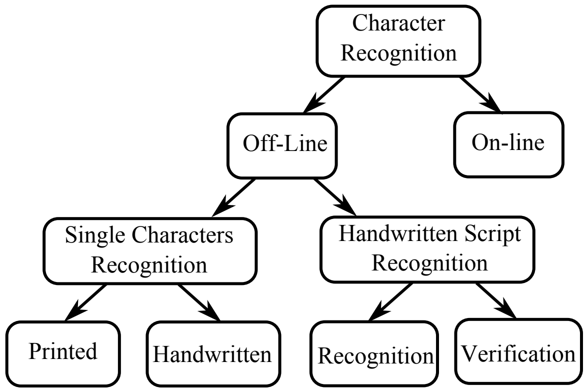

Figure 1.3: The different areas of character recognition.

In the case of off-line, it is indeed very difficult to extract information about the order of strokes that the user has used to compose the character. For the most part, the contribution of on-line penmanship comprises of traces while the off-line handwriting recognition deals with images. Handwritten character recognition is a challenging issue because of the significant amount of variations in the dialects of various languages available across different nations. Typical OCR engines which recognize the printed text fail to identify handwritten documents since the handwriting varies from person to person. Building up OCR not only improves the readability of documents but also makes them editable. The field of handwritten character recognition [26] is the longest established branch of research, and therefore also the aspect which has been studied in most depth. It has enormous applications in various fields such as bank, office, academic, health center, library and apart from these; there are plenty of applications in our day-to-day life. Here, it has been observed that more challenge lies in processing the handwritten document as compared to machine printed. Various commercial and open source OCR systems are available for most primary dialects, including English, Chinese, Japanese, and Korean characters. From the wide assortment of OCR software, the accompanying a portion of the leading solutions which provide higher precision, more speed, and proper page layout reconstruction for the English language. These softwares provide an accuracy rate of up to 99%. However, they maintain the accuracy rate as long as the handwritten source documents are in good condition. When the source documents are of degraded quality, there can be no control over the accuracy. A lot of work is continuing to overcome these limitations. As far as OCR system for Odia (formerly Oriya) language is concerned, Centre for Development of Advanced Computing (C-DAC) in India has developed an Oriya OCR that provides facility to convert text from the scanned image of machine-printed Oriya script. Till now, there is no publicly available OCR system for handwritten Odia language because of its variation in writing style as well as the complexity of the character. In this thesis, we have investigated various features of the Odia handwritten character set off-line and used the standard classifiers to recognize them.

1.3 Related Works

Despite the development of electronic documents and predictions of a paperless world, the importance of handwritten documents has retained its place, and the problems of recognizing character have been an active area of research over the past few years. A wide variety of schemes based on the use of pattern recognition and image processing techniques have been proposed to take care of the issues experienced in automatic analysis and recognition of handwriting characters from a scanned document. A comprehensive survey of the work in optical character recognition for both on-line and off-line until 2000 presented in [28], and [29]. An exhaustive overview of the work in character for *Indic* languages until 2004 reported in [30, 31]. In this regard, various methodologies proposed in the most recent couple of years have been discussed here, thanks to the renewed interest of the document analysis community for this domain. The techniques for handwriting classification are traditionally designed for machine written text and hand printed text. As far as computer printed text analysis is concerned, the only challenge is the extant manuscripts and degraded historical documents. The accuracy of any such system can only be improved by applying some standard and advance preprocessing techniques to transfer the material into a manageable form. The real challenge lies in analyzing a handwritten document due to the various style of writing of individuals. In fact, it is thought-provoking to find the significant feature of a character and identify it accurately.

In the following section, a brief review of works that relate to the recognition of different dialects including *English, Devanagari (Hindi), Arabic, Bangla, Telugu, Tamil, Assamese, Urdu, Kannada, Gurmukhi, Malayalam, Gujarati*, and *Odia* have been presented.

Review of Printed Digit and Character Recognition

Evidently, printed document analysis seems to be much easier then handwritten documents because of the standard representation of each character and moreover very limited fonts available with a little variation in shape and size for a particular character. Usually, the difficulty arises due to the quality of paper used and the noise associated with the scanned copy. In these cases, it is often necessary to preprocess the matter of text into a manageable form so that the recognition process would be more robust in this regard. Many such techniques on printed character and digit classification have been suggested earlier in several languages. Researchers have been tried out for the classification of symbols using statistical and structural features. Many articles are available on the web in the context of identifying printed digits and characters. In the year 1990, Akiyama et al. [32] suggested a system for automatically reading either Japanese or English documents that have complex layout structures that include graphics. Recognition experiments with a model framework for an assortment of complex printed archives demonstrate that their proposed framework is fit for perusing distinctive sorts of printed records at an accuracy rate of 94.8–97.2%. Pal et al. [34]

in the year 2001 identified a problem where both handwritten and machine-printed texts in a single document are present. Looking into the existence of classifiers which can classify either machine printed or hand-printed symbols, they have suggested a scheme, to separate these two types and deal with different classifiers. The structural and statistical features considered for both machine-printed and handwritten text lines. They have experimented on the most popular Indian languages, i.e., Devanagari and Bangla and accuracy of 98.6% has been obtained.

A good number of articles have been published in other *Indic* languages like Gurmukhi [35], Kannada [36], Gujarati [37], Telugu [38], and Tamil [39]. As far as printed Odia digit recognition is concerned, there are very few articles available for the research study. Chaudhuri et al. [40] in the year 2002 have suggested a scheme for printed Odia script recognition with an accuracy of 96.3% considering the stroke features, along with water reservoir feature given in [41].

Review of Handwritten Digit Recognition for *Non-Indic* and *Indic* Languages

Development of handwritten digit recognition has simplified the process of extracting data from the handwritten documents and storing it in electronic formats. These systems are very much useful and necessary in the sectors such as banking, health care industries and many such organizations where handwritten documents used regularly. Available schemes in this regard usually validated on traditional databases such as MNIST, USPS, and CEDAR, which are available publicly for research purpose. More detail description of these databases given in the following chapter. Many researchers have contributed towards the recognition of handwritten English digit. Yann LeCun and his co-researchers have published many articles on the recognition of sample numeral from MNIST database, and these are available at his homepage [42]. Due to the inaccessibility of standard databases on *Indic* vernaculars more often than not, people have designed their database to validate their suggested schemes. Looking into the popularity of languages and more importantly, the resemblance in shape and size of other *Indic* scripts with the Odia language it has been observed that Bangla script is very close to Odia language. Plenty of work has been carried out for Bangla digit recognition which is recorded in Table 1.1, utilizing different methods.

Table 1.1: Comparison of several schemes for the recognition of Bangla digit.

Author(s)	Feature	Classifier	Accuracy in %
Xu et al. (2008) [43]	Whole Image	Hierarchical Bayesian network	87.50
Khan et al. (2014) [44]	Statistical	Sparse Representation	94.00
Basu et al. (2005) [45]	Structural feature	Dempster-Shafer	95.10
Basu et al. (2012) [46]	Shape primitives	MLP	96.67
Hassan et al. (2015) [47]	Local Binary patterns	k-Nearest Neighbors (k-NN)	96.70
Das et al. (2014) [48]	Convex Hull	MLP	99.48

Review on Odia Digit

In the year 2005, Roy et al. [49] have proposed a scheme that segments the image into several blocks and the chain code histogram on the contour of each digit image extracted as the feature. Multiple classifiers such as neural network and quadratic used separately on a database of 3850 samples and achieved the maximum accuracy of 94.8 percentage. Bhowmik et al. [50] have suggested a scheme for Odia handwritten digit and obtained an overall accuracy of 93%. The method utilizes the underlying representation of each numeral as the feature and Hidden Markov Model (HMM) as a classifier. Recently, there are many developments have been made in this area. In 2012, Sarangi et al. [51] have proposed a Hop-field Neural Network (HNN) classifier to identify the Odia digit. The experiment has been carried out on 290 test patterns (29 samples for each digit) with image size 12×12. With this minuscule set of samples, the HNN network recognized 284 samples correctly out of 290. Panda et al. [52] in the year 2014 have suggested Odia handwritten digit recognition scheme using a single layer perceptron. Their scheme uses gradient and curvature feature for classification. An accuracy of 85% has been recorded when the method applied on a database of 100 patterns for each digit collected from 100 peoples. Two years back, Dash et al. [53] proposed a hybrid feature based Odia handwritten digit classification. In this, along with the curvature feature, they have extracted features using Kirsch gradient operator. After making the dimensional reduction of the feature matrix through PCA, Quadratic Discriminant Function (QDF) classifier has been applied to recognized the digit with an error rate of 1.5%. Mishra et al. [54] suggested a fuzzy aggregated feature along with HMM classifier on a dataset of 2500 handwritten samples and a recognition accuracy of 96.3% recorded. Dash et al. [53, 56] have proposed two methods for the recognition of handwritten Odia digits and achieved an overall accuracy of 98.80%. In their first attempt, they have utilized the hybrid feature along with the discriminant classifier for the classification of handwritten Odia digit. Secondly, a non-redundant multi-resolution feature scheme along with many classifiers such as k-Nearest Neighbor (k-NN), Support Vector Machine (SVM), Multi-Layer Perceptron (MLP), and Modified Quadratic Discriminant Function (MQDF) classifier. Pujari et al. [55] have presented a comparative analysis of different techniques.

Review of Handwritten Character Recognition for *Non-indic* and *Indic* Languages

There are many commercially available OCR systems for the recognition of languages spoken all over the world. A decent number of techniques are also available with high recognition rate for various dialects like *Latin* [57], *Chinese* [58, 59], *English*, *Arabic* [60–64], *Japanese* [65] and many more. The English script being the most widely spoken language, we compendious few works that are related to English alphabet recognition. Evidently, there are 52 total characters in English text including lower and upper case alphabet. Few special symbols usually used in English documents such as full-stop ('.'), comma (','), left and right parenthesis, left and right square brackets, single quote, double quotes, backward slash, forward slash, '@', '&', and some other symbols. A significant amount of work published in this area, however, some recent developments have been cited here which is just a blue line in the sky. Eugene Borovikov [66] presented a survey paper on modern optical character recognition techniques in the year 2014. Due to the encroachment and the progress in the information technology, nowadays more emphasis is given to vernaculars in India. In spite of the fact that there are twenty-two official dialects in India, a limited survey has been done for Odia language and other languages similar to Odia. Out of all vernaculars, the most prevalent dialects are Devanagari (Hindi), Bangla, Telugu, Tamil, Gujarati, Assamese, and Odia. In the Indian OCR context, among these languages mostly Bangla and Devanagari have enormous regard for the analyst from the most recent two decades. Unlike, Bangla and Devanagari other Indic languages such as Odia (Formerly "Oriya"), Telugu, and Tamil haven't got the same.

Devanagari

First research report on handwritten Devanagari characters published in the year 1977 by I. K. Sethi and B. Chatterjee [67]. The gradient representation of each character considered as the basis for extraction of features. A decision tree has been applied to arrive at a final choice regarding the class membership label assigned to the input character. In the year 1979, Sinha and Mahabala [68] have purposed a syntactic pattern analysis system for the recognition of Devanagari script. The proposed system also considered the structural descriptions of each character where each input word is digitized and labeled utilizing a local feature extraction process. Template matching algorithm has been used to recognize the character based on the stored description values. Though the experiment gives 90 percent correct recognition of characters, the system fails to recognize similar shape characters present in the script. In 2000, Connell et al. [69] have identified the problem of unconstrained on-line Devanagari character recognition. They have conducted the experiment on a database where the samples collected from 20 different informants with each one writing five samples of each character in an entirely unconstrained way. An accuracy of 86.5% obtained considering the structural features. Khanale and Chitnis [70] in 2011 suggested an artificial neural network for the purpose of recognizing Devanagari script. The scheme tested on a database comprises 18000 samples for all 45 basic characters which they have collected from forty persons. A two layer feed forward neural network with 35 units in the input layer and ten units in the output layer utilized and accuracy achieved up to 96%.

Bangla

An estimable amount of work has been reported from 2005 to till date for Bangla script recognition. Using the digital curvature feature of the basic Bangla characters Angshul

Majumdar [71] has proposed a scheme in which twenty different fonts of Bangla character used and the curvature feature extracted for each sample An accuracy about 96.80% obtained by applying the k-NN classifier. Bhattacharya et al. [72] in the year 2012 have proposed a two-stage character recognition for off-line Bangla script. Basu et al. [73] have also suggested a Multi-Layer Perceptron (MLP) based handwritten Bangla character recognition. Proposed scheme generates a feature vector of length 76 for each test sample. It includes 16 centroid features, 24 shadow features, and 36 longest-run features. They have incurred an average recognition rate of 96.67% over a database of 6000 samples. There are also research articles available in the support of the recognition of *Indic* languages such as *Telugu*, *Tamil*, *Gujarati*, *Assamese*, *Urdu*, *Gurmukhi*, and many other languages. An overall comparison of various schemes on some popular *Indic* vernaculars is shown in Table 1.2.

Table 1.2: Comparison of several schemes for the recognition of character in various *Indic* languages.

Script	Author(s)	Feature	Classifier	Success in %
Telugu	Rajashekararadhya et al. (2008) [74]	Zone and Distance metric	Feed forward back propagation neural network	96.00
	Pujari et al. (2004)[75]	Wavelet	Dynamic Neural Network	95.00
Tamil	Bhattacharya et al. (2007) [76]	Chain code	MLP	91.22
	Shanthi et al. (20010)[77]	Zoning	SVM	92.04
Gujarati	Prasad et al. (2009) [78]	Shape	Neural Network	70.66
	Desai et al. (20010)[79]	Structural	Feed forward neural network	82.00
Gurmukhi	Kumar et al. (2011) [80]	Diagonal and transitions features	*k*-NN	94.12
	Siddharth et al. (2011)[81]	Statistical	SVM with RBF Kernel	95.05
	Kumar et al. (2012)[82]	Structural feature with PCA	*k*-NN and SVM	97.70
	Aggarwal et al. (2015)[83]	Gradient and Curvature Features	SVM	98.56
	Verma et al. (2015)[84]	Zoning	SVM	92.09

Odia

The research on the recognition of Odia character started in late 90's. In the year 1998, S. Mohanty [85] has proposed a system that utilizes the Kohonen neural network for the classification of Odia character. The average distance calculated per pattern in each cycle up to certain threshold value. The prediction for a particular class depends on the output concerning a weighted sum formula. The reliability of the system is uncertain because the simulation carried out only for five characters. Chaudhuri et al. [40] in 2001, have proposed a scheme to discern Odia characters by the use of different structures present within it. Various structural features such as stroke and run length number in character extracted for each character and a decision tree classifier used for classification. The accuracy recorded is about 96.3% on average. However, their system is restricted to printed Odia fonts only. Pal et al. [86] have suggested a scheme utilizing the curvature feature which is calculated using bi-quadratic interpolation method. The gradient energy evaluated for each image and 392 features extracted. Further, a quadratic classifier has been utilized for classification. The suggested scheme tested on a dataset comprises 18,190 samples and the accuracy incurred is about 94.6 percentage. Meher et al. [87] have identified the difficulty of recognizing characters with cursiveness in their written form due to the presence of vowel modifiers and complex characters. A structural characteristic of the character is used to divide the whole dataset into two groups. Separately, the back propagation neural network is used for each group and obtained an overall accuracy of 91.24%. Kumar et al. [88] have suggested an Ant-Miner algorithm for the recognition of Odia character during the year 2013 in which they have achieved the recognition rate up to 90%. For the recognition of Odia character, few schemes have been observed in the literature. One of the essential reason is that no such standard database in this regards available for research use. Secondly, Odisha is not so developed as other states in India, and many regional offices still follow the manual processing of documents. The reported results are still not palatable, and consequently, it requires improvement. Along these lines, in this postulation, an endeavor has been made to make a standard database comprising of every atomic character and to find schemes to recognize them.

1.4 Motivation

Examining different schemes that had undertaken since four decades towards the recognition of many languages divulge a modest contribution on Indian vernaculars. However, it has been observed that the work accomplished in the case of *Odia* script is quite less. Over 25% out of the total 42 million population in Odisha are illiterate [89]. People are using the regional language, i.e., Odia in their day-to-day activity. It has many territorial applications including digitization of old magazines, historical documents, official records and much more. Being one of the classical language of India it needs more thrust towards the development of OCR for this language. The design of such system which can perceive an Odia character and no doubt there exists enough scope to improve the recognition rate of Odia character set. In this thesis, efforts have been made to develop schemes for recognition of Odia characters and numerals. In this regard, it turns out to be very much essential to have a sizable database with samples of various shape, orientation, and of different stroke width. Apart from this, it is quite apparent that feature extraction plays a significant role in the process of OCR.

1.5 Objectives

Necessity is the mother of invention. Looking into the perspective of OCR for Odia vernacular and the amount of work done till date, it impels us to contribute more towards the recognition of Odia script. In this thesis, few schemes have been suggested not only to extract relevant features from Odia character set but also utilize standard classifiers to recognize them. In particular, the objectives are narrowed to –

- design of databases for handwritten Odia numeral and character.
- exploit structural features to recognize the atomic Odia character.
- utilize Discrete Orthogonal S-Transform (DOST) features for recognition.
- propose structural features for each character and use multiple classifiers for recognition.
- explore and investigate Deep Neural Network to recognize the Odia character.

The flow graph of our research work delineated in Figure 1.4.

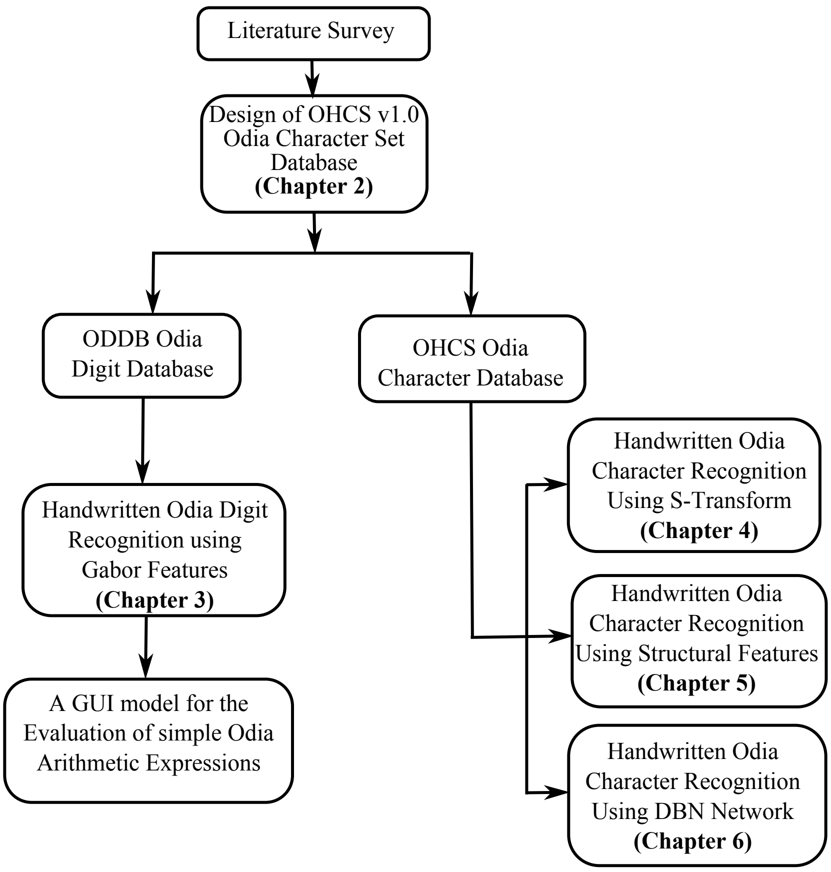

Figure 1.4: Flow Diagram of Our Research Work

1.6 Classifiers Used

From the existing classifiers, the linear SVM and Back Propagation Neural Network (BPNN) have been considered to solve the two class and multi-class problems in the proposed schemes. In this section, the mathematical description of the classifiers has been discussed in brief.

Linear Support Vector Machines (SVM)

It is a supervised learning method [90, 91] that, in general, applied to two class classification problem. This is a binary classifier which is further extend-able to solve multi-class classification problem. Formally, in mathematical language, SVMs construct linear separating hyperplanes in high-dimensional vector spaces. Data points are viewed as $(\vec{x}, y)$ tuples, $\vec{x} = (x_1, ..., x_p)$ where the x_j are the feature values and y is the classification (usually given as $+1$ or -1). Optimal classification occurs when such hyperplanes provide maximal distance to the nearest training data points which are called support vectors. Intuitively, this makes sense, as if the points are well separated, the classification between two groups is much clearer. Any hyperplane can be represented mathematically as,

$$\vec{w}.\vec{x} + b = 0 \tag{1.1}$$

where,

$$\vec{w}.\vec{x} = \sum_{i=1}^{p} w_i.x_i \tag{1.2}$$

A two class problem is said to be linearly separable if there exists no less than one hyperplane by the pair (w, b) which effectively orders all training samples (see Figure 1.5) for a given training set of samples. Once trained with the training samples, the linear SVM

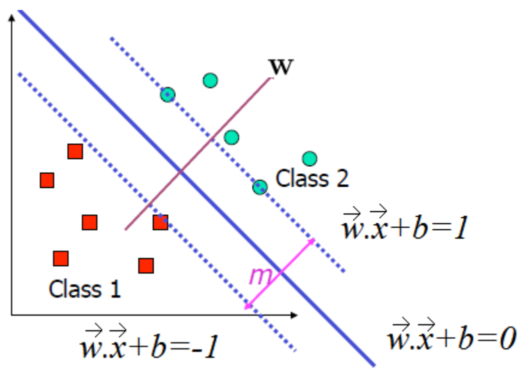

Figure 1.5: Linear SVM classifier with the hyperplane defined by $(\vec{w}.\vec{x} + b = 0)$

used to predict the class of a new pattern, different from the training samples. The hyperplane defined in Equation 1.1, assigns a sample to either +1 when $(\vec{w}.\vec{x} + b > 0)$ or -1 otherwise. Hence the expression $\vec{w}.\vec{x} + b$ decides the class that means a positive value indicates one class, and a negative value the other class. There could be the possibly infinite number of hyperplanes exist, but we must find the hyperplane which has maximum geometric margin.

Artificial Neural Network

Artificial Neural Network works on the similar principle as a neural system of living things. In more subjective terms, neurons can be comprehended as the subunits of a neural system in an organic brain as appeared in Figure 1.6. If the accumulated signals received by the dendrites in the cell body of the neuron exceeds a certain threshold, an output signal is generated that which will be passed on by the axon. A couple of years after, Frank Rosenblatt

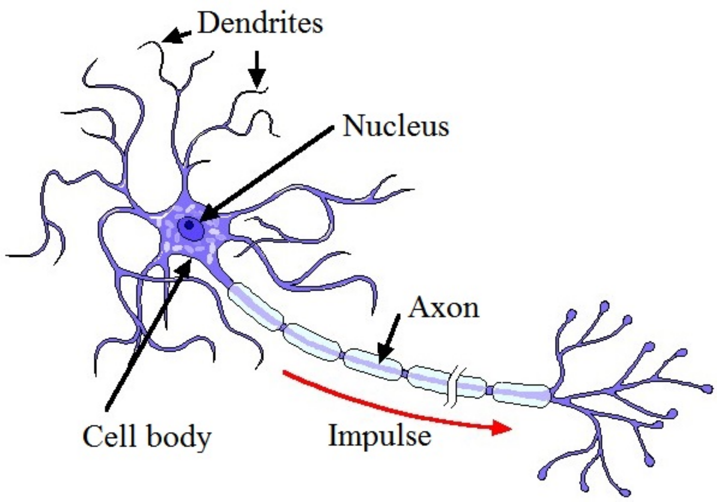

Figure 1.6: Schematic of a biological neuron.

published the main idea of the perceptron learning principle [93]. The initial thought was to characterize an algorithm to take in the estimations of the weights that are then increased with the input features to make a decision whether neuron fire or not. Rosenblatt presented the fundamental idea of a single perceptron in 1958 that computes a single output from multiple real-valued inputs by forming a linear combination according to its input weights. It passes through an activation function, and the output of nonlinear activation function decides the class. Mathematically this is represented as,

$$y = \varphi\left(\sum_{i=1}^{n} w_i x_i + b\right) = \varphi\left(w^T x + b\right) \tag{1.3}$$

where the vector $\mathbf{w}$ denotes weights, $\mathbf{x}$ represents inputs, b refers the bias and φ denotes the activation function. A signal-flow graph of this operation is shown in Figure 1.7. The unit step function used as activation function in Rosenblatt's perceptron. He demonstrated that the perceptron algorithm converges if two patterns are linearly separable. In this way, it makes a line which positioned between the classes. However, issues emerge if the classes can't be separated impeccably by a linear classifier. Over these lines, in 1970 Kohonen

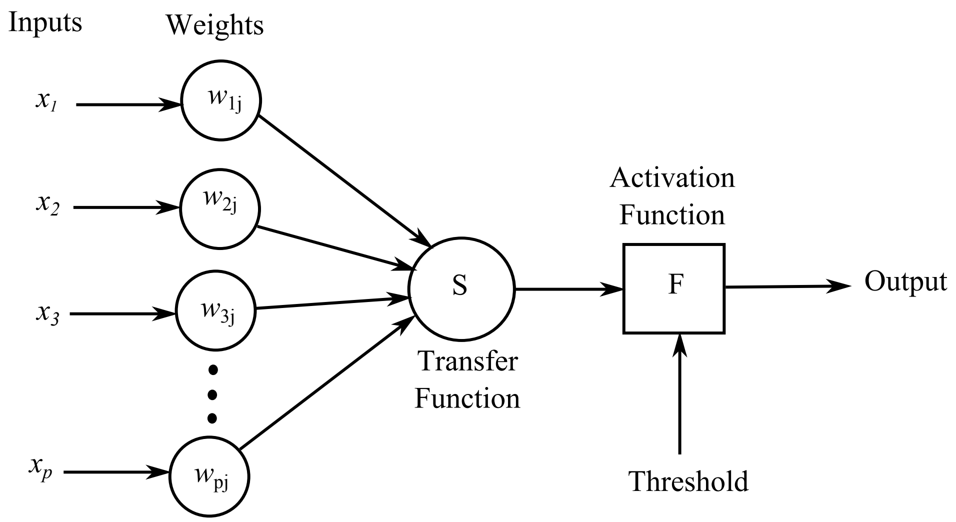

Figure 1.7: Schematic of a simple perceptron.

and Anderson [94, 95] and later in 1980, Carpenter and Grossberg [96] proposed multilayer perceptron on ANN.

Back Propagation Neural Network: A Multilayer Perceptron

Artificial Neural Network (ANN), being the workhorse in today's machine learning algorithms, has turned into a potential tool for non-linear classification [97]. Multi-Layer Perceptrons (MLPs) are stronger than the single-Layer models because the computation is carried out utilizing a set of simple units with weighted connections between them. Moreover, there are learning algorithms for the adjustment of the network parameters and make it compelling to solve many classification problems. When the training accomplished with the back-propagation algorithm, the MLP gives a better result and works faster than prior ways to deal with learning [98]. The algorithm runs in two phases. In the first phase, the predicted outputs corresponding to the given inputs are evaluated. Secondly, the partial derivatives of the cost function on the different parameters are propagated back through the network. The parameters of the network are optimized and the whole process iterated until the weights converged. The abstract model of a three layer BPNN structure shown in Figure 1.8 and the implication of each layer described below,

- **Input layer**: It takes input one feature vector at a time for a given feature matrix.
- **Hidden layer**: The weighted sum with the bias are calculated on the output of the input layer, and pass through an activation function. Multiple layers are considered in the hidden layer for profound learning. The number of units in this layer ought not to be less to model complex decision boundaries and ought not to be more to make the system over-fitting.
- **Output layer**: The number of units in the last layer is determined by the total number of classes to be classified.

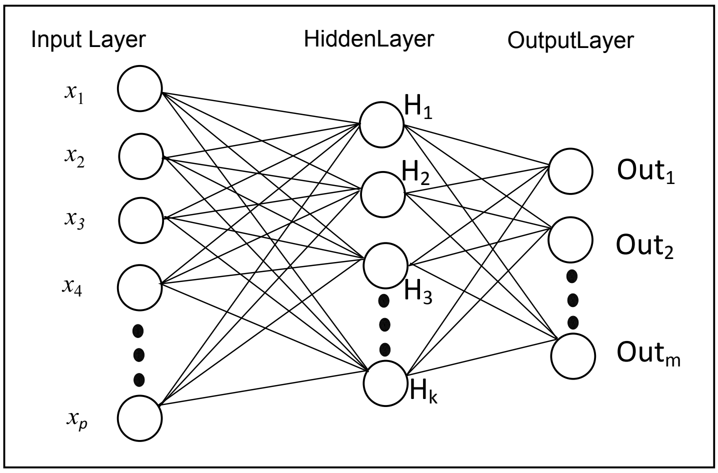

Figure 1.8: General structure of a multi layer multi class ANN

Training Criteria

Assuming we have supervised training samples $\{\{x_1, t_1\}..., \{x_p, t_p\}\}$ where x_i, where $i = 1, 2, 3, .., p$ are the inputs to the network and t_i is the corresponding target value. Training of the network continues till the error rate is acceptable. Following are some standard training methods:

- **Least squares error**

$$E = \frac{1}{2}\sum_{i=1}^{p} \|Out(x_i) - t_i\|^2 \tag{1.4}$$

- **Cross-Entropy for two classes**:

$$CE_1 = -\sum_{i=1}^{p} t_i \log(Out(x_i)) + (1 - t_i) \log(1 - Out(x_i)) \tag{1.5}$$

- **Cross-Entropy for multiple classes**:

$$CE_2 = -\sum_{i=1}^{p}\sum_{j=1}^{m} t_{ij} \log(Out_j(x_i)) \tag{1.6}$$

It is usually better to use cross-entropy error than least square error to evaluate the quality of the neural network because it provides a smoother learning rate curve over various node output values.

Performance Measurement Parameters

Usually, the confusion matrix is considered for the performance evaluation of any classifier. Assume there are two classes namely A (Positive class) and B (Negative class). The performance measures are defined below.

- True Positive (TP): It indicates how many values of A correctly classified as A.
- False Negative (FN): It gives a value showing the number of samples of A classified as B.
- False Positive (FP): Total number of samples of B wrongly classified as A.
- True Negative (TN): It finds the whole samples of class B genuinely classified as B.

Utilizing the above parameter values the $sensitivity$, $specificity$, $fall\ out$, and $miss\ rate$ are evaluated by the following equations.

$$Sensitivity\ (or\ TPR) = \frac{TP}{TP+FN} \tag{1.7}$$

$$Specificity\ (or\ TNR) = \frac{TN}{TN+FP} \tag{1.8}$$

$$Fall\ out\ (or\ FPR) = \frac{FP}{TN+FP} \tag{1.9}$$

$$Miss\ rate\ (or\ FNR) = \frac{FN}{TP+FN} \tag{1.10}$$

$$Accuracy = \frac{TP+TN}{TP+FN+FP+TN} \tag{1.11}$$

where TPR is the true positive rate, TNR is the true negative rate, FPR is the false positive rate, and FNR is the false negative rate.

1.7 Thesis Layout

The overall work is organized into seven different chapters including introduction and conclusion. Out of five contributions, one chapter outlined the design of databases for handwritten Odia character set. Another chapter is devoted to recognition of Odia handwritten digits. Other three propositions are on handwritten Odia character recognition. The chapters are discussed below in sequel.

Chapter 2: Development of Handwritten Databases for Odia Language
In India, some organizations have kept up provincial databases of various dialects, yet the number is observed to be less on account of Odia script. Non-availability of a sizable handwritten Odia database propelled us to design a goodly database for the proposed scheme validation. In this regard, samples have been collected exquisitely from individuals through a digital note maker where each person contributes samples twice at a different time. The database comprises 18240 $(160 \times 2 \times 57)$ samples that are collected from 160 individuals. This database named as Odia handwritten character set version 1.0 (OHCS v1.0). Further, it has been segregated into two subsets namely ODDB and OHCS, where ODDB contains 3200

isolated digit samples and OHCS comprises 15040 Odia atomic characters. To strengthen the size of the ODDB database 580, more samples for each digit are gathered.

Chapter 3: Handwritten Odia Digit Recognition using Gabor Filter Bank (HODR-GFA)

In this chapter, an array of Gabor filters has been utilized for recognition of Odia digits. In this regard, each image is divided into four blocks of equal size. Gabor filters with various scales (S) and orientations (R) applied to these sub-images keeping other filter parameters fixed. The average energy is computed for each transformed image to obtain a feature vector of size $S \times R \times 4$ for each digit. Further, a Back Propagation Neural Network (BPNN) has been employed to classify the samples taking the feature vector as input. It has been observed that filters with $S = 5$ and $R = 12$ gives better performance as compared to other combinations. Besides, the proposed scheme has also been tested on standard digit databases like MNIST [42] and USPS [100]. Toward the end of this chapter, an application has been designed to evaluate simple arithmetic equations written in Odia language.

Chapter 4: Handwritten Odia Character Recognition using Discrete Orthogonal S-Transform (HOCR-DOST)

This chapter presents a multi-resolution based scheme coined as HOCR-DOST, to extract features from Odia atomic character and recognize them using the back propagation neural network. Considering the fact, that few Odia characters have a vertical line present at the end, the whole dataset divided into two subgroups; namely, *Group I* and *Group II* such that all characters in *Group I* have a vertical line and rest are in *Group II*. In this regard, a perceptron has been utilized that takes the shape feature as input. In addition to this, the two-dimensional Discrete Orthogonal S-Transform (DOST) coefficients are extracted from images of each group; subsequently, Principal Component Analysis (PCA) has been applied to find significant features. For each group, a separate BPNN classifier is utilized to recognize the characters. The overall accuracy recorded to be 98.55%.

Chapter 5: Structural Feature-based Classification and Recognition of Handwritten Odia Character (HOCR-SF)

The HOCR-SF scheme works in two phases. In the first phase, the overall Odia character set has been classified into two groups using a Support Vector Machine (SVM) classifier. Group I comprises all characters with a vertical line present at the end whereas rests fall into Group II. For the classification of the samples into two groups, each character image resized to 32×32 and represented as a vector of length 32 containing the number of pixels in each column of that image. The mean value of the lower half and max of the upper half together represents a feature point of the character and used as input to the classifier. The structural features of the character of each group are extracted and fed to a BPNN for recognition. Separate BPNN networks utilized for classifying the characters in each group.

Chapter 6: Recognition of Atomic Odia Character using Deep Learning Network (HOCR-DBN)

A semi-supervised learning strategy, i.e., Deep Belief Network (DBN) [131] has been proposed in this chapter. An approximation algorithm namely Contrastive Divergence (CD) is investigated to optimize the network parameters. Apart from the input and output layers, the proposed DBN structure has three hidden layers. The DBN works on an unlabeled dataset. An accuracy of 91.2% recorded for OHCS dataset. Though the accuracy is not at par with other proposed schemes, it performs better than the state of the art schemes. Another advantage of this scheme is that it requires no prior knowledge about the label of the input data.

Chapter 7: Conclusions and Future Work

This chapter provides the concluding comments of the submitted work. The extensions for further research laid out toward the end.

Till now, we have discussed in a nutshell about the contributions in this thesis. These are discussed more concretely, in subsequent chapters, in sequel.

Chapter 2

Indian Languages and Design of Handwritten Databases for Odia Language

The present work deals with the recognition of characters of Odia language which is one of the dialects spoken in the eastern part of India. India is a nation of various and contrasting cultures, and its semantic outline is pretty much as assorted. Despite the fact that in India, Hindi is the national dialect however as indicated by the Constitution of India, there are twenty-two official languages. These scripts are: *Assamese, Bengali, Bodo, Dogri, Gujarati, Hindi, Kannada, Kashmiri, Konkani, Maithili, Malayalam, Manipuri* (sometimes called Meitei), *Marathi, Nepali, Odia (Formerly "Oriya"), Punjabi, Sanskrit, Santali, Sindhi, Tamil, Telugu* and *Urdu*. Be that as it may, the real spoken dialects are up to 2500 because of the distinctive lingos. The total number of characters, including vowels and consonants, for some most spoken *Indic* scripts, are listed below.

- Devanagari script has about forty-five primary characters, of which eleven are vowels and thirty-four consonants.
- The complete Tamil script consists of the thirty-one letters including 12 vowels, 18 consonants, and a unique character.
- The Bangla script has a total of 11 vowels and 32 consonants.
- ***The modern Odia language has 47 letters in its alphabet system. It includes 11 vowels and 36 consonants.***
- Telugu script has a total of 13 vowels and 39 consonants.
- Gujarati language also has 47 alphabets in its character set including 11 vowels and 36 consonants.
- The Assamese script is similar to the Devanagari, and it has 11 vowels and 41 consonants.

- The Urdu language comprises 52 total characters including 39 basic letters and 13 extra characters.
- Kannada script has a total of 14 vowels and 34 consonants.
- The modern Malayalam alphabet has 15 vowel letters, 41 consonant letters.
- Modern Gurmukhi has thirty-eight consonants, nine vowel symbols, and three special symbols.

Figure 2.1: Odisha (Highlighted) the state in India

Orissa, one of the states in India (see Figure 2.1) with diverse culture and religion, came into existence on 1st April 1936 as a province on a linguistic basis during British rule. It became Odisha under a resolution of the Parliament on the 96th Amendment of the 8th scheduled on 23 September 2011, which substituted “Odia” for “Oriya” and “Odisha” for “Orissa”. It is one of the classical languages in India. The population of Odisha in 2015 was estimated to be 44.3 million. Odia is the mother tongue of this state. It is a predominant language, where not only more than 80% of the population of the state Odisha speaks this language but also, spoken in parts of some nearby states such as Chhattisgarh, Jharkhand, West Bengal, and Andhra Pradesh. In Jharkhand, Odia is considered to be the second official language.

2.1 Odia Character Set and their Characteristics

The Odia script derived from the Brahmi script of ancient India. The writing style of Odia language is unique as compared to any other regional languages in India. The modern Odia script consists of 11 vowels, 36 consonants, ten digits. To acquaint the reader on complete

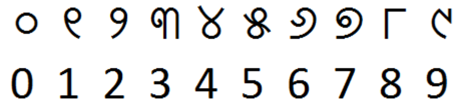

Figure 2.2: Printed Odia digit and their corresponding English numeral

Figure 2.3: Vowels of Oriya Script with English Transliteration.

Figure 2.4: Consonants of Oriya Language with English Transliteration.

Odia character set the printed versions are shown in Figures 2.2, 2.3, and 2.4. Apart from these basic Odia character set, the Odia ligatures may be formed either merging a vowel diacritic with the consonant or by clustering two or more consonants. There are nearly 116 composite characters also known as *Juktakhyaras*. A sample set of Odia ligatures shown in Table 2.1 where the character is combined with all vowels to produce different composite characters. It also may be observed that the Odia characters are mostly round shaped similar

Table 2.1: Sample ligatures formed when all vowels are fused with the consonant 'gha', i.e., 'ଘ'.

Oṛiyā	ଘ	ଘା	ଘି	ଘୀ	ଘୁ	ଘୂ	ଘୃ	ଘେ	ଘୈ	ଘୋ	ଘୌ
Translit.	gha	ghā	ghi	ghī	ghu	ghū	ghṛ	ghe	ghai	gho	ghau

to Devanagari and Bengali but unlike Odia letters, the letters have a horizontal line on the top (called *Sirorekha*). Considering this the following Table 2.2 clearly shows a closer relation of Odia dialect with that of Devanagari and Bengali. Along with consonants and vowels there are some special symbols namely *halanta*, *visarga*, *anusvara*, and *chandrabindu* which when combined with a consonant, a composite character is formed. In Figure 2.5, we demonstrate

Table 2.2: Comparison of the shape of the vowels in Odia, Devanagari, and Bangla language

Oṛiyā	ଅ	ଆ	ଇ	ଈ	ଉ	ଊ	ଋ	ଏ	ଐ	ଓ	ଔ
Devanāgarī	अ	आ	इ	ई	उ	ऊ	ऋ	ए	ऐ	ओ	औ
Bengali	অ	আ	ই	ঈ	উ	ঊ	ঋ	এ	ঐ	ও	ঔ

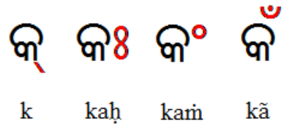

Figure 2.5: Usage of Glyphs with a Consonant ‘କ’

the use of those modifiers over the character “କ’, i.e., ‘ka’. All characters are of different shape and size but there are few characters that looks exactly same. The digit ‘0’ (*Zero*) and the Odia alphabet ‘ଠ’ (spell as ‘tha’) are similar in shape. Few such characters are shown in Figure 2.6. Odia script is not case sensitive and the end of a sentence is marked by a vertical

Figure 2.6: Similar shape characters in Odia alphabet

line (‘|’), not by a period or dot (‘.’). But, few characters have a vertical line at the end. Hence, character level segmentation, as well as the similarity in shape of few characters, makes the process quite challenging. In this thesis, an attempt has been made to propose notable features of isolated Odia digits and atomic characters for recognition.

2.2 Why at all we need a database?

Necessity is the mother of inventions. It has been observed that a good number of articles published in the area of handwritten character recognition in many languages, which depend on some nearby databases. Learning method could be supervised, semi-supervised, unsupervised, and reinforcement learning that is used to identify the character in a text which is independent of font and size. In supervised learning, the system is groomed up with a labeled dataset and then it is used to predict the output of the system given new inputs [19]. In the case of semi-supervised learning paradigm a labeled dataset small in size used for training the network and a large sample of test images used for classification. The unsupervised algorithm works on the unlabeled dataset. Thus, the database plays a significant role in the process of recognizing any object. It is needless to say that the OCR designed for printed documents cannot extend to handwritten recognition due to enormous variability in the style of writing of different individuals. In India, some organizations have maintained databases of different *Indic* languages.

2.2.1 Popular Digit Databases for *Non-indic* and *Indic* Languages

There exist standard handwritten English digit databases, viz., Modified National Institute of Standards and Technology database (MNIST), United State Postal Service (USPS), CEDAR, and Semeion databases. As far as databases for *Indic* languages are concerned, few organizations have designed and maintained databases for some popular languages.

MNIST

MNIST [99] is a handwritten English digit database comprises 60000 samples in the training set and 10000 in the testing set. The MNIST database is available at `http://yann.lecun.com/exdb/mnist/`. The images of MNIST database derived from two NIST's databases, i.e., Special Database 1 (SD-1) and Special Database 3 (SD-3). The high school students and employees of the United States have contributed to the design of NIST database. Half of the samples in the training set are from SD-3 and rest are from SD-1. Similarly, 5000 samples from SD-3 and another 5000 from SD-1 are chosen for the test set. All sample images are in grayscale, and each of the samples is size-normalized to 28×28. A Hundred samples of MNIST database shown in Figure 2.7.

Figure 2.7: Hundred sample digit images from MNIST dataset

USPS

The U.S. Postal Service envelopes are scanned and the digits are extracted and stored in a database named USPS. The original scanned digits are binary and of different sizes and orientations. Thus, each image is processed through several modules such as noise reduction, slant correction, and size normalization. As a result all images size is standardized to 16×16. The whole dataset consists of 9298 samples, and the frequency of each digit is shown in the

Figure 2.8: Hundred sample digit images from USPS dataset

Table 2.3. Sample hundred images from USPS database are shown in Figure 2.8. The dataset and this description is made available at `http://www-stat.stanford.edu/\~{}tibs/ElemStatLearn/data.html`.

Table 2.3: Frequency of digits in USPS

Frequency	0	1	2	3	4	5	6	7	8	9	Total
Training set	1194	1005	731	658	652	556	664	645	542	644	**7291**
Testing Set	359	264	198	166	200	160	170	147	166	177	**2007**

Semeion Handwritten Digit Data Set

The dataset was first created by Tactile Srl, Brescia, Italy and in 1994 they have donated to Semeion Research Center of Sciences of Communication, Rome, Italy, for machine learning research [100] development. A total of 1593 handwritten digits, which are collected from

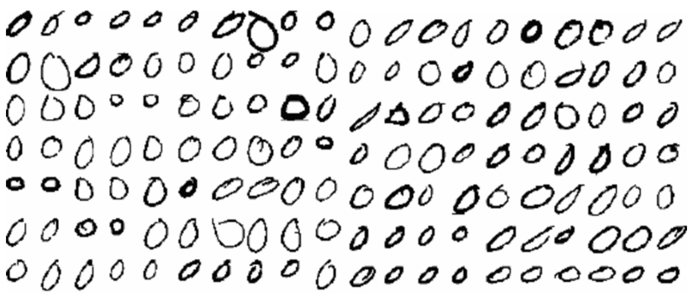

Figure 2.9: One Hundred forty sample of the digit Zero from Semeion Handwritten Digit Database

around 80 individuals, are scanned, resized to 16×16 in a gray scale of 256 values. Further, all images are converted to binary with a threshold of intensity value for each pixel as 127. To keep the variation in samples, all the informants wrote the digit the first time in the normal way and the second time in a faster way. Figure 2.9 gives 140 sample images of the digit zero. The size of the database is not adequate to be tested.

CEDAR for English Digit

Center of Excellence for Document Analysis and Recognition (CEDAR) is a research center at the University at Buffalo, New York. CEDAR owned databases for different languages which are available at `http://www.cedar.buffalo.edu/Databases/index.html`. The handwritten digits collected from the ZIP codes of envelopes at the Buffalo Post Office. The whole database divided into training and testing sets. Approximately 10% of the images included in the testing set and the remaining of the data are retained for training purpose. The frequency of the digits in both sets is shown in Table 2.4.

Table 2.4: Frequency of each digit in CEDAR database

Frequency	0	1	2	3	4	5	6	7	8	9
Training set	811	1160	778	467	607	342	469	429	346	393
Testing Set	102	136	103	68	63	41	47	48	46	53

Bangla Digit Database

Ujjwal Bhattacharya and his team [101, 102] have designed the database for Bangla digits at ISI Kolkata, India. The samples of the database have been collected from 465 postal letters and 268 job application forms. To make the database voluminous, separately a formatted form has been circulated among different individuals the informants has to fill-up manually. The existing database of handwritten isolated Bangla numeral consists of 23392 samples written by 1106 persons. The shape and look of few Bangla digits are almost similar to some of the Odia numerals.

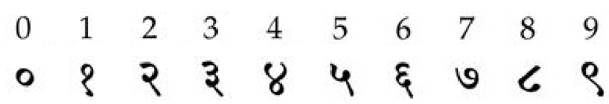

Figure 2.10: Bangla digits from 0 to 9 along with their corresponding English values

Odia Digit Database

A database of isolated handwritten Odia numeral has been prepared by Bhattacharya and Chaudhuri [101] in the year 2005 at ISI Kolkata, India. Exactly 356 persons were engaged in the process of data collection. It has 5970 samples gathered from 166 job application forms, 105 mail pieces, and rest samples collected in person. Finally, the dataset is divided into a training set consisting of 4970 samples, and a test set comprises 1000 samples. As compared to the dataset size of Bangla digits, the size of Odia digit database is too thin.

2.2.2 Handwritten Character Database for *Non-indic* and *Indic* Languages

Here, are some well known standard databases on the handwritten character.

CEDAR for English Alphabet

Since 1978, the Department of Computer Science of University at Buffalo has been contributing in the area of pattern recognition. The database prepared by CEDAR contains a good number of segmented characters collected from address blocks of envelopes at the Buffalo Post Office. Approximately ten percent of the whole available images belongs to the test set. The frequency of the characters in both training and testing sets given in Figure 2.5 and 2.6 respectively. The training set comprises 24947 samples whereas there are only 2890 samples in the test set.

Table 2.5: Frequency of each English alphabet in the Training set of CEDAR dataset

Character	A	B	C	D	E	F	G	H	I	J	K	L	M
Frequency	1237	595	588	388	490	287	143	274	490	68	160	563	588
Character	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Frequency	1022	905	516	3	749	834	441	268	201	249	112	259	24
Character	a	b	c	d	e	f	g	h	i	j	k	l	m
Frequency	527	84	211	249	736	120	93	205	803	1	94	684	184
Character	n	o	p	q	r	s	t	u	v	w	x	y	z
Frequency	469	833	129	5	460	490	428	333	127	102	173	136	15

Table 2.6: Frequency of each English alphabet in the Testing set of CEDAR dataset

Character	A	B	C	D	E	F	G	H	I	J	K	L	M
Frequency	162	69	51	49	57	32	19	25	56	15	19	87	59
Character	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Frequency	123	102	59	2	97	77	54	31	27	30	19	39	7
Character	a	b	c	d	e	f	g	h	i	j	k	l	m
Frequency	68	8	17	26	107	16	14	23	70	2	14	69	21
Character	n	o	p	q	r	s	t	u	v	w	x	y	z
Frequency	60	84	7	2	47	29	46	26	20	15	10	15	0

Bangla Basic Character Database

Indian Statistical Institute (ISI) have designed the Bangla off-line handwritten basic character database having a total of 37858 sample images for all characters in the script. The number of samples for each character disseminated unequally. Even though the training set consists of 500 samples for each character and the corresponding test set contains the remaining images.

Figure 2.11: All Consonants of Bangla Script along with their English transliteration

Devanagari Character Database

In a cursory look, the Devanagari script appears different from other *Indic* scripts such as Bangla, Odia or Gurmukhi, but a closer examination reveals that they are similar to some extent. The people of Computer Vision and Pattern Recognition Unit (CVPR) at ISI Kolkata have maintained a repository on the Devanagari off-line handwritten character. The present database consists of approximately 30000 samples of handwritten characters. However, these are not evenly distributed over 49 possible classes. A few samples from this database shown in Figure 2.12.

Figure 2.12: List of alphabets from Hindi Script along with their English transliteration

2.3 Design of ODDB and OHCS Database for Odia Language

The Odia numeral database consists available at the home page of Computer Vision and Pattern Recognition Unit, ISI Kolkata, comprises 5970 samples but no repository of handwritten Odia character is available for research. Certainly, we urged strongly to prepare a sizable database not only for Odia character but also for Odia digit. The first version of

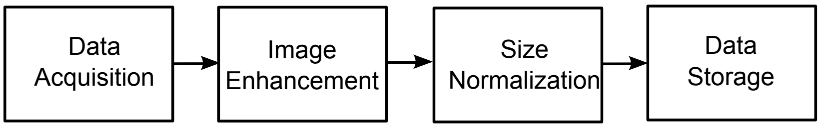

Figure 2.13: Steps involved in the process of Database Design

the database for Odia handwritten character set named as OHCS v1.0. Samples collected from various informants at different point of time and processed at Image Processing and Computer Vision (IPCV) Laboratory, Rourkela, India. It includes both Odia alphabet and digit with a total of 18240 samples. The database segregated into two sets namely Odia Digit Database (ODDB) and Odia Handwritten Character Set (OHCS). The ODDB database contains initially 3200 samples of digit images which further increased to 9000 after collecting 580 more samples for each digit. On the other hand, the OHCS database has 15040 samples of Odia atomic characters. The steps adopted in the process of designing the database on Odia character set depicted in Figure 2.13. A few standard techniques have been applied amid the way toward making the databases. Every stage used in this regard exposited next in sequel.

Figure 2.14: TechNote A414 from i-Ball

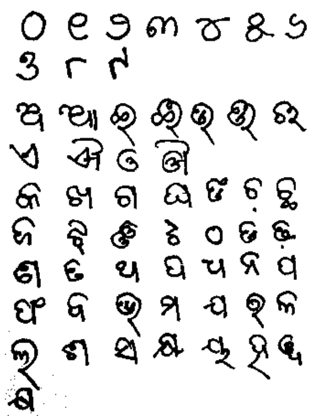

Figure 2.15: Sample Odia Handwritten character set

2.3.1 Data Collection

For data collection, commercially available digital note-maker device from i-Ball shown in Figure 2.14 has been utilized. It is a plug and play device and comes with a digital pen. A formatted A4 size paper with all characters printed before placed on top of it and the informant asked to write the character using the digital pen. It captures and stores everything in it. It has internal memory of size 32MB and capable of storing more than 100 A4-size pages of full handwritten notes. Each image in the database preprocessed and stored in the computer system directly. Corresponding to the Odia character set, a handwritten sample version provided in Figure 2.15. All samples including ten digits, eleven vowels, and thirty-six consonants collected from 160 distinct individuals belonging to various age, gender, and community at different point of time. Two samples from a person are collected at different point of time to generate a larger dataset.

Image Enhancement

Image enhancement helps in transforming the image into precise form so that the results are more suitable for analysis. Methods such as removal of noise, sharpening, contrast stretching, histogram equalization, image binarization [104], skew angle detection and correction [105], segmentation, image filtration, etc. applied to an image not only to improve the quality of the picture but also helps in identifying the key features. Initially, each of the samples thus collected is binarized and further, Wiener filter [106, 107] has been applied to remove isolated unwanted pixels. The skew/slant corrections have been made, and all samples are size-normalized and centered in a fixed-size of 81×81.

2.3.2 Data Repository

All images in both ODDB and OHCS are stored in *.bmp* format to hold most extreme image information. This database made in the IPCV Laboratory of National Institute of Technology Rourkela, India. The documents alongside their storage capacity for the two databases appeared in Table 2.7. Separate folders have been created to store samples of a particular character, and all images of one sample doled out one label. The complete database for

Table 2.7: List of files with size in bytes.

Database	File name	Size in bytes
OHCS	OHCS-train-samples.gz	10246
	OHCS-train-labels.gz	4980
	OHCS-test-samples.gz	3614
	OHCS-test-labels.gz	1016
ODDB	ODDB-train-samples.gz	3246
	ODDB-train-labels.gz	1206
	ODDB-test-samples.gz	1986
	ODDB-test-labels.gz	816

Odia handwritten character is available at `http://nitrkl.ac.in/Academic/Academic_Centers/Data_Computer_Vision.aspx` and down-loadable for research purpose. The zipped folder contains forty seven folders: NITROHCS001 through NITROHCS047. Each of the folder contains 320 handwritten atomic samples of a character class. The files inside each folder are named OHCS001 through OHCS320. Sample set of letter 'ଣ' from OHCS v1.0 database is shown in Appendix. The preprocessed handwritten dataset is available for ready use by the researchers.

Chapter 3

Handwritten Odia Digit Recognition using Gabor Filter Bank (HODR-GFA)

From the literature, it has been observed that amongst all phases of OCR, the exactness of any system exceptionally relies upon the feature extraction stage. The input to any OCR is either the image as a whole or the features extracted from an image. An image consists of low-frequency components that comprise smooth regions in the image and high-frequency components that come from the edges, i.e., the sharp change in the intensity values. To analyze any image we need to transform the image to a specified domain. Image transformation is a necessary step to extract the discriminant features from an image. Filtering, data compression, feature extraction, etc. are few applications [109] of image transformation. The Fourier transformation is commonly used to analyze the frequency components of an image but, after transformation, we lose the time information, and it's hard to tell where a particular frequency occurs. Thus, to find the discriminant features from an image, we need multi-resolution analysis that gives us time-frequency representation of a signal. In the realms of image processing, image transformation has got plenty of applications such as character realization, texture segmentation, edge detection, object identification, document analysis, image representation [110], etc. Gabor filters are widely used in image processing because it is invariant to illumination, rotation, scale, and translation. In addition to this, they impute optimal localization in both spatial and frequency domain and becomes a good candidate for feature extraction problems [111]. Based on the above findings, a scheme has been suggested that extracts Gabor feature from each sample at different scales and orientations and to classify a back propagation neural network is employed. Simulation is carried out on databases like MNIST, USPS, and ODDB.

This chapter organized as follows. Section 3.1 discusses the 1-D as well as 2-D Gabor functions, and also it includes mathematical properties along with the derivations. The proposed scheme portrayed in Section 3.2 with all necessary steps. Simulations are explained in Section 3.3. One of the applications on digit recognition presented in section 3.4 followed by the summary of work.

3.1 Gabor Filter

D. Gabor [112] introduced one-dimensional Gabor filter in the year 1946. Later in 1985, J. G. Daugman [113] has extended it to two-dimensional Gabor filter. A 2D Gabor filter is formed by modulating a sinusoidal plane of particular frequency and orientation with a Gaussian envelope. It is defined as:

$$G(x, y) = g(x, y) * s(x, y) \tag{3.1}$$

where $g(x, y)$ is a 2D Gaussian function and $s(x, y)$ is a complex sinusoid known as the carrier signal. More precisely, the $G(x, y)$ is defined as follows,

$$G_{\sigma,\phi,\theta}(x, y) = g_\sigma(x, y)e^{2\pi j\phi}(x \cos\theta + y \sin\theta) \tag{3.2}$$

where

$$g_\sigma = \frac{1}{2\pi\sigma^2}e^{-(x^2+y^2)/2\sigma^2} \tag{3.3}$$

is the Gaussian function. The parameters used in the above equations are described below.

- ϕ: is the frequency
- θ: is the orientation
- σ: is the standard deviation of the Gaussian envelope, i.e., scale.

Clearly, it shows that the filter has a real and an imaginary component. The two components represent the orthogonal directions, and together they form a complex number. The orientation must vary between the interval $[0^0, 180^0)$ to avoid redundant values due to the symmetry. The Gabor filter $G_{\sigma,\phi,\theta}(x, y)$ represented by a complex valued function as shown in equation 3.4.

$$G_{\sigma,\phi,\theta}(x, y) = R_{\sigma,\phi,\theta}(x, y) + jI_{\sigma,\phi,\theta}(x, y) \tag{3.4}$$

The real and imaginary parts are given in equation 3.5 and 3.6 respectively.

$$R_{\sigma,\phi,\theta}(x, y) = g_\sigma(x, y). \cos[2\pi\phi(x \cos\theta + y \sin\theta)] \tag{3.5}$$

and

$$I_{\sigma,\phi,\theta}(x, y) = g_\sigma(x, y). \sin[2\pi\phi(x \cos\theta + y \sin\theta)] \tag{3.6}$$

Given a window of size $W \times W$ (generally odd, i.e., $W = 2k + 1$), the discrete convolution of $f(x, y)$ with respective real and imaginary components of $G_{\sigma,\phi,\theta}(x, y)$ are

$$C_R(x, y|\sigma, \phi, \theta) = \sum_{n=-k}^{k}\sum_{m=-k}^{k} f(x + n, y + m)R_{\sigma,\phi,\theta}(x, y) \tag{3.7}$$

$$C_I(x,y|\sigma,\phi,\theta) = \sum_{n=-k}^{k}\sum_{m=-k}^{k} f(x+n,y+m)I_{\sigma,\phi,\theta}(x,y) \tag{3.8}$$

The energy $E(x,y|\sigma,\phi,\theta)$ at (x,y) is calculated by the equation 3.9 given below.

$$E(x,y|\sigma,\phi,\theta) = C_R^2(x,y|\sigma,\phi,\theta) + C_I^2(x,y|\sigma,\phi,\theta) \tag{3.9}$$

For extracting useful features from an image, an array of Gabor filters can be applied by varying the frequency, scale and orientation [114]. The proposed scheme creates a Gabor filter bank for different values of scale and orientation keeping the frequency constant. The imaginary and real parts of the array of Gabor filters in five scales and twelve orientations $(0, \pi/12, \pi/6, \pi/4, ..., 11\pi/12)$ at each scale are shown in Figure 3.1 and 3.2 respectively.

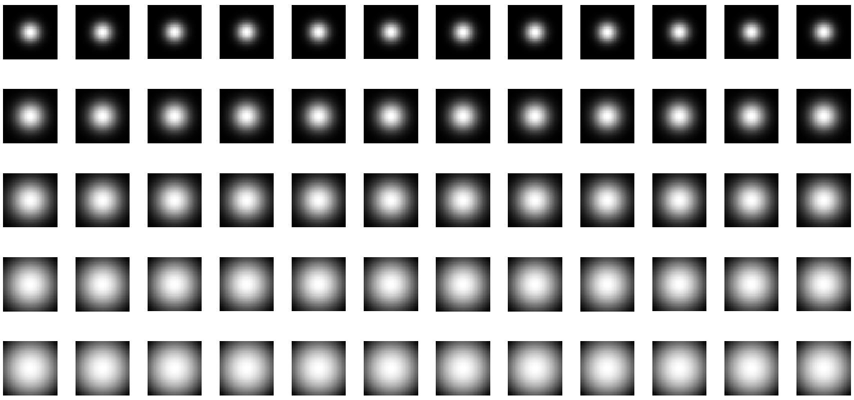

Figure 3.1: Imaginary part of Gabor filters 5 scales and 12 orientations. Each sub-figure shows the magnitude in log scale.

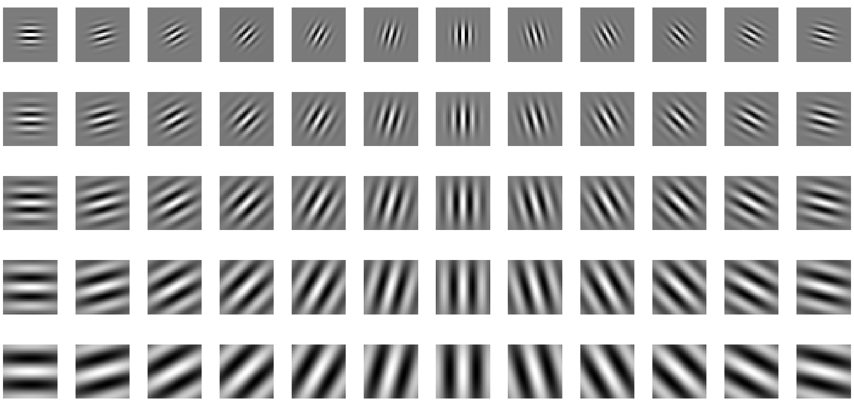

Figure 3.2: Real part of Gabor filters with five scales and 12 orientations.

3.2 Proposed Scheme

The proposed method uses Gabor filters in different scales (S) and orientations (R) at each scale. A whitening filter is used initially on the input gray scale image to preserve the dominant structural details and then normalized on the local contrast. The input image resized to $N \times N$. The image thus obtained is divided into four blocks of size $N/2 \times N/2$ each. These Gabor filters applied to each sub-image followed by a convolution operation. The average energy is computed passing each block sequentially through the Gabor filter array thus producing a feature vector of size $S \times R \times 4$ for each digit sample. Hence for N sample of images in the training set of ODDB database, the size of the feature matrix is $N \times S \times R \times 4$. To each feature set, a corresponding class label is assigned. For a unique character feature set along with the label constitutes a pattern and used subsequently in an artificial neural network (ANN) to classify any new symbol. The suggested network consists of $S \times R \times 4$ neurons in the input layer and ten at the output layer. The number of neurons in the hidden layer is chosen experimentally for the better convergence performance of the network. The schematic block diagram of the proposed scheme has been delineated in Figure 3.3. For validation of the proposed scheme, the train, test and validation sets are generated randomly for generalization. For all the three datasets we have used a randomized method

3.2.1 Experimental Set-up

Along with the ODDB dataset, the proposed scheme has been validated using two standard digit databases namely MNIST and USPS. The images of both MNIST and USPS datasets scaled to 32×32. Simulation is carried out in the MATLAB environment. The *dividerand()* method has been used to split the whole dataset into three sets: training, validation, and testing. The *training set* determines the optimal weights with the back-propagation rule. The *validation set* finds the optimal number of hidden neurons or decides a stopping criterion for the back-propagation algorithm. The *testing set* estimates the error rate and assess the performance of the network. The other network parameters need to be adjusted to avoid over-fitting of the network. The conventional back propagation is used to train the network. Each sample image of size 32×32 from the database divided into four blocks of equal size, i.e., 16×16. The number of scales considered to be as 4, 5, 6, and 8 and the number of orientations are 8 and 12. Following assumptions have been made in our experiment.

- The value of ϕ is considered to be 1.0.
- The number of blocks is chosen to be 4, i.e., 2×2.
- Scaled conjugate gradient method adopted for faster training of the network.

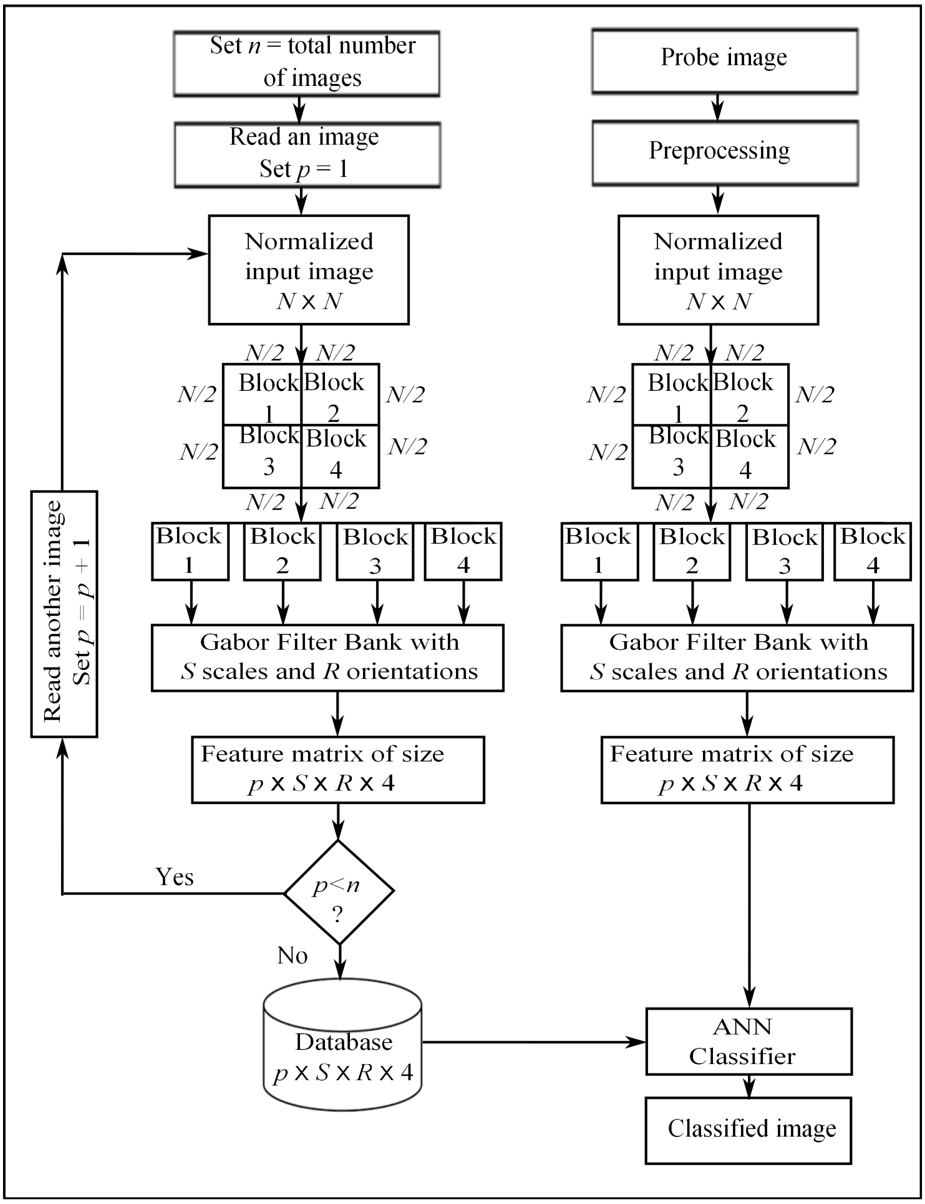

Figure 3.3: Block diagram of the proposed scheme HODR-GFA

3.3 Results and Discussions

The overall simulation has been divided into three different experiments concerning the databases used. In each case, the confusion matrix has been considered for the performance evaluation.

3.3.1 Experiment on ODDB Dataset

Out of 9000 handwritten Odia numerals of the ODDB dataset, 6300 samples are used for training the network, 1350 samples for validation and the rest 1350 samples are in the test set. The size of each sample in the ODDB dataset reduced to 32×32 from its original size of 81×81. Further, it is divided into four blocks of size 16×16. Each of the four blocks of

Table 3.1: Confusion matrix for ODDB dataset

Digit	୦	୧	୨	୩	୪	୫	୬	୭	୮	୯
୦	894	0	1	1	0	0	1	0	1	2
୧	0	891	2	1	3	0	2	0	0	1
୨	2	1	881	0	2	3	4	6	0	1
୩	2	0	1	882	1	4	0	3	2	5
୪	2	0	3	0	886	4	1	3	0	1
୫	1	0	0	2	3	892	0	1	0	1
୬	3	1	5	0	2	1	885	3	0	0
୭	2	1	4	1	2	0	4	886	0	0
୮	0	2	0	0	0	2	1	2	890	3
୯	1	7	1	0	2	0	1	3	2	883

an image passed sequentially through the Gabor filter array with all possible combinations of scales and orientations. The system gives a better accuracy rate in five scales with twelve directions in each scale, i.e., for $4 \times 5 \times 12 = 240$ features. The number of neurons in the hidden layer of the neural network is experimentally determined to be 48 for better convergence characteristics. Hence, the ANN classifier has the structure of $240 - 48 - 10$. The Table 3.1 gives the confusion matrix of the ODDB database. The fraction of samples misclassified is 0.024, i.e., 2.4% error. The performance parameters listed in the Table 3.2

Table 3.2: Confusion values for ODDB dataset

Odia digit	FNR	FPR	TPR	TNR
୦	0.0007	0.0167	0.9833	0.9993
୧	0.0016	0.0175	0.9825	0.9984
୨	0.0044	0.0351	0.9649	0.9956
୩	0.0040	0.0284	0.9716	0.9960
୪	0.0026	0.0234	0.9766	0.9974
୫	0.0027	0.0310	0.9690	0.9973
୬	0.0014	0.0197	0.9803	0.9986
୭	0.0021	0.0248	0.9752	0.9979
୮	0.0038	0.0251	0.9749	0.9962
୯	0.0038	0.0251	0.9749	0.9962

and the performance curve between the cross-entropy and number of epochs taken during testing presented in the Figure 3.4. The receiver operating characteristic (ROC) curve is a plot of true positive rate and the false positive rate at various threshold settings. The Figure 3.5 shows the ROC curves for ten classes of ODDB dataset. An accuracy of 97.6% has been recorded for ODDB dataset.

3.3.2 Experiment on MNIST Dataset

For the supervised learning of MNIST dataset, we utilize 10000 specimens out of 70000 accessible examples to diminish the computational expense. Out of these 7000 samples are chosen for training, 1500 are for validation and rest are used for testing. It gives a good accuracy of 98.8% with five scales and 12 orientations in each scale. The performance

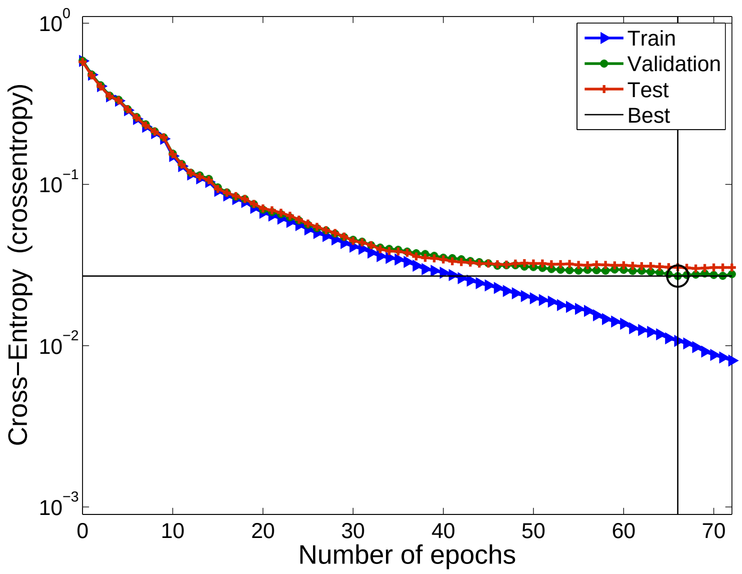

Figure 3.4: Performance curve for ODDB dataset

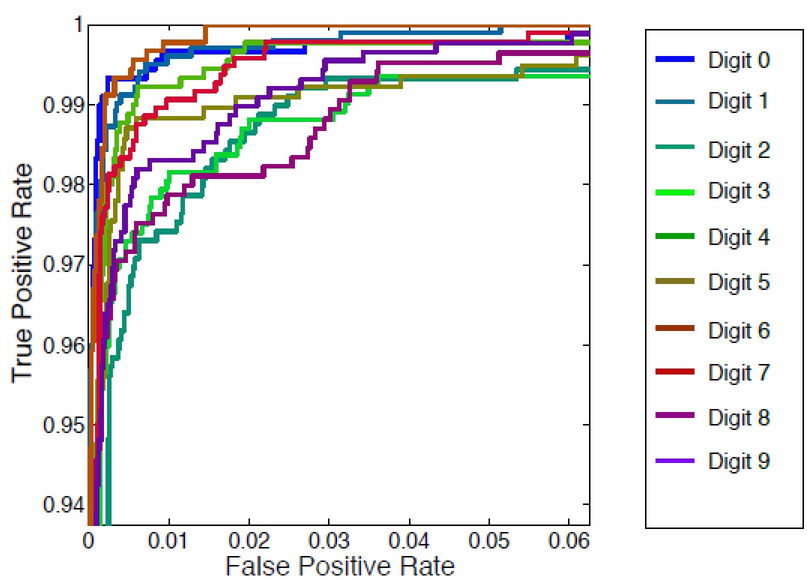

Figure 3.5: ROC curves for ten classes of ODDB dataset

parameters have been recorded in the Table 3.3 and also the performance curve between the cross-entropy and number of epochs presented in Figure 3.6. The ROC curves for ten classes shown in Figure 3.7. The amount of error incurred is about to 0.012 percentage during the classification of the samples of MNIST dataset.

3.3.3 Experiment on USPS Dataset

A similar procedure followed for USPS dataset. With various scales and orientations, it has been observed that it gives a good accuracy of 98.4% with five scales and 12 orientations. The confusion matrix showing the digit misclassification is given in the Table 3.5. It happens

Table 3.3: Confusion matrix for MNIST dataset

Digit	0	1	2	3	4	5	6	7	8	9
0	993	0	0	1	1	1	1	1	1	2
1	0	1114	5	0	2	2	0	0	2	2
2	1	5	969	4	1	3	0	3	5	0
3	2	0	8	1008	0	6	0	4	4	1
4	0	0	4	0	965	1	2	0	2	6
5	1	0	0	5	0	847	5	0	4	0
6	3	1	1	0	3	0	1006	0	0	0
7	0	3	3	3	1	0	0	1057	0	3
8	2	2	2	2	0	4	4	5	922	1
9	0	1	0	1	11	1	0	3	1	960

Table 3.4: Confusion values for MNIST database

English Digit	False Negative rate	False Positive rate	True Positive rate	True Negative rate
0	0.0009	0.0090	0.9910	0.9991
1	0.0015	0.0107	0.9893	0.9985
2	0.0024	0.0232	0.9768	0.9976
3	0.0028	0.0156	0.9844	0.9972
4	0.0017	0.0193	0.9807	0.9983
5	0.0016	0.0208	0.9792	0.9984
6	0.0009	0.0118	0.9882	0.9991
7	0.0015	0.0149	0.9851	0.9985
8	0.0024	0.0202	0.9798	0.9976
9	0.0020	0.0154	0.9846	0.9980

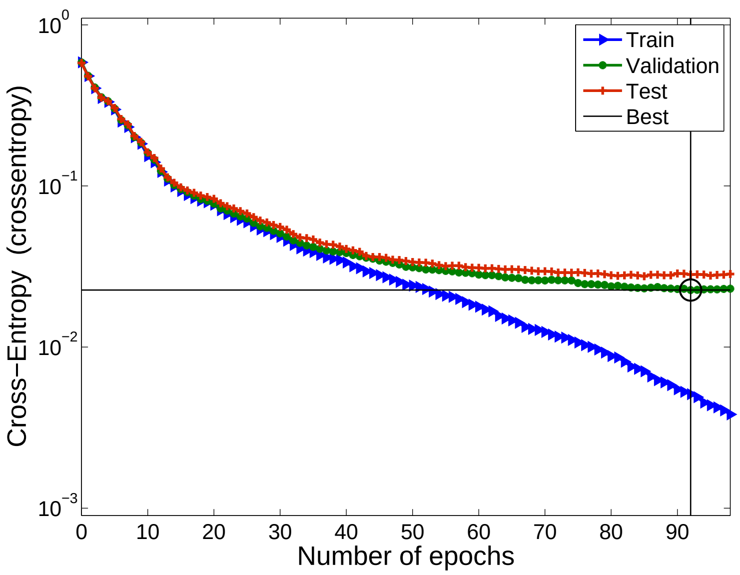

Figure 3.6: Performance curve for MNIST dataset

due to the structural similarity between the digits. The performance parameters recorded in the Table 3.6 used to evaluate the proposed scheme over USPS dataset. The performance curve has been displayed in Figure 3.8 and the ROC for all ten classes has been shown in Figure 3.9. The fraction of samples misclassified has been recorded as 0.016 means 1.6%

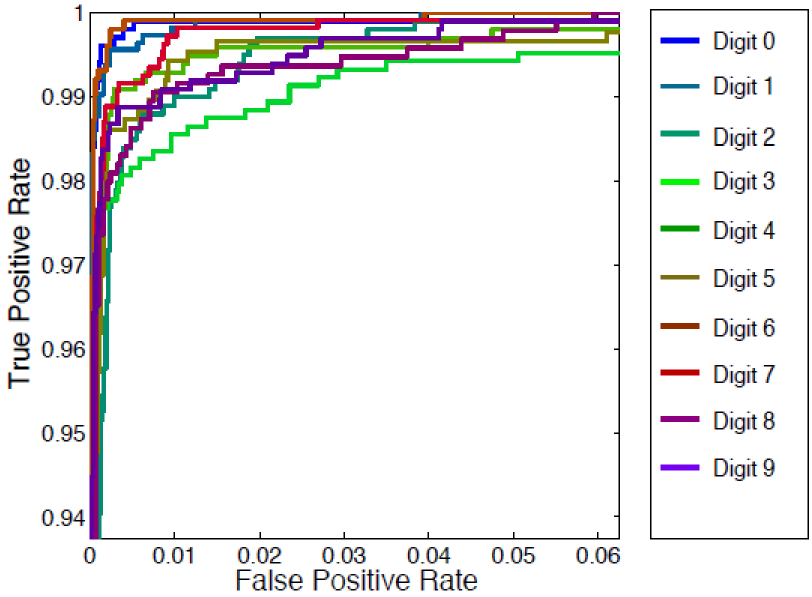

Figure 3.7: ROC curves for ten classes of MNIST dataset

Table 3.5: Confusion matrix for USPS dataset

Digit	0	1	2	3	4	5	6	7	8	9
0	1545	0	3	0	1	0	1	1	1	1
1	0	1261	2	0	1	0	3	2	0	0
2	1	1	919	0	4	0	1	3	0	0
3	1	1	4	809	1	7	0	1	0	0
4	0	4	2	0	840	0	1	1	0	4
5	4	0	1	2	0	703	3	1	2	0
6	0	1	0	0	0	0	832	0	1	0
7	1	0	7	1	6	0	0	774	0	3
8	2	1	6	4	0	2	1	1	691	0
9	0	1	0	1	4	0	0	3	1	811

error. For all three datasets, an accuracy comparison has been made with different scales and orientations and are listed in the Table 3.7. In general, the suggested Gabor feature is found out to be a better feature for handwritten Odia digit as well as English numeral recognition.

Table 3.6: Confusion values for USPS database

English Digit	False Negative rate	False Positive rate	True Positive rate	True Negative rate
0	0.0010	0.0058	0.9942	0.9990
1	0.0010	0.0071	0.9929	0.9990
2	0.0012	0.0265	0.9735	0.9988
3	0.0018	0.0098	0.9902	0.9982
4	0.0014	0.0198	0.9802	0.9986
5	0.0015	0.0126	0.9874	0.9985
6	0.0002	0.0119	0.9881	0.9998
7	0.0021	0.0165	0.9835	0.9979
8	0.0020	0.0072	0.9928	0.9980
9	0.0012	0.0098	0.9902	0.9988

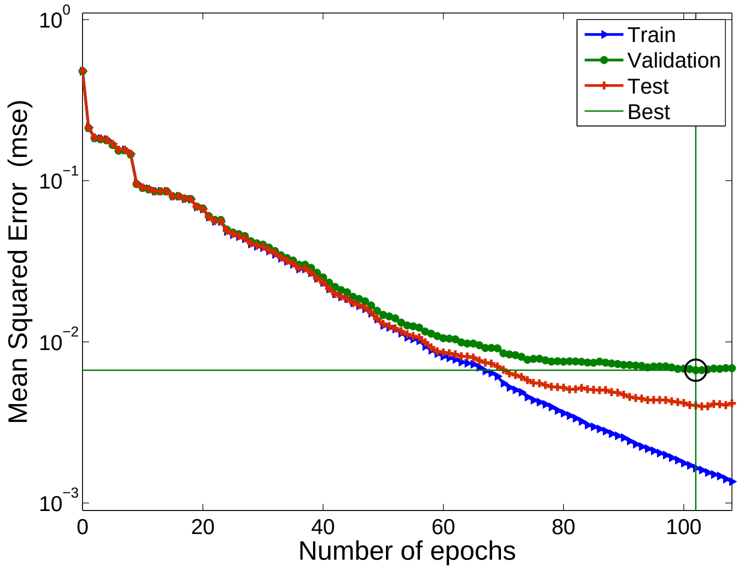

Figure 3.8: Performance curve of USPS datset

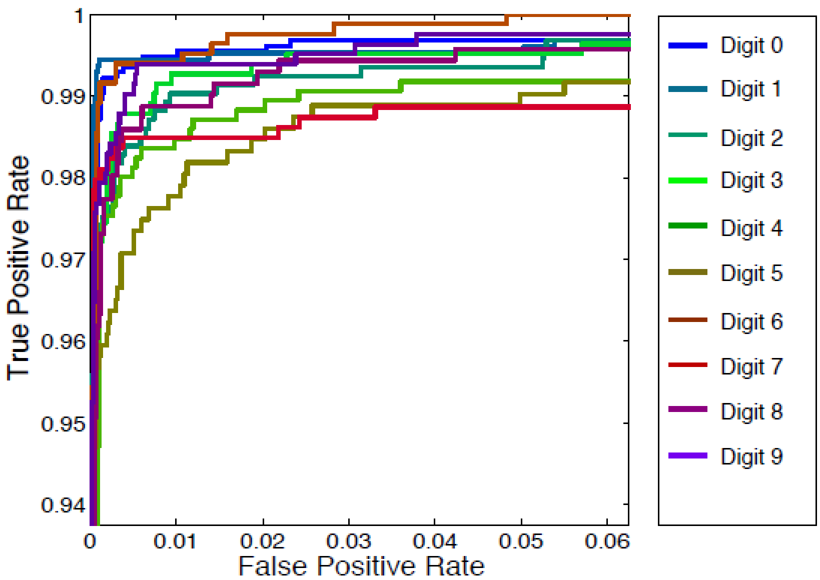

Figure 3.9: ROC curves for USPS dataset

Table 3.7: Accuracy of the scheme with various scale and orientation on different datasets

Dataset	Number of scales S and number of orientations R					
	(4,8)	(4,12)	(5,8)	(5,12)	(6,8)	(6,12)
MNIST	92.1%	93.4%	97.3%	**98.8%**	98.2%	98.5%
USPS	94.6%	95.6%	96.6%	**98.4%**	97.9%	98.1%
ODDB	93.5%	94.8%	95.5%	**97.6%**	96.2%	95.9%

3.4 An application: Automatic Evaluation of Simple Odia Handwritten Arithmetic Expression

To demonstrate the validity of our proposed handwritten digit recognition using Gabor feature and neural classifier, we have suggested a **G**raphical **U**ser **I**nterface(GUI) model for the **A**utomatic **E**valuation of **S**imple **O**dia **H**andwritten **A**rithmetic **E**xpression (AESOHAE). The expression involves only integers and real constants with basic arithmetic operations. The GUI based application not only converts the scanned expression into its similar English format but also evaluates the expression. Initially, the scanner is used to get the digitized version of the page containing the mathematical expression. Then it is fed to the system, where all preprocessing modules are exercised to get the expression in the required form. A total of twenty symbols are considered that includes ten digits (zero to nine), seven basic arithmetic operators (‘unary +’, ‘unary -’,‘+’, ‘-’, ‘*’, ‘/’, ‘%’), precision (‘.’), left (‘(’), and right (‘)’) parenthesis which are listed in the Table 3.8. For the identification of Odia digit

Table 3.8: List of operators and operands along with their meaning and corresponding symbol in Odia language

Symbol	Meaning	In Odia
+	Unary Plus	+
-	Unary Minus	-
+	Addition	+
-	Subtraction	-
*	Multiplication	$\times$
/	Division	$\div$
%	Modulo	%
(	Left Parenthesis	(
)	Left Parenthesis	)
.	Precision	.

Digit	Meaning	In Odia
0	Zero	୦
1	One	୧
2	Two	୨
3	Three	୩
4	Four	୪
5	Five	୫
6	Six	୬
7	Seven	୭
8	Eight	୮
9	Nine	୯

and basic arithmetic operators/symbols the Gabor features are taken into consideration along with back-propagation neural network (BPNN) for recognition. To validate the proposed model five hundred sample expressions have been collected from various sources. It has been observed that out of those only 449 sample expressions correctly evaluated. At this point, the accuracy of our system recorded as 89.80%. The complete process is delineated in Figure 3.10 considering a sample test image. The steps followed in due course of evaluating a handwritten Odia arithmetic expression are given below.

Data Collection

For data collection, all samples from ODDB digit database have been taken into account. Due to the unavailability of any standard database for handwritten Odia arithmetic operators, samples collected from 200 different informants belonging to various groups and ages. A formatted sheet prepared with all distinct eighteen symbols printed and leaving a blank box just underneath every character. A total of 3600 patterns, i.e., 200 specimens for every symbol have been collected and put away in our local system. In this way, the database has 11000 digit samples and 1600 Odia arithmetic operator samples. It has been named as modified-ODDB database. All images are preprocessed and normalized to a size of 20×20.

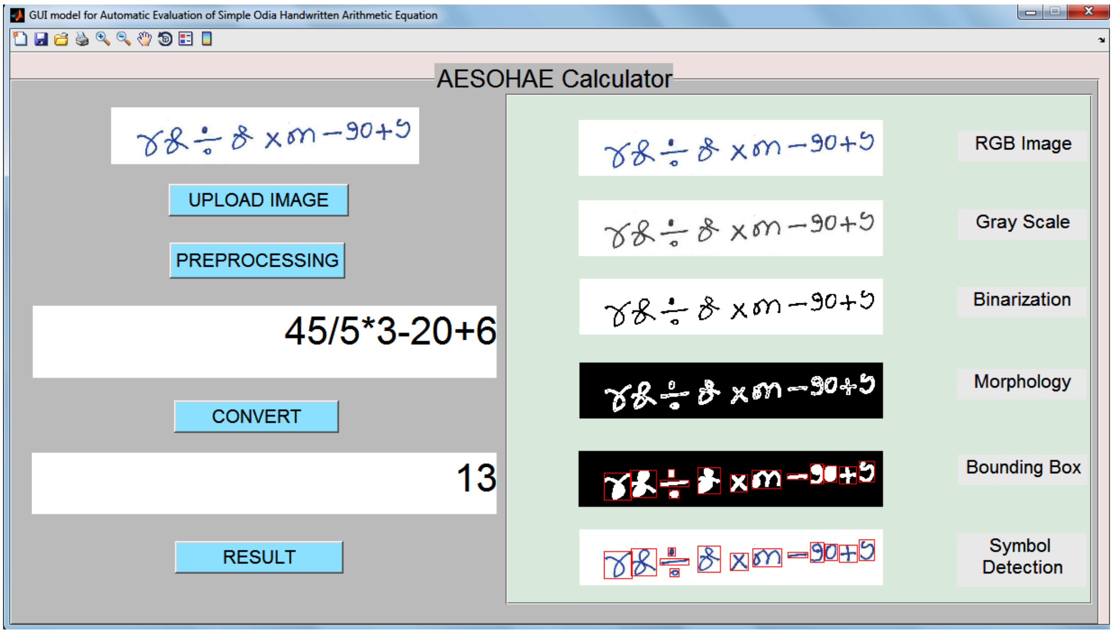

Figure 3.10: Evaluation of a sample expression

Experimental Set-up

The experiment was carried out on a system with the following configuration: Intel® core™ $i7$4770CPU @ 3.40GHz Dual Core Processor, 64 bit Operating System, and 6GB RAM. Being a potential tool for non-linear classifications, Artificial neural network (ANN) has been utilized to classify the symbols. The conventional back propagation has been used to train the network, and the number of hidden neurons are chosen experimentally. The modified-ODDB dataset divided into three sets namely train set with 70% of the total samples and rest are divided equally among the validation and test sets. For faster training, scaled conjugate gradient method has been employed. The confusion matrix obtained is shown in Table 3.9. Initially a sample expression is uploaded which then passed through

Table 3.9: Confusion matrix for modified-ODDB dataset

Digit	୦	୧	୨	୩	୪	୫	୬	୭	୮	୯	+	−	×	÷	%	(	)	.
୦	1091	1	2	0	3	0	1	1	0	0	0	0	0	0	0	0	0	2
୧	2	1083	3	0	0	0	5	4	0	0	0	0	0	0	0	3	0	0
୨	0	0	1094	1	0	0	2	0	0	0	0	0	0	0	3	0	0	0
୩	2	0	6	1078	5	3	0	2	4	0	0	0	0	0	0	0	0	0
୪	0	0	4	0	1089	1	2	0	2	2	0	0	0	0	0	0	0	0
୫	1	2	0	5	0	1089	2	1	0	0	0	0	0	0	0	0	0	0
୬	2	9	0	0	2	0	1081	6	0	0	0	0	0	0	0	0	0	0
୭	0	13	1	0	1	0	10	1175	0	0	0	0	0	0	0	0	0	0
୮	2	0	2	0	5	7	2	3	1078	1	0	0	0	0	0	0	0	0
୯	3	7	2	0	2	1	3	2	3	1087	0	0	0	0	0	0	0	0
+	0	0	0	0	0	0	0	0	0	0	190	0	4	6	0	0	0	0
−	0	0	0	0	0	0	0	0	0	0	0	197	0	3	0	0	0	0
×	0	0	0	0	0	0	0	0	0	0	0	0	189	1	0	0	0	0
÷	0	0	0	0	0	0	0	0	0	0	7	0	3	187	3	0	0	0
%	1	1	0	0	0	3	1	0	0	6	0	0	0	5	183	0	0	0
(	0	5	0	0	0	0	0	0	0	4	0	0	0	0	0	191	0	0
)	0	0	4	0	0	0	3	4	0	0	0	0	0	0	0	2	187	0
.	2	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	196

several preprocessing phases including binarization using Otsu's method [115, 116], digit and operator identification utilizing bounding box technique [12]. All sub-images are normalized to a size of 20×20. The Gabor features are extracted from each symbol and fed to the back propagation neural network for recognition.

3.5 Summary

For feature extraction of handwritten Odia digit, an array of Gabor filters with various scales and orientations have been utilized followed by a back-propagation neural network classifier for recognizing the digit. An accuracy of 97.6% has been recorded. Simulation results demonstrate that Gabor filter at five scales and 12 orientations in each scale give higher accuracy in ODDB dataset. Similar observations have also been made for MNIST and USPS dataset. To demonstrate the proposed scheme, a GUI model has been developed for the evaluation of simple arithmetic expression written in Odia language. The proposed system has been tested on five hundred sample Odia handwritten expressions collected from different individuals. Out of these four hundred forty-nine samples have been evaluated correctly. The bottleneck of the proposed model is that it operates on integers, real constants, and basic arithmetic operations.

Chapter 4

Handwritten Odia Character Recognition using Discrete Orthogonal S-Transform (HOCR-DOST)

Researchers have given more thrust to develop OCR for the most popular *Indic* languages such as Devanagari, Bangla, Tamil, and Telugu. However, it has been observed that the development of OCR for Odia language is at its incipient days. Odia is the mother tongue of the state Odisha. There are no OCR systems available commercially that makes digitized printed or handwritten Odia document searchable. Development of OCR for Odia language is necessary because it has many regional based applications such as in banking, offices, legal industries, health-care, education, finance, and government agencies. With the assistance of OCR for Odia dialect, individuals no more need to retype important archives physically when entering them into electronic databases. Recognition of handwritten Odia character is currently one of the active areas of research as far as *Indic* languages are concerned. The main problem arises since we are doing it for handwritten text. A lot of variations of the writing style of different individuals makes the recognition process challenging. Looking into these difficulties, we reason out that feature extraction is indeed a critical phase of OCR. Multi-resolution techniques are suitable for extracting meaningful features from an image because it gives information in both time and frequency. In this chapter, a multi-resolution technique, i.e., Two-dimensional Discrete Orthogonal Stockwell Transform (2D-DOST) [117] has been used to extract the coefficients of the basis functions from each Odia character image. Further, a feature reduction technique has been applied to pick out the most significant information. Multiple back-propagation neural networks have been designed in the due process of recognizing Odia character. In the first phase of the work the characters are divided into two groups, i.e., *Group I* and *Group II* by looking for a vertical bar at the end of a character. In Odia character set, there are only 20 such characters having a vertical line toward the end.

The chapter has been adumbrated as follows. Section 4.1 describes the method of extracting the 2D-DOST coefficients for a character. The proposed scheme has been delineated in Section 4.2. The experimental results obtained on the OHCS database have been presented in Section 4.3.

4.1 Feature Extraction using 2D-DOST

The discrete orthogonal S-transform [118] is the variation of Stockwell Transform (ST) recommended by R. G. Stockwell in the year 1996. The redundant information in the resulting time-frequency representation makes the Stockwell transform computationally hard when a huge information needs to be processed. Hence, to overcome the difficulty, the discrete orthonormal Stockwell transform (DOST) has been proposed. The DOST based on a set of orthonormal basis functions that localize the Fourier spectrum of the signal [119]. It has already been applied successfully in various fields of research such as image restoration, texture analysis, and pattern recognition [117]. A brief description on Stockwell Transform is given below.

4.1.1 Stockwell-Transform (ST)

The Stockwell transform is a hybrid of the short-time Fourier transform and the wavelet transform. It provides a time-frequency representation of a signal. The Stockwell transform of a continuous 1D signal $h(t)$ is defined as,

$$s(\tau, f) = \int_{-\infty}^{\infty} h(t) \frac{|f|}{\sqrt{2\pi}} e^{-(\tau-t)^2 f^2/2} e^{-i2\pi f t} dt. \tag{4.1}$$

where f is the frequency variable, t is the time variable, and τ is time translation. It gives a full time decomposition of a signal. Though it is similar to a continuous wavelet transform (CWT) but unlike the wavelet transform the S-transform retains absolutely referenced phase information. In addition, the ST can be expressed in the Fourier domain as,

$$s(\tau, f) = \int_{-\infty}^{\infty} H(\psi + f) e^{-\frac{(2\pi\psi)^2}{2f^2}} e^{i2\pi\psi\tau} d\psi \tag{4.2}$$

where $H(f)$ is the Fourier spectrum of $h(t)$. The discretization of (4.2) leads to the discrete Stockwell transform (DST) and is given by,

$$S[k, n] = \sum_{m=0}^{N-1} e^{-\frac{2\pi^2 m^2}{n^2}} H[m+n] e^{\frac{i2\pi mk}{N}} \tag{4.3}$$

where k is the index of time translation and n is the index of frequency shift with $n \neq 0$. Here $H(.)$ is the DFT of $h(.)$. The S-Transform of a discrete 2-dimensional signal $f(x, y)$ is given by,

$$S(x, y, p_x, p_y) = \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} F(k + p_x, l + p_y) e^{-2\pi^2 \left(\frac{k^2}{P_x{}^2} + \frac{l^2}{P_y{}^2} \right)} e^{2\pi i (kx + ly)} \tag{4.4}$$

4.1.2 Discrete Orthogonal S-transform and its Properties

The phase of the S-transform referenced to the time origin provides useful and supplementary information about spectra. However, the major limitation of S-transform is its high time and space complexity due to its redundant nature. The 2D-DST of an image of size $N \times N$ has a computational complexity of $O(N^4 + N^4 log(N))$ and storage requirements of $O(N^4)$. Due to high complexity and redundant information of S-transform, it is not frequently used in many applications. Due to its exponential computational overhead, Wang et al. [120] in the year 2009 has suggested a faster version of the S-transform, namely, discrete orthonormal Stockwell transform (DOST). It provides a spatial frequency representation of an image, with computational and storage complexity as $O(N^2 + N^2 log(N))$ and $O(N^2)$ respectively. A six order DOST is shown in Figure 4.1. The square indicates the sub-image of each order.

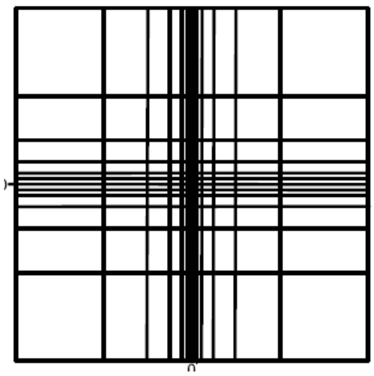

Figure 4.1: A six order DOST. The square indicates the sub-image for each order.

The 2D-DOST is based on a set of orthonormal basis functions that localize the Fourier spectrum of the signal. The time-frequency representation given by the Stockwell transform with zero information redundancy is sampled and the phase properties of the ST are retained. However, DOST can be used to represent data efficiently, which is the orthonormal version of the DST, producing N point time-frequency representation for a signal of length N. Thus, DOST gives features with zero information redundancy. The following features of DOST have made it useful for various applications.

- Absolute Referenced Phase Information
- Fourier Transform can be calculated from DST.
- Linearity

The process of evaluating DOST coefficients of a 2D image in the frequency domain using a dyadic sampling scheme is given in the Algorithm 1.

Algorithm 1: DOST Feature Extraction

Input: A normalized discrete image $img[x, y]$ of size $N \times N$.

Output: One dimensional N^2 number of DOST coefficients.

Considering the sampling interval of one in the x-and y-direction calculate the forward 2D-Fourier Transform of the img by the following equation $F[m, n] = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} img[x, y]exp^{-2\pi i(\frac{mx}{N}+\frac{ny}{N})}$.

The 2D-DOST of $N \times N$ image $img[x, y]$ is calculated by partitioning the 2D-FT of the image, $F[m, n]$, multiplying by the square root of the number of points in the partition. $D[x', y', v_x, v_y] = \frac{1}{\sqrt{2^{(p_x+p_y-2)}}} \sum_{m=-2^{p_x-2}}^{2^{p_x-2}-1} \sum_{m=-2^{p_y-2}}^{2^{p_y-2}-1} F[m + v_x, n + v_y]exp^{2\pi i\left(\frac{mx'}{2^{p_x-1}}+\frac{ny'}{2_y^p-1}\right)}$, where $v_x = 2^{p_x-1} + 2^{p_x-2}$ and $v_y = 2^{p_y-1} + 2^{p_y-2}$ are the horizontal and vertical voice frequencies.

The value of (p_x, p_y) at image location (x, y) can be found at $D[x/N \times 2^{p_x-1}, y/N \times 2^{p_y-1}, v_x, v_y]$.

The array D is reshaped to one dimensional array of size N^2.

return D

4.2 Proposed HOCR-DOST Scheme

In this chapter, a scheme has been proposed for the recognition of Odia handwritten character utilizing DOST features. The overall block diagram of our proposed scheme is shown in Figure 4.2. The normalized database OHCS called the Odia handwritten character set had been used in our experiment. The whole dataset has been divided into two groups such as *Group I* and *Group II* through chain code histogram. In *Group I* all the characters with a vertical bar at the end are present, whereas others are stored in *Group II*. Each sample is size normalized to 32×32. The DOST feature vector of length 1024 is extracted from each sample. The process continues for all images in the training set. Further, the dimension of the feature matrix is reduced for faster processing. Recognition of any new character is accomplished by training two neural networks for each group. The detail description of each phase of our proposed model is explained briefly here in a sequel.

4.2.1 Pre-processing

Standard methods like binarization, noise reduction, and character segmentation have been used to get the segmented isolated character image from the scanned document. The orientation of the segmented character has been measured, and the slant correction has been made accordingly. The image thus obtained is normalized to a size of 32×32 image. Wiener filter [106] has been applied for noise removal. The thinning process has been applied to each image so as to convert it into an image of a single pixel width. The chain code

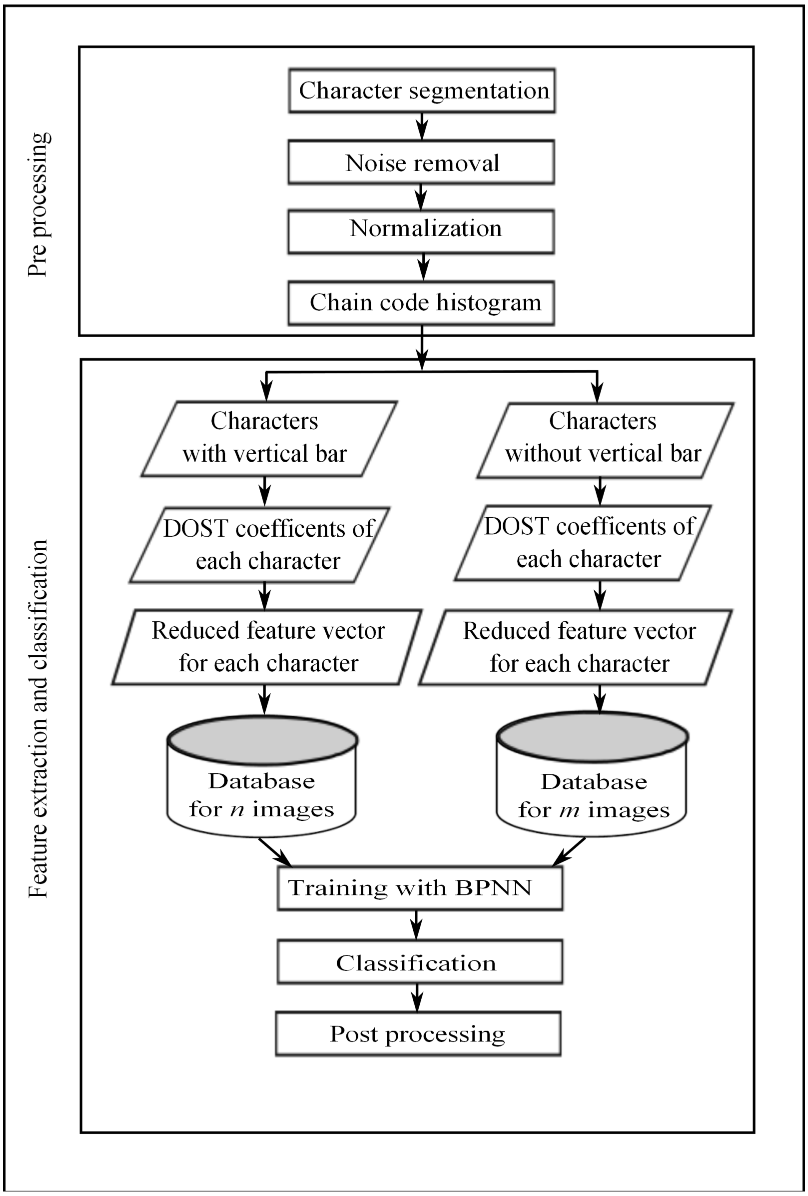

Figure 4.2: Block Diagram of Proposed HOCR-DOST Scheme.

histogram for each image is plotted which computes the number of pixels in each column. It has been observed that a character with a vertical line has a vertical bar in the histogram. Thus, it becomes easy to classify the whole character set into two groups as shown in Figure 4.3. Existing recognizers for, e.g., English, Chinese, Japanese, and numbers have achieved reasonable processing rates based on small sets of symbols. In Asian languages there are many "characters", but only small sets of strokes. When the lexicon increases, it is challenging to achieve high accuracy and speed. One goal is to reduce the amount of computation for recognition when a large number of symbols is used. One solution is to group these symbols into classes. Unknown characters would first be placed into a group, followed by recognition within the group instead of comparing with the whole set of samples. The present work is to identify those characteristics that may be used to separate characters into classes effectively and design of neural network for recognizing the

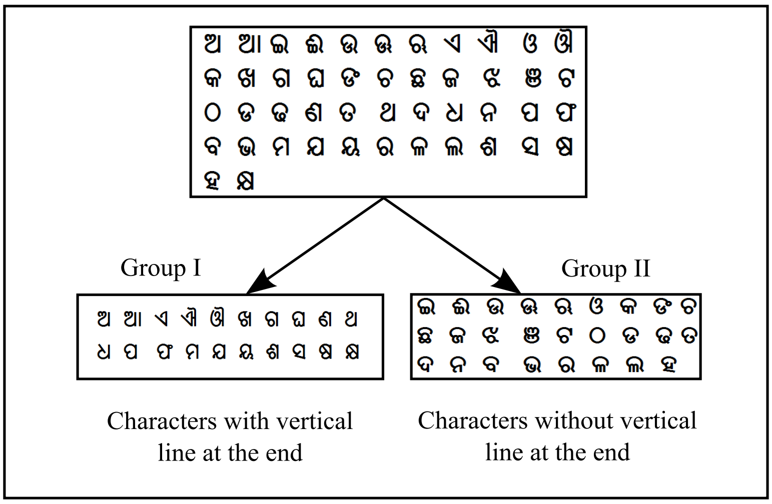

Figure 4.3: Division of basic Odia characters in two groups.

character encountered. The following method is used in this regard. Each character, we generate the chain code histogram vector (CCHV) of length 32. As a whole we have 14100 samples, hence at the end, the size of chain code histogram matrix (CCHM) M would be 14100×32. We have considered level matrix L of size 14100×2 having two levels 1 for *Group I* characters and 2 for *Group II* characters. The following MATLAB code fragment is used to design the network "*net*" that helps to assign a character to either *Group I* or *Group II*.

$$load \quad M.mat;$$
$$load \quad L.mat;$$
$$x = M';$$
$$t = L';$$
$$net = patternnet(10);$$
$$net = train(net, x, t);$$
$$y = net(x);$$
$$perf = perform(net, t, y);$$
$$classes = vec2ind(y);$$
$$plotconfusion = (t, y);$$

The confusion matrix is given in the Table 4.1. The threshold value chosen to be fifteen

Table 4.1: Confusion matrix while grouping the characters

	Group I	Group II
Group I	5993	7
Group II	7	8093

for the number of pixels in a column to ensure that the character has a vertical line toward the end. Two characters have been considered, i.e., ଖ ('Kha') from *Group I* and ଝ ('Jha') from *Group II* as shown in Figures 4.4 and 4.5 respectively. The chain code histogram are also shown in Figures 4.6 and 4.7 respectively. It is evident that the letter ଖ ('Kha') has a vertical bar in the histogram having 28 pixels hence, it belongs to *Group I*. Similarly, the letter ଝ ('Jha') has less than ten pixels hence, it is assigned to *Group II*.

Figure 4.4: Odia character 'ଖ'.

Figure 4.5: Odia character 'ଝ'.

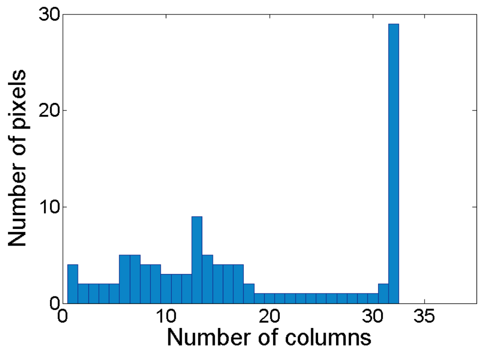

Figure 4.6: CCH for Odia character 'ଖ'.

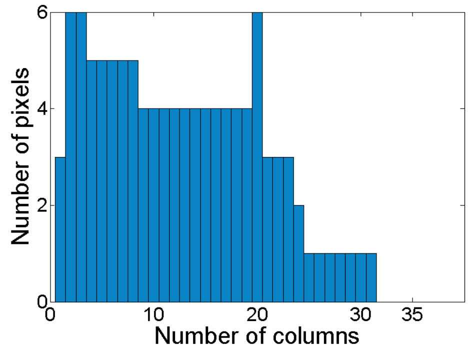

Figure 4.7: CCH for Odia character 'ଝ'.

4.2.2 DOST Coefficient Feature Evaluation

Within the bandwidth of $2^{p_x-1} \times 2^{p_y-1}$ frequencies, the DOST gives information about the voice frequencies (v_x, v_y). It uses the minimum number of points required to describe the amplitude of a Fourier frequency component in each of the horizontal and vertical directions. If we represent the voice frequencies as a complex image where, v_x corresponds to real and v_y corresponds to imaginary part then the magnitude and phase angle is obtained by $M_v = \sqrt{v_x^2 + v_y^2}$ and $\theta_v = tan^{-1}(\frac{v_y}{v_x})$ respectively. The magnitude on a log scale of each

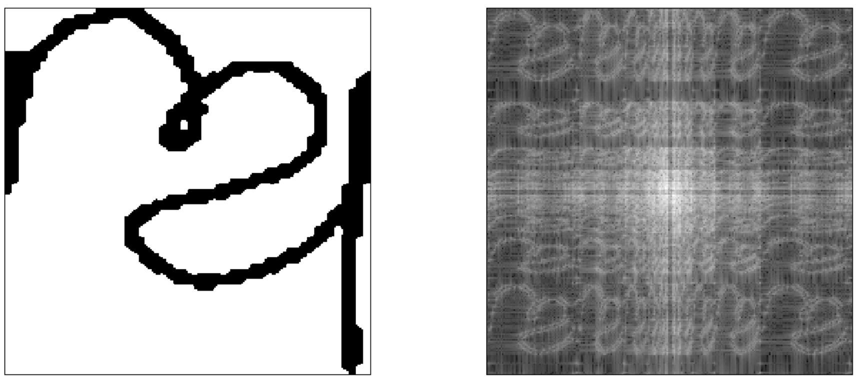

Figure 4.8: DOST (right) of the first letter 'ଅ' in Odia script (left) of size 32×32 where the brightness represents the magnitude of each coefficient on a log scale.

DOST coefficients for the normalized image ଅ ('a') of size 32×32 is shown in Figure

4.8. Thus, each sample generates 1024 DOST coefficients where the size of the image is 32×32. So, the size of feature matrix for m samples is $m \times 1024$. The recognition accuracy is adversely affected by high feature dimensionality of any datasets so, using Principal Component Analysis (PCA) (discussed in Section 4.2.3) the feature matrix size has been reduced.

4.2.3 Feature Reduction

The Principal Component Analysis (PCA) has been utilized to reduce the dimension of the DOST feature matrix. PCA transforms a number of (possibly) correlated variables into a (smaller) number of uncorrelated variables called principal components [122]. The detail of PCA based dimensionality reduction has given in Algorithm 2. In the suggested scheme, the original feature matrix of size $m \times 1024$ has been reduced to $m \times 256$.

Algorithm 2: PCA based feature reduction

Input: An original dataset D of size $m \times n$, where m denotes the total number of samples and n denotes the number of features in each sample.

Output: The reduced dataset D' of size $m \times k$, where $k \leq n$.

1 Compute the means for every dimension of the whole dataset to get the n-dimensional mean vector.

2 Find the covariance matrix C, eigenvectors and eigenvalues $([v, d] = eig(D))$ of the whole dataset so that $C * v = v * d$, where d is the diagonal matrix of eigenvalues and v is a matrix whose columns are the corresponding eigenvectors

3 Sort the eigenvectors by decreasing eigenvalues and choose k eigenvectors with the largest eigenvalues to form a $n \times k$ dimensional matrix W (where every column represents an eigenvector).

4 Use this $n \times k$ eigenvector matrix to transform the samples onto the new subspace. $D' = D \times W$

5 **return** D'

4.3 Result and Discussion

Artificial neural network (ANN) being the potential tool for non-linear classification, it has been utilized to classify the characters in each group. We employ two different ANN for two groups of characters. In each case, conventional back propagation is used to train the network, and the number of hidden neurons is chosen experimentally. To validate the proposed scheme, simulation has been carried out on OHCS dataset, and other datasets using 70% of the data for training and rest 30% is divided equally into two parts for validation and testing. For faster training, scaled conjugate gradient method is used. The training of the neural network stops when either the gradient reaches a value of 10^{-6} or the number of consecutive validation checks reaches 6, i.e., there is no better improvement of the gradient for six continuous iterations. The performance of the classification is evaluated with the help of confusion matrix. It is a table that shows the predicted and actual class accomplished by the classifier.

Table 4.2: Performance parameters of Group I characters

BOC	FNR	FPR	TNR	TPR	BOC	FNR	FPR	TNR	TPR
ଅ	0	0.0099	0.9901	1.0000	ଧ	0.0005	0.0198	0.9802	0.9995
ଆ	0.0005	0	1.0000	0.9995	ପ	0.0014	0.0102	0.9898	0.9986
ଏ	0	0.0066	0.9934	1.0000	ଫ	0.0005	0.0067	0.9933	0.9995
ଐ	0.0005	0.0198	0.9802	0.9995	ମ	0.0019	0.0103	0.9897	0.9981
ଓ	0.0005	0.0067	0.9933	0.9995	ଯ	0.0011	0.0516	0.9484	0.9989
ଖ	0.0016	0	1.0000	0.9984	ଝ	0.0004	0.0197	0.9803	0.9996
ଗ	0.0011	0.0726	0.9274	0.9989	ଶ	0.0040	0.0072	0.9928	0.9960
ଘ	0.0005	0.0166	0.9834	0.9995	ଷ	0.0004	0.0230	0.9770	0.9996
ଣ	0.0014	0.0426	0.9574	0.9986	ସ	0.0019	0.0069	0.9931	0.9981
ଥ	0.0005	0	1.0000	0.9995	କ୍ଷ	0.0005	0.0294	0.9706	0.9995

Experimental setup for Group I characters

A multilayer perceptron of size $256 - 75 - 20$ was used. The input and output layer size are based on the dimension of features and output classes respectively, while for the hidden layer, the accuracy at different sizes were compared. Experimentally best result is recorded with 75 neurons in the hidden layer. The entire feature set of *Group I* character (a total of 6000 samples) sets has been divided into the following sets,

- 4200 samples are used for training.
- 900 samples are used for testing.
- 900 samples are used for validation.

In Table 4.2, we illustrate different performance measures for all samples of Basic Odia Character (BOC) in *Group I*. It has been observed from the confusion matrix for *Group I* characters given in Table 4.3 that almost all characters are classified correctly, however, the Odia character 'ଣ' is wrongly classified as 'ଖ' 3 times, 'ଗ' 3 times, 'ଶ' 3 times and as 'ପ' 2 times. It happens due to the structural similarity between the characters. The performance of the network is shown in Figure 4.9 and it is observed that the best validation performance is 0.01763 at 47 epochs. The convergence characteristic is shown in Figure 4.10. Receiver Operating Characteristic curve (ROC) is given in Figure 4.11. It is a plot of the true positive rate against the false positive rate for the different possible cut points of a test. This curve indicates an increase in sensitivity and a decrease in specificity. The accuracy of 98.9% is recorded for *Group I* characters.

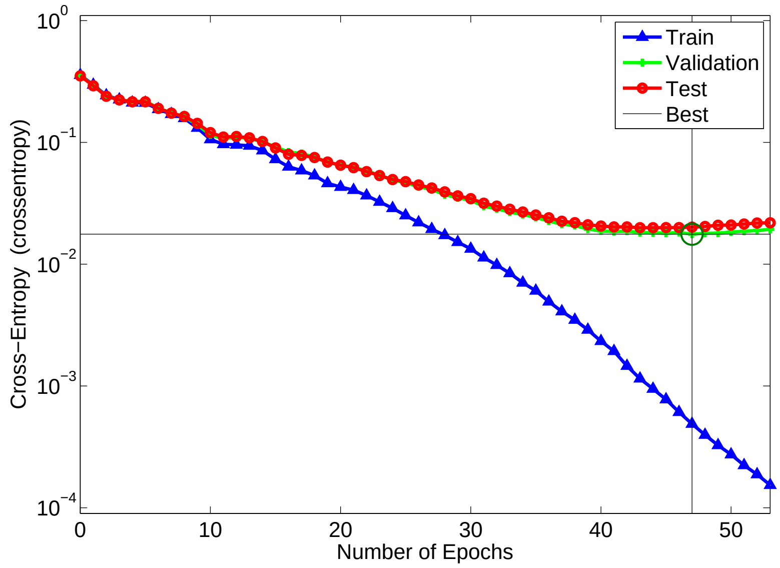

Figure 4.9: Performance of the network for Group I characters

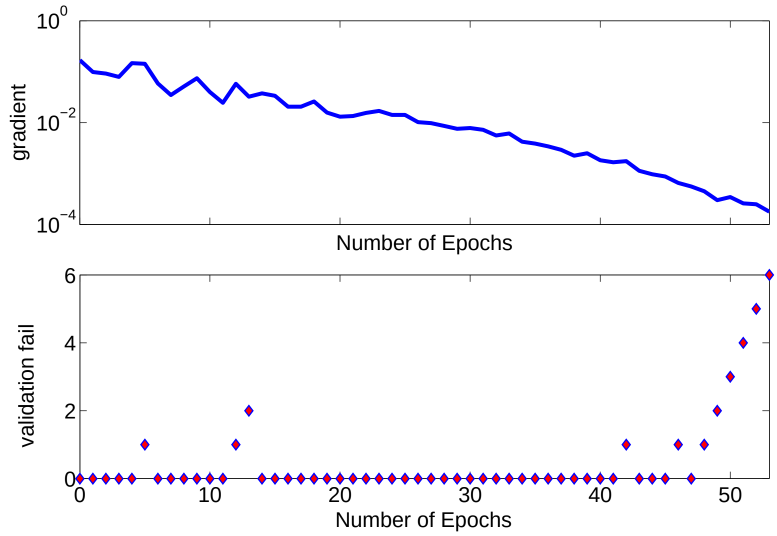

Figure 4.10: Convergence states of Group I characters

Experimental Setup for Group II Characters

Another multilayer perceptron of size $256 - 90 - 27$ was used for classifying *Group II* characters. Experimentally best result comes when the number of hidden neurons is 90. The entire feature set of *Group II* characters (a total of 8,100 samples) sets are divided into the following subsets:

- 5670 samples are used for training.
- 1215 samples are used for testing.

Table 4.3: Confusion matrix for Group I characters

	ଅ	ଆ	ଏ	ଐ	ଔ	ଖ	ଗ	ଘ	ଣ	ଥ	ଧ	ପ	ଫ	ମ	ଯ	ଝ	ଶ	ଷ	ସ	କ୍ଷ
ଅ	300	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ଆ	0	300	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ଏ	0	0	300	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ଐ	0	0	0	291	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3
ଔ	0	0	0	3	297	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ଖ	0	0	0	0	0	297	0	0	0	0	3	0	0	0	0	0	0	0	0	0
ଗ	0	0	0	0	0	0	294	3	0	0	0	3	0	0	0	0	0	0	0	0
ଘ	0	0	0	0	0	0	0	297	0	0	0	0	0	0	0	0	0	3	0	0
ଣ	0	0	0	0	0	3	3	0	289	0	0	3	0	0	0	0	2	0	0	0
ଥ	0	0	0	0	0	0	0	0	0	300	0	0	0	0	0	0	0	0	0	0
ଧ	0	0	0	0	0	0	0	0	0	0	300	0	0	0	0	0	0	0	0	0
ପ	0	0	0	0	0	0	0	0	0	0	0	297	0	0	0	0	0	0	3	0
ଫ	0	0	0	0	0	0	0	0	0	0	0	0	300	0	0	0	0	0	0	0
ମ	0	0	0	0	0	2	0	0	0	0	0	0	0	296	0	0	2	0	0	0
ଯ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	300	0	0	0	0	0
ଝ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	295	0	0	2	3
ଶ	0	0	0	0	0	2	0	3	0	0	0	0	0	0	0	0	295	0	0	0
ଷ	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	297	0	0
ସ	0	2	0	0	0	0	2	0	0	0	0	0	0	0	0	2	0	0	294	0
କ୍ଷ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	1	0	296

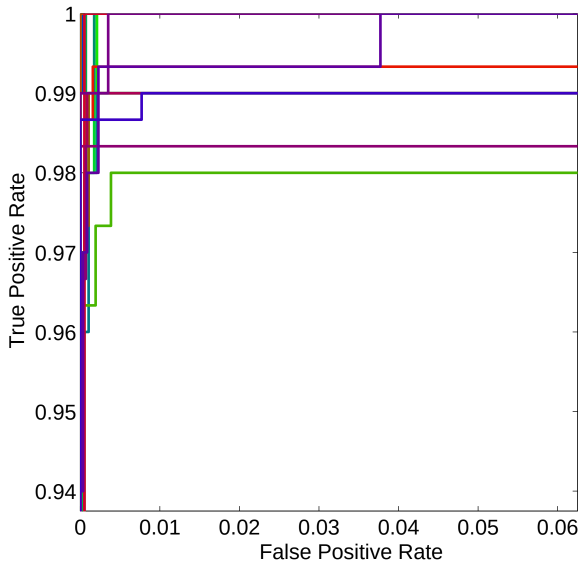

Figure 4.11: ROC curves of Group I characters

- 1215 samples are used for validation.

In Table 4.4 illustrates the false negative rate, false positive rate, true negative rate, and true positive rate for all samples of basic Odia character in *Group II*. The confusion matrix for *Group II* has been portrayed in Table 4.6. It has been noticed that all characters are classified correctly, however the Odia character 'ନ' is wrongly classified as 'ଲ' 10 times, and as 'ଳ' 6 times. It happens due to the structural similarity between the characters. The performance of the network is shown in Figure 4.12 and it is observed that the best validation performance is 0.020167 at 65 epochs. The ROC for *Group II* characters is depicted in Figure 4.14. The convergence characteristics are also shown in Figure 4.13. The accuracy recorded as 98.2% for *Group II* characters. The average overall accuracy of our scheme is recorded as 98.55%. A sample character correctly recognized by the proposed HOCR-DOST scheme and one

Table 4.4: Performance parameters of Group II characters

BOC	FNR	FPR	TNR	TPR	BOC	FNR	FPR	TNR	TPR
ଇ	0	0.0354	0.9646	1.0000	ଠ	0.0003	0	1.0000	0.9997
ଈ	0.0004	0.0230	0.9770	0.9996	ଡ	0	0	1.0000	1.0000
ଉ	0.0004	0.0294	0.9706	0.9996	ଢ	0.0004	0	1.0000	0.9996
ଊ	0.0012	0.0364	0.9636	0.9988	ତ	0.0005	0.0199	0.9801	0.9995
ଋ	0.0012	0.0169	0.9831	0.9988	ଦ	0.0004	0.0166	0.9834	0.9996
ଓ	0.0004	0.0100	0.9900	0.9996	ନ	0.0020	0.0173	0.9827	0.9980
ଖ	0.0004	0	1.0000	0.9996	ବ	0.0008	0.0297	0.9703	0.9992
ଙ	0.0008	0.0361	0.9639	0.9992	ଭ	0.0009	0.0579	0.9421	0.9991
ଟ	0.0004	0	1.0000	0.9996	ର	0.0018	0.0239	0.9761	0.9982
ଛ	0.0004	0.0198	0.9802	0.9996	ଲ	0.0005	0.0295	0.9705	0.9995
ଜ	0.0004	0	1.0000	0.9996	ଳ	0.0004	0	1.0000	0.9996
ଝ	0.0018	0.0272	0.9728	0.9982	ୱ	0.0010	0.0201	0.9799	0.9990
ଞ	0.0003	0.0197	0.9803	0.9997	ହ	0.0014	0	1.0000	0.9986
ଚ	0.0001	0.0033	0.9967	0.9999					

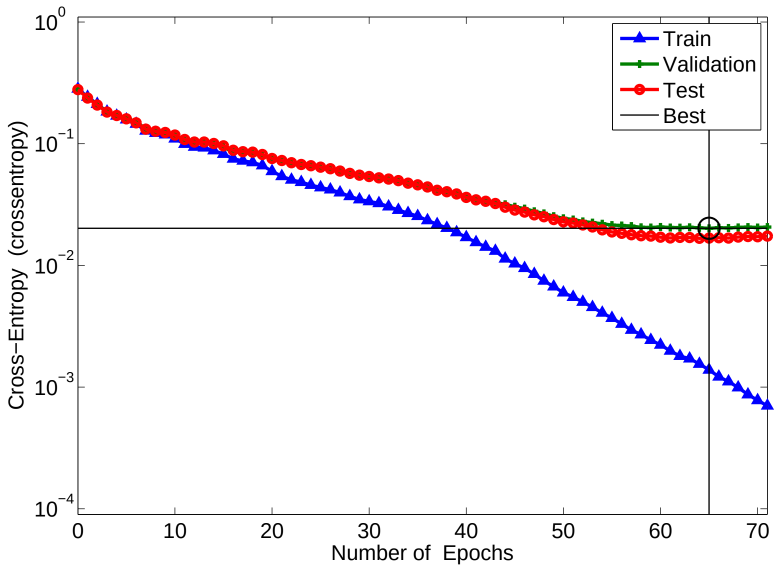

Figure 4.12: Performance of the network for Group II characters

Table 4.5: Accuracies of different schemes along with our proposed scheme.

Feature extraction	Classifier	Accuracy
Structural features [85]	Kohonen neural network	95.60%
Structural features [123]	Decision tree	96.30%
Curvature feature [86]	Quadratic	97.40%
Proposed Scheme	Back Propagation Neural network	98.55%

instance of misclassification is shown in Figure 4.15. The reason behind this is that the input image and the recognized character are almost similar in structure.

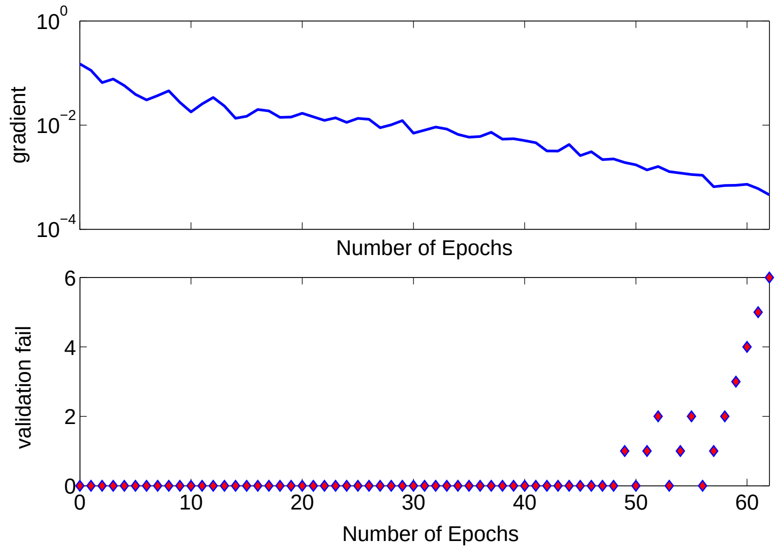

Figure 4.13: Convergence states of Group II characters

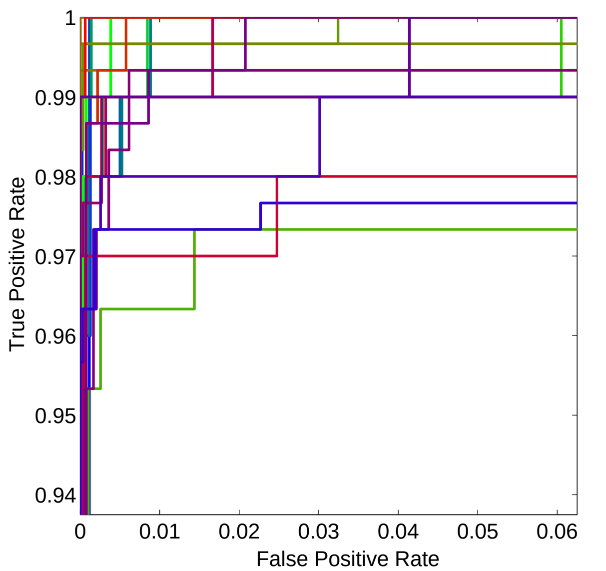

Figure 4.14: ROC curves of Group II characters

4.4 Summary

In this chapter, a multi-resolution technique, i.e., discrete orthogonal S-transform has been applied to recognize the handwritten Odia atomic characters including vowels and consonants from OHCS database. The overall dataset is divided into two groups based on a vertical line present in the character. The DOST feature coefficients are extracted from each normalized character of OHCS dataset. To overcome the computational complexity the feature matrix has been reduced using PCA and subsequently these features are used as inputs to the ANN classifier in the specific group. Performance comparison has been made with the existing schemes and it is observed that the proposed DOST feature has an overall accuracy of 98.55%.

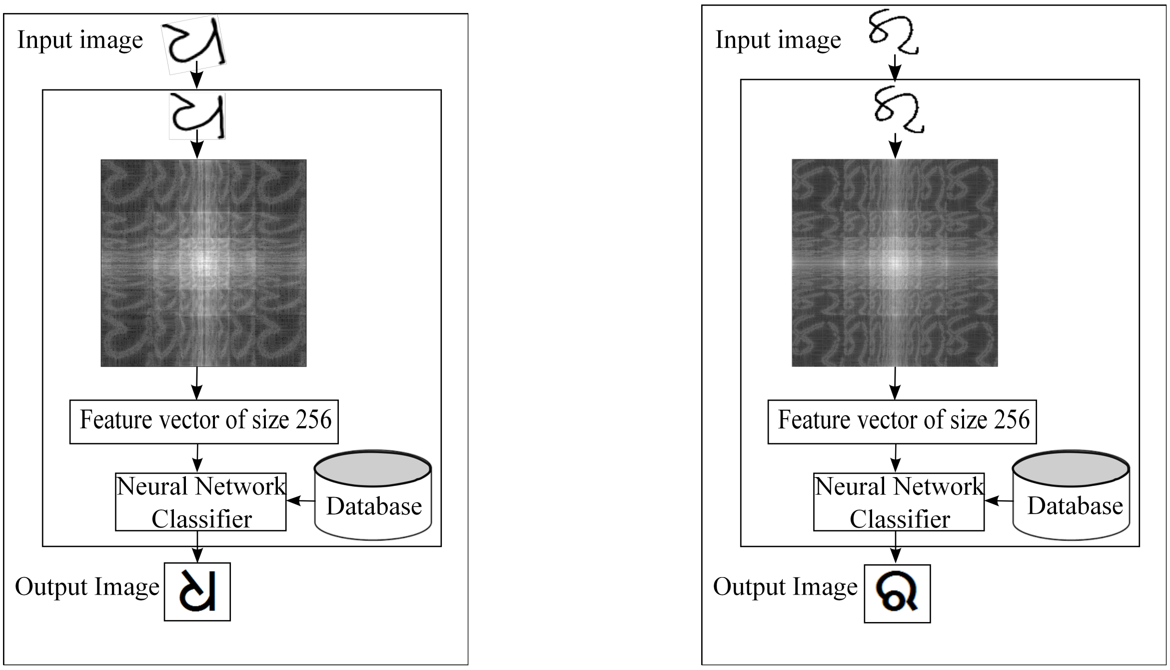

Figure 4.15: Odia consonant character ‘ଧ’ is recognized correctly (left) where as the vowel ‘ଈ’ is misclassified as ‘ର’ (right).

Table 4.6: Confusion matrix for Group II characters

	ଇ	ଈ	ଉ	ଊ	ଋ	ଓ	କ	ଙ	ଚ	ଛ	ଜ	ଝ	ଞ	ଟ	ଠ	ଡ	ଢ	ତ	ଦ	ନ	ବ	ଭ	ର	ଲ	ଳ	ଵ	ହ
ଇ	300	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ଈ	3	297	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ଉ	0	0	297	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0
ଊ	0	0	3	291	3	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ଋ	0	0	0	0	291	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	3	0
ଓ	0	3	0	0	0	297	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
କ	0	0	0	0	0	0	297	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ଙ	0	0	0	0	0	0	0	294	0	0	0	3	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0
ଚ	0	0	0	0	0	0	0	0	297	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ଛ	0	0	0	0	0	0	0	0	0	297	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ଜ	0	0	0	0	0	0	0	0	0	3	297	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ଝ	0	0	0	0	0	0	0	3	0	0	0	286	0	0	0	0	0	6	0	0	3	2	0	0	0	0	0
ଞ	2	0	0	0	0	0	0	0	0	0	0	0	298	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ଟ	0	0	1	0	0	0	0	0	0	0	0	0	0	299	0	0	0	0	0	0	0	0	0	0	0	0	0
ଠ	0	0	0	0	2	0	0	0	0	0	0	0	0	0	298	0	0	0	0	0	0	0	0	0	0	0	0
ଡ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	300	0	0	0	0	0	0	0	0	0	0	0
ଢ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	297	0	0	3	0	0	0	0	0	0	0
ତ	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	296	2	0	0	0	0	0	0	0	0
ଦ	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	297	0	0	0	0	0	0	0	0
ନ	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	284	0	0	0	10	0	0	0
ବ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	294	6	0	0	0	0	0
ଭ	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	2	3	293	0	0	0	0	0
ର	0	0	5	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	286	3	0	3	0
ଲ	0	2	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	296	0	0	0
ଳ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	297	0	0
ଵ	6	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	292	0
ହ	0	0	0	0	0	0	0	3	0	0	0	0	0	1	0	0	0	0	0	0	0	0	4	3	0	0	289

Chapter 5

Structural Feature-based Classification and Recognition of Handwritten Odia Character (HOCR-SF)

Handwritten information used as an imperative method of communication between the people since the start of humankind. Therefore, development of schemes regarding document image analysis for the modern dialects will facilitate the living style of human being. Character recognition is one of the significant subfields of document image processing. It involves identifying the various characters that make up the text of the document. Despite everything it remains an open issue, regardless of the fact that with the coming of the advanced gadgets for transforming the document into its digitized form. Penmanship varies from individual to individual and extraordinarily impacted by numerous factors such as the received handwriting education, the type of paper used, the writing instrument, the background and furthermore factors like anxiety, inspiration and even the motivation behind the penmanship. The challenging nature of handwriting recognition has pulled in consideration of researchers from various fields. The industry has shown significant interest in handwriting recognition research due to a large number of applications that exist. The development in the process of recognizing Odia character started in late 90's. Odia language has 47 basic characters including vowels and consonants altogether. When the number of classes increases, it becomes more challenge to achieve high accuracy and speed of any OCR designed for Odia language. One feasible solution to this problem is to reduce the amount of computation for recognition. In this regard, the characters need to be segregated into multiple groups, and separate classifiers can be utilized to recognize the character. There are two profound approaches for implementing OCR systems namely statistical and structural that give a lead for various methods to describe and classify. The statistical method uses decision theory to discriminate among objects belonging to different groups based on their quantitative features. On the other hand, structural approaches to pattern recognition use syntactic grammars to discriminate among objects belonging to the various groups based on the shape features. Structural feature analysis has proven to be useful for classification of objects. However, it has certain constraints but, the advantage is that it provides a good shape description of the image and makes a system useful for analyzing handwritten documents. Hence, structural features give a pattern having the global and local properties of an object. The objective of the present work is to distinguish those attributes that might be utilized to separate characters into classes successfully and further to use other structural features to recognize a character within a group.

The remainder of the chapter machinated as follows. Proposed scheme along with the feature extraction in each phase discussed in Section 5.1. PCA-based feature reduction is presented in Section 5.2. Classification followed by the recognition given in section 5.3. Finally, the results are outlined in Section 5.4 followed by a summary of our proposed scheme.

5.1 Proposed Methodology

The suggested scheme which has been named as Handwritten Odia Character Recognition using Structural Features (HOCR-SF) operates in two phases as shown in Figure 5.1. In the first phase, the overall Odia character set has been classified into two groups namely *Group I* and *Group II* using a linear Support Vector Machine (SVM) based on the shape properties of the characters. In *Group I* all characters with a vertical line are present whereas others fall into *Group II*. Further, some more unparalleled structural features of the character of each group are extracted and fed to a BPNN for recognition. Separate BPNN networks have been designed for classifying the characters in each group. The Overall accuracy of 98.87% has been obtained.

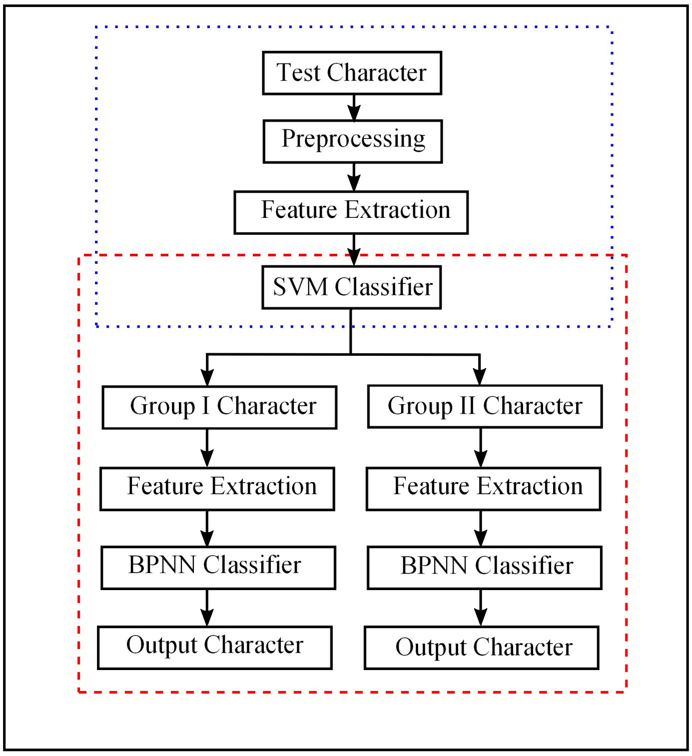

Figure 5.1: Block diagram of proposed two stage classification scheme

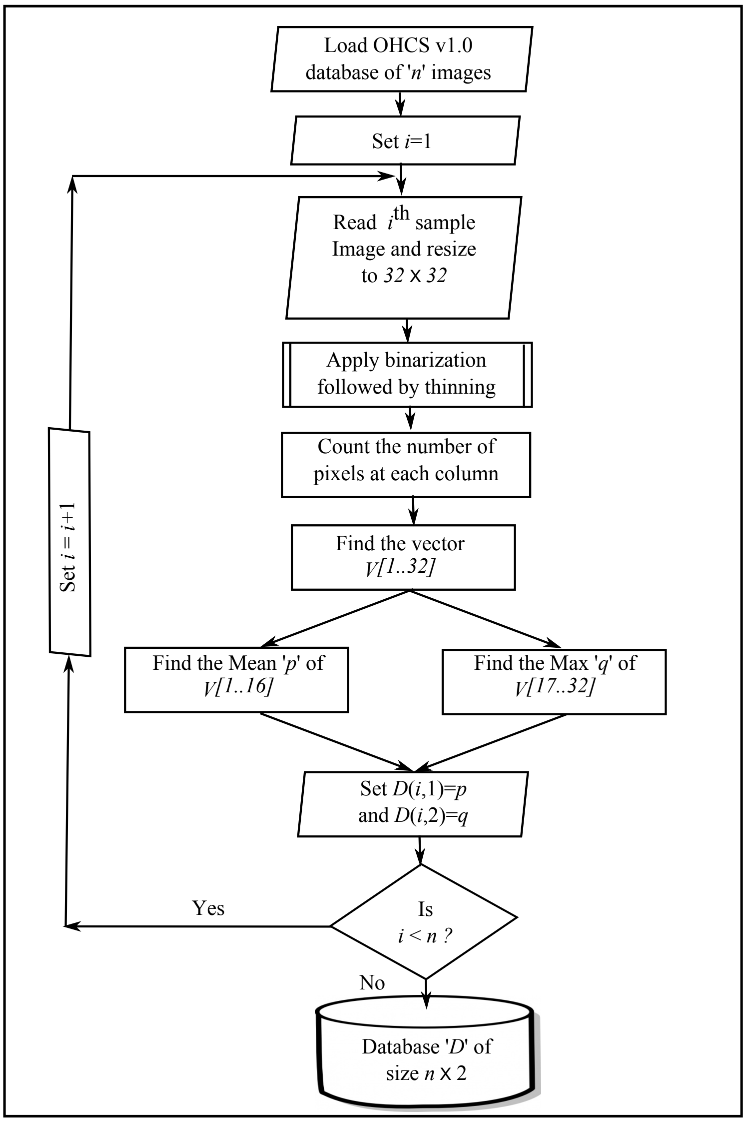

Figure 5.2: Flow graph of first stage classification of OHCS database.

5.1.1 First Stage Feature Extraction and Classification

The flowgraph of the process in the extracting feature during the first phase has been shown in Figure 5.2. The images from OHCS database have been size normalized to 32×32 from 81×81 just to reduce the computational overhead. Applying the Otsu's method [115], each image has been binarized and thinned to a single pixel width. By using vertical projection method, a one-dimensional vector V of length 32, i.e., $V[1...32]$ computed where the values represent the number of pixels in each column of the character image. The mean value of the lower half, i.e., $V[1...16]$ and the maximum value of the upper half, i.e., $V[17...32]$ together represents a feature point of a character and used as input to the classifier. For the sake of the readers, we have explained the process considering a sample character (see Figure 5.3(a)) from the testing set. The character is thinned and shown in Figure 5.3(b). In Figure 5.3(c), the histogram is plotted. Then the number of pixels are added up column wise and a vector V of length 32 is generated. Now, the mean values p of the first sixteen values of the vector

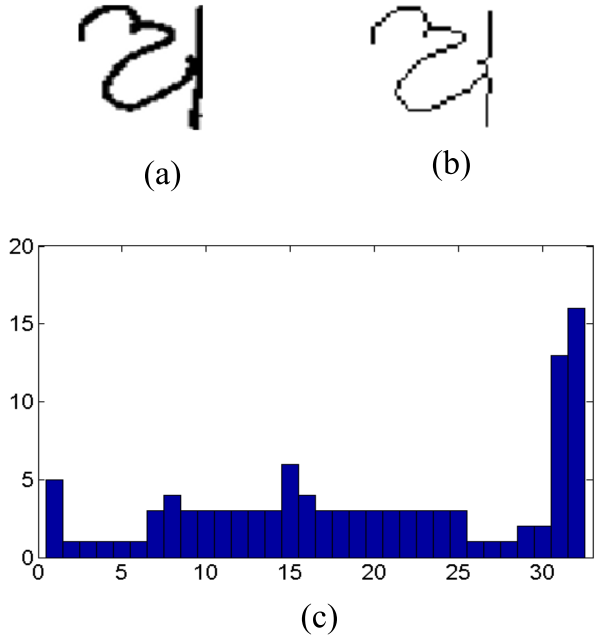

Figure 5.3: The sample first letter is considered (a), the image is thinned and is shown in (b), the histogram is shown in (c).

V and the maximum value q in the last sixteen values of the vector V are computed and stored as a point in two dimensional. Repeating the above process for all images of OHCS database leads to a matrix of size 15040×2. Our objective is to divide the whole database into two groups based on finding a vertical bar at the end of the character. To solve this two class problem a linear SVM has been utilized, and the steps followed are given in Algorithm 3. Out of the 15040 samples only 7520 patterns have been used in the training set and the

---

Algorithm 3: LinearSVM(D,y_i)

Input: Database D of size 15040×2 and the label matrix y_i where each value is either +1 or -1.

Output: Confusion matrix (CT).

1. Divide the whole matrix into two set, i.e., *training* and *testing* set having 7520 samples in each set using cross validation technique.
2. Design and train a simple linear binary SVM classifier using the training samples.
3. Predict the label for all samples in the *testing set* using the trained SVM model.
4. Store the result in CT.
5. **return** *CT*

---

rest in testing set. For example, in case of the character shown in Figure 5.3(a), the value of p is calculated as 2.8125 and q is found to be 16. So, the point is $(2.8125, 16)$, which is located in the upper half of the Figure 5.4 and hence, it is classified as *Group I*. It ensures that the character under testing has a vertical bar at the end. Certainly, the proposed method of grouping the samples gives an error about 0.0014% due to some characters which are written in unorthodox manner. We have also presented in Table 5.1, the confusion matrix

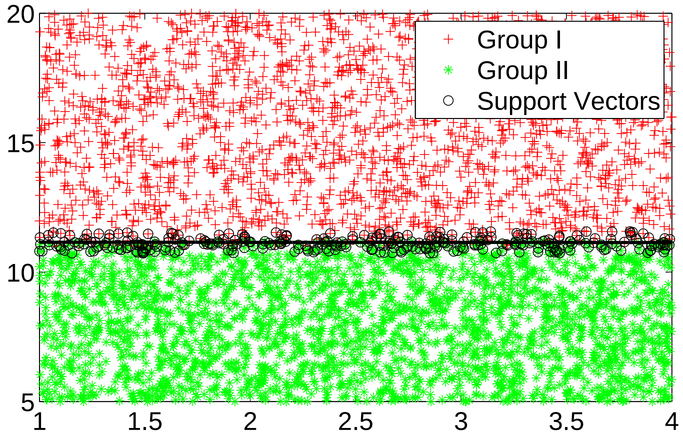

Figure 5.4: Output of the SVM classifier

which clearly indicates the error rate is about 0.0014%.

Table 5.1: Confusion matrix while grouping the characters on the testing set consisting of 7520 total samples

	Group I	Group II
Group I	3197	3
Group II	5	4315

5.1.2 Second Stage Feature Extraction

Once the character is classified to either *Group I* or *Group II*, the scale invariant structural features for each image are calculated to classify the character within the group. The structural features of size 210 are calculated individually for each character which is further reduced to 64 using principal component analysis. Proposed algorithm for the evaluation of structural features for each image generating 210 features are shown in the following steps,

1. The thinned image is divided into 30 pieces of uniform length.

2. Find the centroid of that whole image.

3. Calculate the following values l, r, θ for all 30 pieces of a character as shown in Figure 5.5.

4. Seven features for each piece of arc are extracted as follows $l_i, r_i, \theta_i, l_i + r_i + \theta_i, \frac{l_i}{l_i+r_i+\theta_i}, \frac{r_i}{l_i+r_i+\theta_i}, \frac{\theta_i}{l_i+r_i+\theta_i}$, where i=1,..,30.

5. So, a total of $7 \times 30 = 210$ features are extracted for each character.

6. The steps from 1 to 4 are repeated for all images present in the training set.

7. Apply the PCA based feature reduction

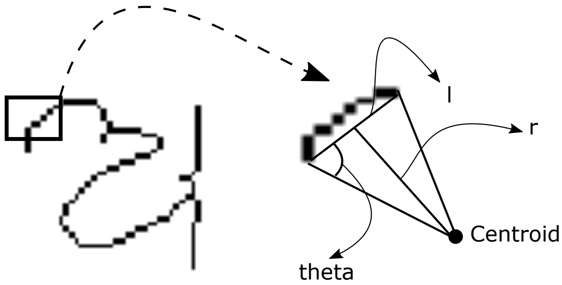

Figure 5.5: Finding l, r, θ from a piece of arc

5.2 Feature Reduction using PCA

Principal Component Analysis (PCA) is a dimensionality-reduction technique was first introduced by Pearson in 1901 and then by Hotelling in 1933 [124, 125]. It is often used to convert a high-dimensional dataset into a smaller-dimensional subspace prior to running a machine learning algorithm on the dataset [126]. The steps for PCA based dimensionality reduction is given in Algorithm 4. It is a statistical method to describe the variation in a

Algorithm 4: PCA-based feature reduction

Input: An original dataset D of size $N \times d$, where N denotes the total number of samples and d is the total number of features in each sample.

Output: The reduced dataset D' of size $N \times k$, where $k \leq d$.

1 For a given $N \times d$ data matrix, subtract the mean μ from each row vector in D where mean is calculated by

$$\mu_j = \frac{1}{m} \sum_{i=1}^{m} D(j, i) \tag{5.1}$$

2 Calculate the covariance matrix Σ of D.

3 Find eigenvectors and eigenvalues of Σ.

4 Sort the eigenvectors by decreasing eigenvalues and choose k eigenvectors with the largest eigenvalues to form a $N \times k$ dimensional matrix W.

5 Use this $N \times k$ eigenvector matrix to transform the samples onto the new subspace. $D' = D \times W$

6 **return** D'

set of multivariate data regarding a set of discriminant principal components[122]. These values are orthogonal because they are the eigenvectors of the covariance matrix, which is symmetric. In the suggested scheme, the original feature matrix of size 6402×210 of *Group I* characters has been reduced to 6402×64 and for *Group II* it is reduced from 8638×210 to 8638×64 .

5.3 Recognition using BPNN

The first stage of classification assign the character into either *Group I* or *Group II* based on the vertical bar at the end of the character. There are twenty characters in *Group I* and twenty-seven in *Group II*. Discriminating the samples within a group is a multi-class problem. Thus, two back propagation neural network have been applied for each group to predict the actual class of a new sample. To validate the proposed scheme, simulation has been carried out on OHCS dataset using 70% of the data for training and rest 30% is divided equally into two parts, i.e., validation and test sets. For faster training, the scaled conjugate gradient method is used. The training of the neural network stops when either the gradient reaches a value of 10^{-6} or the number of consecutive validation checks reaches 6, i.e., there is no better improvement of the gradient for six continuous iterations. The performance of the classification is evaluated with the help of the confusion matrix.

5.4 Results and Discussions

All experiments are carried out on a Dell machine with the following configuration: Intel(R) Core(TM) $i7-4770$ processor, CPU@ 3.4GHz, 6GB RAM and 64 bit operating system. The scheme is simulated in the MATLAB environment. Two experiments have been conducted wherein the first the whole dataset has been taken into account and in the second approach the dataset is divided into two groups followed by recognition. In our first experiment, the whole database is considered and a three-layered back propagation neural network of size $64-58-47$ has been used for classification of the characters. The input and output layer size are based on the dimension of features and output classes respectively, while for the hidden layer, the accuracy at different sizes were compared. Experimentally, it is observed that the network gives a good result when the number of hidden neurons is 58. The whole dataset is divided into the following sets,

- 10528 samples are used for training.
- 2256 samples are used for testing.
- 2256 samples are used for validation.

The confusion matrix is shown in Table 5.9 and the accuracy is recorded as 95.87%. To improve the accuracy of the system a two stage based classification method has been adopted in which all characters of OHCS database is divided into two groups by a vertical line present at the end of few characters. The experimental setup and the steps followed are discussed below in detail.

5.4.1 Experimental Setup for the Recognition of Group I Characters

The entire feature set of *Group I* character (a total of 6402 samples) sets is divided into the following subsets,

- 4482 samples are used for training.
- 960 samples are used for testing.
- 960 samples are used for validation.

A three layer perceptron of size $64 - 38 - 20$ has been used for classification. The input and output layer size are based on the dimension of features and output classes respectively, while for the hidden layer, the accuracy at different sizes were compared. For our experiment we varied the size of the hidden layer from 30 to 80 and the percentage of accuracy is recorded and presented in Table 5.2. Experimentally best result comes when the number of hidden neurons is 38. Table 5.3 illustrates different performance measures for all samples of basic

Table 5.2: Accuracy analysis with varying neurons in the hidden layer for Group I dataset

Number of Neurons in the Hidden Layer	Number of samples in the training set	Number of samples in the validation set	Number of samples in the testing set	Accuracy % in percentage
30	5762	320	320	98.28
	5122	640	640	96.37
	4482	960	960	95.46
	3842	1280	1280	94.12
	3202	1600	1600	88.36
	2562	1920	1920	82.44
	1922	2240	2240	75.28
38	5762	320	320	99.11
	5122	640	640	99.02
	4482	**960**	**960**	**98.93**
	3842	1280	1280	94.26
	3202	1600	1600	92.48
	2562	1920	1920	87.73
	1922	2240	2240	78.46
45	5762	320	320	97.98
	5122	640	640	95.39
	4482	960	960	94.16
	3842	1280	1280	93.72
	3202	1600	1600	82.15
	2562	1920	1920	78.27
	1922	2240	2240	72.48

Odia character (BOC) in *Group I*. The confusion matrix for *Group I* is displayed in Table 5.4. It has been noticed that almost all characters are classified correctly, however, the Odia character ‘ଣ’ (‘na’) is wrongly classified as ‘ଥ’ (‘tha’) 3 times, ‘ଶ’ (‘sa’) 6 times. Similarly the character ‘ପ’ (‘pa’) is miss-classified as ‘ଅ’ (‘a’) 3 times, as ‘ଗ’ (‘ga’) 3 times, as ‘ଧ’ (‘dha’) 3 times, and as ‘ଘ’ (‘pa’) 3 times. It is due to the structural similarity between the characters. The performance of the network is shown in Figure 5.6 and it is observed that

Table 5.3: Performance parameters of Group I characters

BOC	FNR	FPR	TNR	TPR	BOC	FNR	FPR	TNR	TPR
ଅ	0.0005	0.0365	0.9635	0.9995	ଧ	0	0.0184	0.9816	1.0000
ଆ	0	0	1.0000	1.0000	ପ	0.0020	0.0065	0.9935	0.9980
ଏ	0	0	1.0000	1.0000	ଫ	0.0005	0	1.0000	0.9995
ଐ	0	0.0093	0.9907	1.0000	ମ	0.0015	0	1.0000	0.9985
ଔ	0.0005	0	1.0000	0.9995	ଯ	0.0010	0.0095	0.9905	0.9990
ଖ	0.0010	0	1.0000	0.9990	ୟ	0.0005	0	1.0000	0.9995
ଗ	0.0010	0.0187	0.9812	0.9990	ଶ	0	0.0361	0.9639	1.0000
ଘ	0.0015	0.0189	0.9811	0.9985	ଷ	0	0.0093	0.9907	1.0000
ଣ	0.0015	0.0189	0.9811	0.9985	ସ	0.0008	0.0094	0.9906	0.9992
ଥ	0.0005	0.0276	0.9724	0.9995	ଝ	0.0010	0.0368	0.9632	0.9990

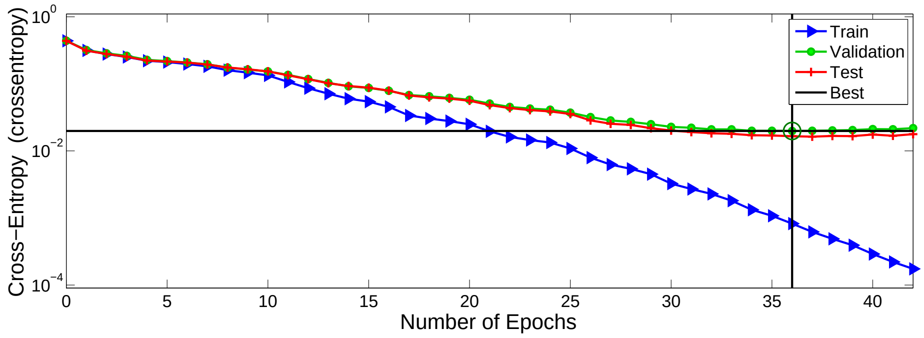

Figure 5.6: Performance of the network for Group I characters

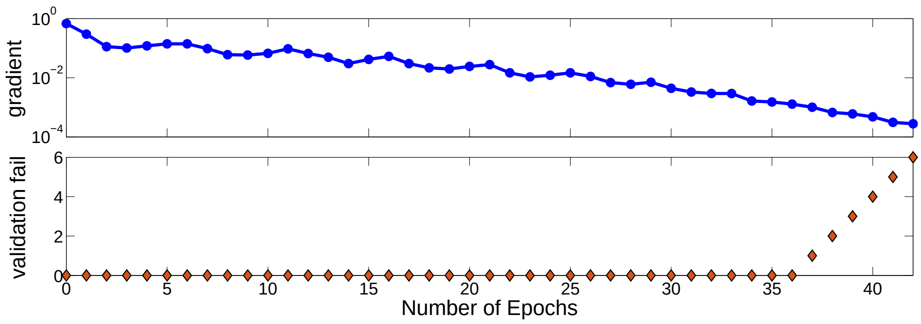

Figure 5.7: Convergence states of Group I characters

the best validation performance is 0.019859 at 36 epochs. The convergence characteristic is shown in Figure 5.7. The accuracy of 98.93% is recorded for *Group I* characters.

5.4.2 Experimental Setup for the Recognition of Group II Characters

The entire feature set of *Group II* characters (a total of 8,638 samples) sets is divided into the following subsets,

- 6046 samples are used for training.
- 1296 samples are used for testing.
- 1296 samples are used for validation.

Table 5.4: Confusion matrix for Group I characters

	ଅ	ଆ	ଏ	ଐ	ଔ	ଖ	ଗ	ଘ	ଣ	ଥ	ଧ	ପ	ଫ	ମ	ଯ	ୟ	ଶ	ଷ	ସ	କ୍ଷ
ଅ	317	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0
ଆ	0	320	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ଏ	0	0	320	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ଐ	0	0	0	320	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ଔ	0	0	0	0	317	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0
ଖ	0	0	0	0	0	314	0	3	0	0	0	0	0	0	0	0	3	0	0	0
ଗ	0	0	0	0	0	0	314	3	3	0	0	0	0	0	0	0	0	0	0	0
ଘ	3	0	0	0	0	0	0	311	0	0	0	0	0	0	3	0	0	0	0	3
ଣ	0	0	0	0	0	0	0	0	311	3	0	0	0	0	0	0	6	0	0	0
ଥ	3	0	0	0	0	0	0	0	0	317	0	0	0	0	0	0	0	0	0	0
ଧ	0	0	0	0	0	0	0	0	0	0	320	0	0	0	0	0	0	0	0	0
ପ	3	0	0	0	0	0	3	0	0	3	3	308	0	0	0	0	0	0	0	0
ଫ	0	0	0	0	0	0	0	0	0	0	0	0	317	0	0	0	0	0	3	0
ମ	3	0	0	0	0	0	3	0	0	0	0	0	0	311	0	0	0	3	0	0
ଯ	0	0	0	0	0	0	0	0	3	0	0	0	0	0	314	0	0	0	0	3
ୟ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	317	0	0	0	3
ଶ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	320	0	0	0
ଷ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	320	0	0
ସ	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	315	3
କ୍ଷ	0	0	0	3	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	314

Another multilayer perceptron of size $64 - 55 - 27$ was used. Similarly, we varied the size of the hidden layer from 30 to 80 and the percentage of accuracy is recorded in each case and presented in Table 5.5. Experimentally best result comes when the number of hidden neurons is 55. Table 5.6 lists the false negative rate, false positive rate, true negative

Table 5.5: Accuracy analysis with varying neurons in the hidden layer for Group II dataset

Number of Neurons in the Hidden Layer	Number of samples in the training set	Number of samples in the validation set	Number of samples in the testing set	Accuracy % in percentage
45	7774	432	432	95.38
	6910	864	864	91.57
	6046	1296	1296	90.17
	5182	1728	1728	89.20
	4318	2160	2160	85.63
	3454	2592	2592	80.14
	2590	3024	3024	74.28
55	7774	432	432	99.48
	6910	864	864	99.12
	6046	**1296**	**1296**	**98.82**
	5182	1728	1728	96.26
	4318	2160	2160	90.62
	3454	2592	2592	87.43
	2590	3024	3024	78.80
65	7774	432	432	97.72
	6910	864	864	92.19
	6046	1296	1296	89.60
	5182	1728	1728	86.32
	4318	2160	2160	82.65
	3454	2592	2592	75.17
	2590	3024	3024	70.81

rate, and true positive rate for all samples of basic Odia character in *Group II*. The confusion matrix for *Group II* is portrayed in Table 5.8. Almost all characters are classified correctly,

Table 5.6: Performance parameters of Group II characters

BOC	FNR	FPR	TNR	TPR	BOC	FNR	FPR	TNR	TPR
ଇ	0.0004	0.0186	0.9814	0.9996	ଠ	0.0004	0.0094	0.9906	0.9996
ଈ	0.0007	0	1.0000	0.9993	ଡ	0	0	1.0000	1.0000
ଉ	0.0004	0.0276	0.9724	0.9996	ଢ	0.0014	0.0096	0.9904	0.9986
ଊ	0.0007	0.0542	0.9458	0.9993	ତ	0	0.0093	0.9907	1.0000
ଋ	0.0014	0.0096	0.9904	0.9986	ଦ	0.0011	0.0372	0.9628	0.9989
ଓ	0.0007	0.0063	0.9937	0.9993	ନ	0	0.0093	0.9907	1.0000
ଘ	0.0004	0	1.0000	0.9996	ବ	0	0.0184	0.9816	1.0000
ଙ	0	0	1.0000	1.0000	ଭ	0.0004	0	1.0000	0.9996
ଚ	0	0	1.0000	1.0000	ର	0.0007	0.0095	0.9905	0.9993
ଛ	0.0007	0.0368	0.9632	0.9993	ଲ	0.0004	0.0094	0.9906	0.9996
ଜ	0.0011	0	1.0000	0.9989	ଳ	0	0.0093	0.9907	1.0000
ଝ	0.0007	0	1.0000	0.9993	ଵ	0	0.0361	0.9639	1.0000
ଞ	0.0010	0.0095	0.9905	0.9990	ହ	0.0004	0.0365	0.9635	1.0000
ଟ	0.0011	0	1.0000	0.9989					

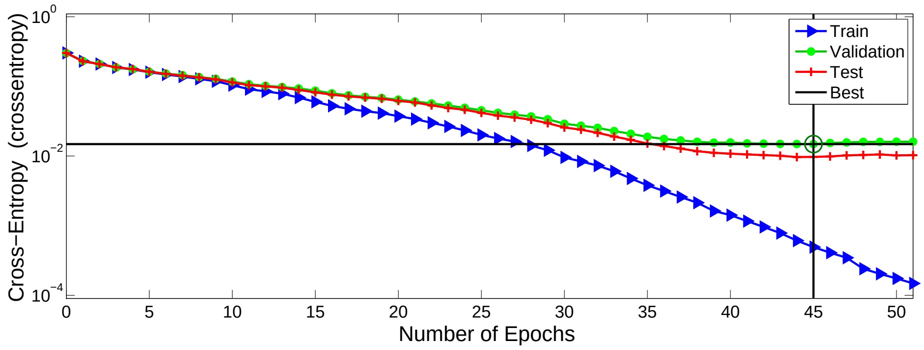

Figure 5.8: Performance of the network for Group II characters

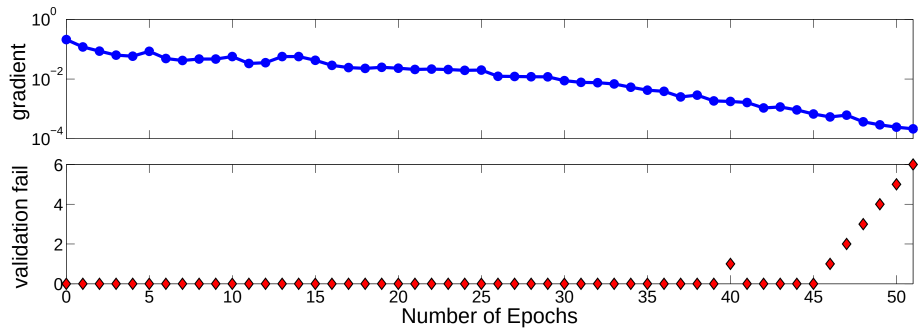

Figure 5.9: Convergence states of Group II characters

Table 5.7: Accuracies of different schemes on Odia character along with our proposed scheme.

Feature extraction	Classifier	Accuracy
Structural features [85]	Kohonen neural network	95.60%
Structural features [123]	Decision tree	96.30%
Curvature feature [86]	Quadratic classifier	94.60%
Structural feature (proposed)	SVM + Neural network	98.87%

however the Odia character 'ଇ' ('i') is wrongly classified as 'ଉ' ('u') 6 times. Similarly the character 'ଋ' ('ru') is wrongly classified as 'ଊ' ('uu') 3 times, as 'ବ' ('ba') three times, and as 'ଲ' ('la') also three times. Their are also few other characters which are also wrongly classified to some other character. It happens due to the structural similarity between the characters. The performance of the network is shown in Figure 5.8 and it is observed that the best validation performance is 0.014855 at 45 epochs. The convergence characteristics are also shown in Figure 5.9. The accuracy recorded as 98.82% for *group II* characters. The average overall accuracy of our scheme is recorded as 98.87%. We have given the accuracy of few existing schemes and is shown in the Table 5.7.

5.5 Summary

A two-stage classification mechanism has been adopted for identifying atomic Odia character written by hand. The OHCS database has been split into two groups based on shape characteristics of the character. A linear support Vector Machine (SVM) classifier has been applied for the above two class problem. In the second phase, the structural features are extracted followed by PCA based feature reduction technique to derive the 64 discriminant features from each character in the groups for classification. Back propagation neural networks (BPNN) with different configuration are employed for the recognition of character present in each group. The first stage classification of OHCS database using linear support vector machine gives an accuracy of 99.99%. The percentage of error associated with *Group I* classification is about 1.07% and it is 1.18% in case of *Group II* characters. A relative investigation has been made on a sensibly generous dataset with the competent schemes. From test outcomes, it is evident that the proposed method outmaneuvers different schemes on OHCS database and provides an overall accuracy of 98.87%.

Table 5.8: Confusion matrix for Group II characters

	ଇ	ଈ	ଉ	ଊ	ଋ	ଓ	କ	ଙ	ଚ	ଛ	ଜ	ଝ	ଞ	ଟ	ଠ	ଡ	ଢ	ତ	ଧ	ନ	ବ	ଭ	ର	ଲ	ଳ	ଵ	ହ
ଇ	317	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0
ଈ	0	314	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ଉ	0	0	317	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3
ଊ	0	0	3	314	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0
ଋ	0	0	0	3	308	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	3	0	3
ଓ	0	0	0	3	0	314	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0
କ	0	0	0	3	0	0	317	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ଙ	0	0	0	0	0	0	0	320	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ଚ	0	0	0	0	0	0	0	0	320	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ଛ	0	0	0	0	0	0	0	0	0	314	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0
ଜ	0	0	0	0	0	0	0	0	0	0	311	0	0	0	0	0	0	0	3	0	3	0	0	0	0	0	3
ଝ	0	0	0	0	0	0	0	0	0	0	0	314	0	0	0	0	3	0	0	3	0	0	0	0	0	0	0
ଞ	3	0	0	3	0	2	0	0	0	0	0	0	312	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ଟ	0	0	0	3	0	0	0	0	0	3	0	0	0	311	3	0	0	0	0	0	0	0	0	0	0	0	0
ଠ	3	0	0	0	0	0	0	0	0	0	0	0	0	0	317	0	0	0	0	0	0	0	0	0	0	0	0
ଡ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	320	0	0	0	0	0	0	0	0	0	0	0
ଢ	0	0	0	0	0	0	0	0	0	3	0	0	3	0	0	0	308	3	3	0	0	0	0	0	0	0	0
ତ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	320	0	0	0	0	0	0	0	0	0
ଧ	0	0	0	3	0	0	0	0	0	3	0	0	0	0	0	0	0	0	311	0	0	0	3	0	0	0	0
ନ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	320	0	0	0	0	0	0	0
ବ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	320	0	0	0	0	0	0
ଭ	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	317	0	0	0	0	0
ର	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	314	3	0	0	0
ଲ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	317	0	0	3
ଳ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	320	0	0
ଵ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	320	0
ହ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	317

Table 5.9: Confusion matrix for the whole dataset

314	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0
0	320	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	302	6	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	3	0	0	0	0	3	0
0	0	0	308	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	3	0	0	0	0	0	0	0	0
3	0	5	0	300	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	3	0	0	0	0	0	0
0	0	3	0	0	311	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0
0	0	0	0	3	0	305	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	3	0	3	0	0	0	0	3
0	0	0	0	0	0	3	317	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	314	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0
0	0	0	0	0	0	0	0	0	308	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	3
0	0	0	0	0	0	0	0	0	0	320	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	3	0	0	0	0	0	0	0	3	0	302	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	3	0	0	3
0	0	0	0	0	0	0	0	0	0	0	0	320	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	320	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	308	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	9	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	317	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	317	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	314	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	317	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	311	0	0	0	0	3	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	3	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	308	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0
0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	311	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	317	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	320	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	317	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	317	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	3	0	0	0	0	0	311	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	3	0	0	314	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	314	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	314	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	317	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	3	0	0	0	0	0	0	0	0	3	0	308	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	311	0	0	0	3	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	317	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	3	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	311	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	314	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	320	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	320	0	0	0	0	0	0	0	0	0
0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	314	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	6	0	0	0	0	308	0	0	0	0	0	0	0
0	0	3	0	3	3	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	308	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	320	0	0	0	0	0
0	0	0	0	0	0	3	0	0	3	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	308	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	3	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	311	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	2	0	4	0	0	0	0	2	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	306	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	314	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	320

Chapter 6

Recognition of Atomic Odia Character using Deep Learning Network (HOCR-DBN)

Till now, we have examined the statistical as well as structural features of the Odia handwritten character or numeral. These days it is exceptionally prevalent to utilize deep architectures in machine learning. A stack of Restricted Boltzmann Machines (RBM) is used to create a powerful generative model called Deep Belief Networks (DBNs). Geoffrey Hinton discovered this model in the year 2006. It has the ability like feature extraction, classification and also used in some practical applications such as object recognition, speech recognition, and natural language processing [127]. This concept comes from the way the human brain memorize any incident. A healthy human mind is always faster than any system. The complex structure of our brain helps in learning simple concepts and then compose them to represent more abstract ones. So far, our observations are based on supervised learning. The development of Deep Belief Network (DBN) hurtled us to apply the technique on our database. Data representation plays a vital role in the pattern recognition [128, 129]. The benefit of feature learning by DBN is that it uses unlabeled data in high-level feature extraction, and moreover it increases the discrimination between extracted features. Restricted Boltzmann Machine (RBM) is a learning tool which has been utilized as the feature extractor in a large variety of classification problems. Restricted Boltzmann machines have become powerful generative models and used mostly to extract features from the unlabeled data as well as it is applied to construct the Deep Belief Network [130]. Hinton et al. [131] used DBN for the recognition of digit from MNIST data set. He has considered the structure of the network as $784 - 500 - 500 - 2000 - 10$, where the first layer takes 784 inputs. The final layer has ten neurons, and other three layers are hidden layers with stochastic binary neurons. Their scheme achieved 98.75% classification accuracy on MNIST test data set and moreover it works on unlabeled data. Multiple DBNs namely, back propagation DBN (BP-DBN) and the associative memory DBN (AM-DBN) have been utilized for phone recognition by Mohamed et al. [132]. The simulation was carried out on the standard TIMIT corpus, and their suggested scheme achieved a phone error rate (PER) of 23.0% on the TIMIT core test set. In this chapter, a scheme utilizing the deep belief network has been suggested. It not only extracts useful features but also recognize them with acceptable discrimination between them without using label information. The experiment has been carried out on the training set of OHCS Odia character dataset. A comparison has been made of the proposed scheme with the state of the art methods.

6.1 Restricted Boltzmann Machines (RBM)

Multiple layers of RBMs are used to construct DBN. Restricting the connections between nodes in a Boltzmann machine to only those between a hidden and a visible node, gives rise to the Restricted Boltzmann machine (RBM). Figure 6.1 shows a simple rendering of an RBM with four visible nodes and three hidden nodes. In the Following section, the RBMs and the training of RBM are discussed. More technically, a Restricted Boltzmann Machine

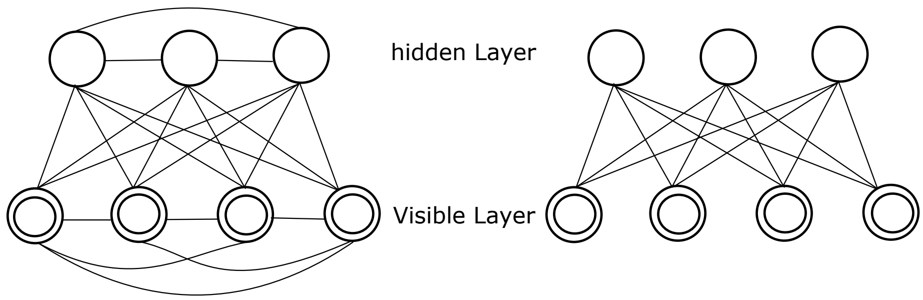

Figure 6.1: Boltzmann Machine is a fully connected graph(left) where as the RBM(right) has no hidden-hidden and visible-visible connectivity.

is a stochastic neural network [130] consisting of:

- One layer of visible units
- One layer of hidden units
- A bias unit

Furthermore, each visible unit is connected to all the hidden units (this connection is undirected, so each hidden unit is also connected to all the visible units), and the bias unit is connected to all the visible units and all the hidden units. To make learning easier, we restrict the network so that no visible unit is connected to any other visible unit and no hidden unit is connected to any other hidden unit. Hence it is called as Restricted Boltzmann Machine. It has been successfully applied to problems involving high-dimensional data such as images [133]. In this context, two approaches are usually followed. First, an RBM is trained in an unsupervised manner to model the distribution of the inputs (possibly more than one RBM could be trained stacking them on top of each other). Then, the RBM is used in one of two ways: either its hidden layer is used to preprocess the input data by replacing it with the representation given by the hidden layer, or the parameters of the RBM are used to initialize a feed-forward neural network. In both cases, the RBM is paired with some other learning algorithm (the classifier using the preprocessed inputs or the neural network) to solve the supervised learning problem at hand. Moreover, since the RBM is trained in an unsupervised manner, it is blind to the nature of the supervised task that needs to be solved and provides no guarantees that the information extracted by its hidden layer will be useful. The graphical

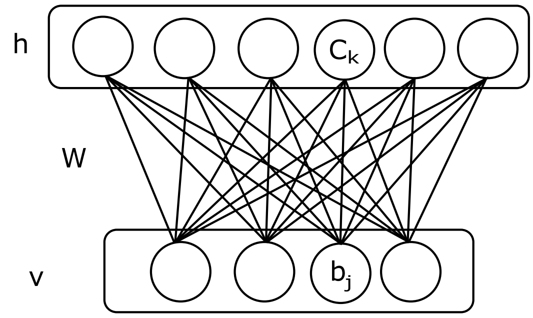

Figure 6.2: Two layer RBM model

model of an RBM is a fully-connected bipartite graph as shown in Figure 6.2, where **h** is the hidden layer (binary values), **v** is the visible layer (binary units), **W** is the weight matrix, and both b_j and c_k are bias vectors. The original Boltzmann Machine has a connection between the nodes in a given layer where as RBM don't have these edges for which we call it as restricted Boltzmann Machine. The joint distribution of an input **v** = $(v_1, ..., v_d)$ and using a hidden layer of binary stochastic units h = $(h_1, ..., h_n)$ is given by the following probabilistic function,

$$P(\mathbf{v}, \mathbf{h}) = \frac{1}{Z} \exp\{-E(\mathbf{v}, \mathbf{h})\} \tag{6.1}$$

where $E(\mathbf{v}, \mathbf{h})$ is the energy function and it is defined as

$$E(\mathbf{v}, \mathbf{h}) = -\sum_{i \in \mathbf{v}} b_i v_i - \sum_{j \in \mathbf{h}} c_j h_j - \sum_{i,j} v_i h_j w_{ij} \tag{6.2}$$

where v_i is the binary state of i^{th} visible unit, h_j is the binary state of the j^{th} hidden unit. Here, b_i and c_j are the bias units and w_{ij} is the weight between them. Every possible pair of a visible and a hidden vector is represented by the energy function $P(\mathbf{v}, \mathbf{h})$ The constant Z is known as partition function which is defined as follows

$$Z = \sum_{\mathbf{v}} \sum_{\mathbf{h}} \exp\{E(\mathbf{v}, \mathbf{h})\} \tag{6.3}$$

In graphical model, computation of Z is intractable it means solving $P(v, h)$ is also intractable. But conditional probability is easily computable. That means $P(v|h)$ (probability distribution over v given h) and $P(h|v)$ (probability distribution over h given v) are simple to compute. As we know,

$$P(h|v) = \frac{P(v, h)}{P(v)}$$

where $P(h|v)$ is the conditional probability of h given v, $P(v)$ is the probability of v, and $P(h,v)$ is the joint probability of v and h. The above equation can be rewritten as,

$$\begin{aligned}
P(h|v) &= \frac{\frac{1}{Z}\exp\{-E(h,v)\}}{\sum_h P(v,h)} \\
&= \frac{\frac{1}{Z}e^{(b^Tv+c^Th+v^TWh)}}{\frac{1}{Z}\sum_h e^{(b^Tv+c^Th+v^TWh)}} \\
&= \frac{e^{(b^Tv+c^Th+v^TWh)}}{\sum_h e^{(b^Tv+c^Th+v^TWh)}} \\
&= \frac{e^{(b^Tv+c^Th+v^TWh)}}{e^{(b^Tv)}\sum_h e^{(c^Th+v^TWh)}} \\
&= \frac{e^{(c^Th+v^TWh)}}{\sum_h e^{(c^Th+v^TWh)}} \\
&= \frac{1}{Z'}e^{(c^Th+v^TWh)} \\
&= \frac{1}{Z'}\left[e^{\left(\sum\limits_{j=1}^{n} c_jh_j+\sum\limits_{j=1}^{n} v^TW_jh_j\right)}\right] \\
&= \frac{1}{Z'}\left[\prod_{j=1}^{n} e^{\left(c_jh_j+v^TW_jh_j\right)}\right]
\end{aligned}$$

It is indeed very easy to find an unbiased sample of $\langle v_i, h_j\rangle$ since there is no connection between hidden units in an RBM. The conditional probability for a given sample **v** with binary state $h_j = 1$ is defined as follows

$$\begin{aligned}
P(h_{j=1}|v) &= \frac{P(h_{j=1},v)}{P(v)} \\
&= \frac{P(h_{j=1},v)}{\sum_{h_j} P(h_j,v)} \\
&= \frac{P(h_{j=1},v)}{P(h_{j=0},v)+P(h_{j=1},v)} \\
&= \frac{e^{(c_j+v^TW_j)}}{e^{(0)}+e^{(c_j+v^TW_j)}} \\
&= \frac{e^{(c_j+v^TW_j)}}{1+e^{(c_j+v^TW_j)}}
\end{aligned}$$

Hence,

$$P(h_{j=1}|v) = \sigma(c_j + v^TW_j) \tag{6.4}$$

where $\sigma(.)$ is the logistic sigmoid function. The conditional probability of an unbiased sample of the state of a visible unit for a given hidden vector is given by,

$$P(v_{i=1}|h) = \sigma(b_i + v^TW_j) \tag{6.5}$$

Hence, the conditional probability $P(h|v)$ and $P(v|h)$ can be defined as follows,

$$P(h|v) = \prod_{j=1}^{n} \sigma(c_j + v^T W_j),$$
$$P(v|h) = \prod_{i=1}^{n} \sigma(b_i + W_i h)$$

6.2 RBM Training

Hinton [134] proposed a training algorithm in the year 2002. He has introduced the states of the visible units to a training vector, and the hidden units are computed in parallel. After initializing the hidden units, a "reconstruction" is produced by setting each v_i to 1 with a probability given in equation 6.5. The log-likelihood function is provided by,

$$\begin{aligned} l(W, b, c) &= \sum_{t=1}^{n} \log P(v^{(t)}) \\ &= \sum_{t=1}^{n} \log \sum_{n} P(v^{(t)}) \\ &= \sum_{t=1}^{n} \log \sum_{n} \frac{1}{Z} e^{-E(h,v)}) \\ &= \sum_{t=1}^{n} \log \frac{1}{Z} \sum_{n} e^{-E(h,v)}) \\ &= \sum_{t=1}^{n} \log \left(\frac{\sum_n e^{-E(h,v)}}{Z} \right) \\ &= \sum_{t=1}^{n} \log \sum_{n} e^{-E(h,v)} - \sum_{t=1}^{n} \log Z \\ &= \sum_{t=1}^{n} \log \sum_{n} e^{-E(h,v)} - n \log Z \\ &= \sum_{t=1}^{n} \log \sum_{n} e^{-E(h,v)} - n \log \sum_{v,h} e^{-E(v,h)} \end{aligned}$$

Now, our objective is to maximize the likelihood function by calculating the partial derivative over b, c, and W. Let these parameters be represented by $\theta(b, c, W)$. Here we represent Δ as the partial derivative.

$$
\begin{aligned}
l(\theta) &= \Delta_\theta \sum_{t=1}^{n} \log \sum_{n} e^{-E(h,v)} - n\Delta_\theta \log \sum_{v,h} e^{-E(v,h)} \\
&= \sum_{t=1}^{n} \frac{\sum_n e^{-E(v,h)} \Delta_\theta \left[-E(v,h)\right]}{\sum_n e^{-E(v,h)}} - n \frac{\sum_{v,h} e^{-E(v,h)} \Delta_\theta(-E(v,h))}{\sum_{v,h} e^{-E(v,h)}} \\
&= \sum_{t=1}^{n} \mathbb{E}_{p(h,v(t))} \left[\Delta_\theta(-E(v(t),h))\right] - n\mathbb{E}_{p(h,v(t))} \left[\Delta_\theta(-E(v,h))\right]
\end{aligned}
$$

where $\mathbb{E}_{p(h,v(t))}$ is the expectation of energy with respect to h for different samples of observation.

6.3 Architecture of Deep Belief Network

Since the partition function is intractable in calculating the log-likelihood gradient of a Restricted Boltzmann Machine, contrastive divergence (CD) has been applied which is an approximation algorithm. It works in two phases, and the first phase, an input sample vector v is used in the input layer and propagated to the hidden layer to find the activations h. In the second phase, the vector h is propagated back to the visible layer and produces v'. Again, the new v' propagated back to the hidden layer and gives activations result in h'. The weight matrix gets updated by $w(t+1) = w(t) + a(vh^T - v'h'^T)$ where a is the learning rate and v, v', h, h', and w are vectors. The input layer of the first RBM is the input layer for the whole network, and the hidden layer of n^{th} RBM acts as a visible layer for $(n+1)^{th}$ RBM. The network trains the first RBM using contrastive divergence with all the training samples. Pre-training the network can be extended by connecting one or more fully connected layers to the final RBM hidden layer. The multi-layer perceptron uses back-propagation method to fine-tune the network parameters.

6.4 Proposed Architecture

Our proposed scheme has two phases, i.e., training and testing phase as shown in Figure 6.3 and has been applied on OHCS v1.0 database. Here we have gone up to three level of stacking. The module “ DBN feature extraction ” is further unboxed in Figure 6.4. The training of these three RBMs are performed sequentially. The $p(x)$ in RBM1 is calculated by the following equation,

$$p(x) = \sum_{h^1} p(x, h^1) \tag{6.6}$$

where $p(.)$ is the prediction function, $p(x, h^1)$ is the joint probability of x and h^1. It is defined as follows,

$$p(x, h^1) = p(x|h^1).p(h^1)$$

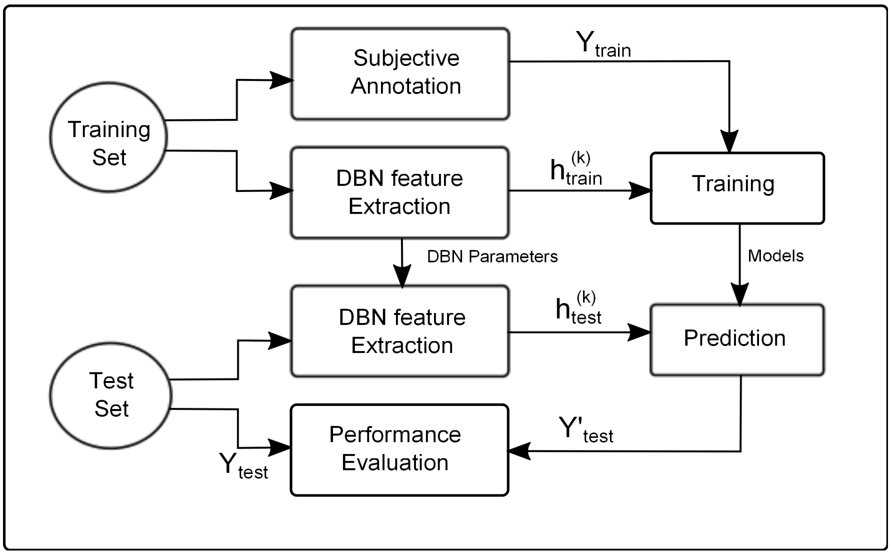

Figure 6.3: Proposed Model

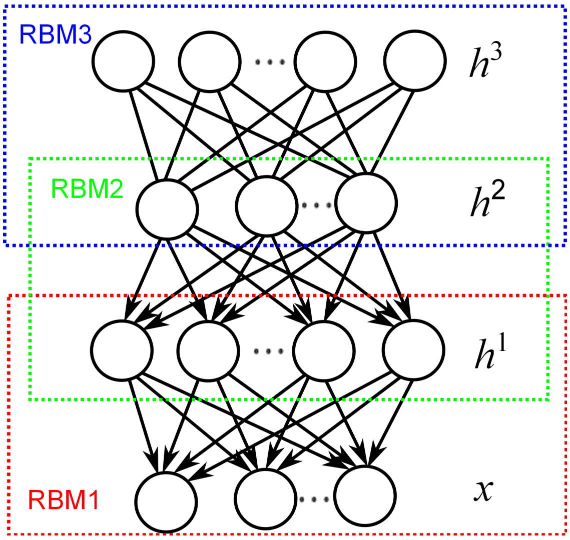

Figure 6.4: DBN Architecture used for feature extraction

where $p(h^1)$ is defined by the following equation,

$$p(h^1) = \sum_{h^2} p(h^1, h^2) \tag{6.7}$$

The value of joint probability between h^1 and h^2, i.e., $p(h^1, h^2)$ is defined by

$$p(h^1, h^2) = p(h^1|h^2) \sum_{h^3} p(h^2, h^3) \tag{6.8}$$

Hence the full distribution of a DBN is as follows

$$p(x, h^1, h^2, h^3) = p(h^2, h^3)p(h^1|h^2)p(x|h^1)$$

where,

$$p(h^2, h^3) = exp(h^{2^T} W^3 h^3 + b^{2^T} h^2 + b^{3^T} h^3)/Z,$$
$$p(h^1|h^2) = \prod_j p({h^1}_j|h^2),$$
$$p(x|h^1) = \prod_i p(x_i|h^1)$$

6.5 Results and Discussion

Implementation is carried out on a machine with following configuration: Intel(R) Core(TM) $i7 - 4770$ processor, CPU@ 3.4GHz, 6GB RAM and 64 bit operating system. The propose scheme has been simulated on MATLAB environment. Experiment has been conducted on OHCS database that consists of a total of 15040 samples having 320 patterns for each of the 47 characters. All images in the database are binarized and resized to 32×32. The whole image is converted to a one dimensional vector of length 1024. This feature vector is the input to a DBN as shown in Figure 6.5 where the number of neurons in the hidden layers are chosen experimentally. Since there are only forty seven possible outcomes, number of neurons in the output layer is 47. In the first attempt, the whole dataset is taken into consideration. The

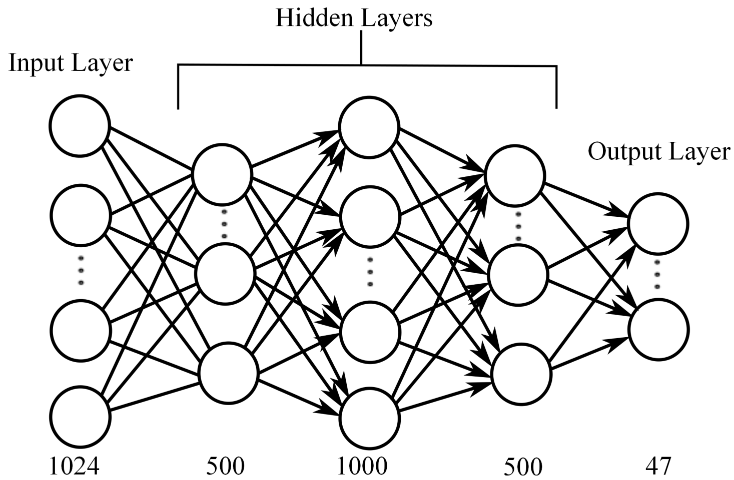

Figure 6.5: DBN Architecture configuration without Back propagation

suggested DBN network is trained without back propagation. For training 5040 samples were taken and the rest 10000 samples are used for testing. An accuracy of 87% is recorded. The back propagation algorithm is applied to fine-tune the network parameters. An accuracy of 88.2% is observed. To improve the recognition rate, the OHCS dataset is divided into two groups namely *Group I* and *Group II* as discussed in Chapter 6. The 6400 samples in *Group I* is divided into two sets with 1800 images in training and 4600 in the testing set. Similarly, out of 8640 patterns in *Group II*, only 2640 are chosen for the training set and the rest 6000 are in the testing set. Two DBN networks with back propagation have been

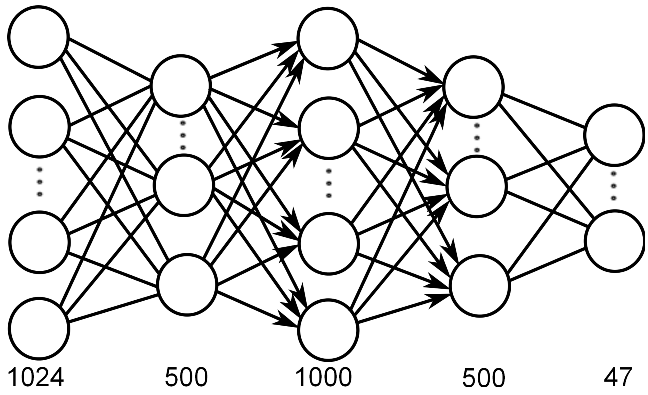

Figure 6.6: DBN Architecture configuration with Back propagation

utilized as shown in Figures 6.7 and 6.8 with the only difference in the number of neurons considered in the output layer. The accuracy recorded for DBN1 and DBN2 are 93.6% and

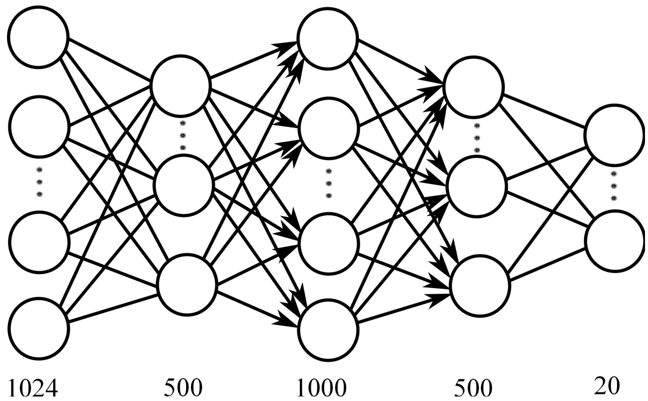

Figure 6.7: DBN1 with Back propagation

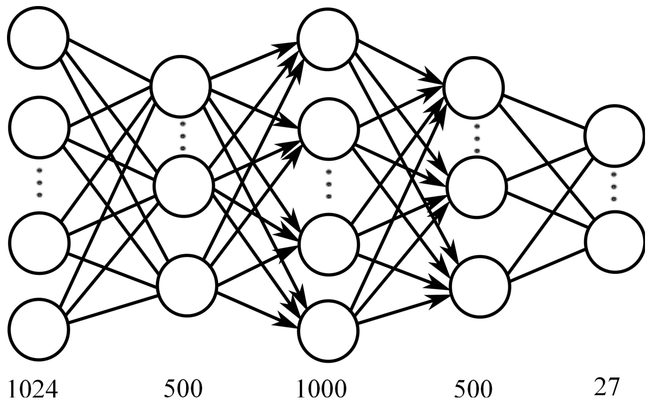

Figure 6.8: DBN2 with Back propagation

88.80% respectively. Thus, the overall accuracy is 91.20%. These features has been learnt without using their labels.

Table 6.1: Accuracy of the proposed scheme considering the whole dataset

Database	Back Propagation	Total Samples	Training Set Size	Test Set Size	Accuracy in Percentage
OHCS	No	15040	5040	10000	87.0
	Yes	15040	5040	10000	88.2

Table 6.2: Accuracy of the proposed scheme after splitting the dataset into two groups

Database	Back Propagation	Total Samples	Training Set Size	Test Set Size	Accuracy in Percentage
OHCS	Group I	6400	1800	4600	93.60
	Group II	8640	2640	6000	88.80

Table 6.3: Comparison Analysis of Suggested Schemes

Scheme	Feature Used	Feature Length	Classifier Used	Overall Accuracy
Mohanty *et al.*[85]	Structural	49	Kohonen Network	95.60%
Pal *et al.* [86]	Curvature	392	Quadratic	94.60%
Meher *et al.* [87]	Structural	Variable	BPNN	91.24%
Padhi *et al.* [136]	Structural	46	BPNN	94.00%
HOCR-DOST	multi-resolution	**256**	**BPNN**	**98.55%**
HOCR-SF	Structural	**64**	**SVM+BPNN**	**98.87%**
HOCR-DBN	**Structural**	**1024**	**DBN**	**91.20%**

6.6 Summary

This chapter provides a survey of the relevant literature on DBNs and introduces a method of recognizing the character from OHCS dataset. Also, the results of some conducted experiments using proposed model scheme extract useful features with acceptable discrimination between them without using label information. The obtained accuracy rate applying the method to the whole set was 88.2%, whereas in the second experiment we have divided the dataset into two subsets. Individual DBN networks are applied, and the overall accuracy is recorded as 91.2%.

Chapter 7

Conclusions and Future Work

7.1 Conclusion

Current research aims at developing schemes for the recognition of characters of a vernacular language because it has many regional based applications in industries and other organizations such as bank, offices, security, academic, health-care, finance, government agencies, etc. Development of a good OCR system helps people in avoiding the manual entry of relevant documents when entering them into electronic databases. No such OCR system commercially available for Odia script that makes digitized printed or handwritten text searchable with 100% accuracy. It is due to the freestyle writings of the individual and varies from person to person. Due to stumble writing or laziness, it is needless to say that the same writer's specimens on a particular character may differ at the various instances. Moreover, handwriting of different individuals varies because it influenced by many factors such as received education for writing, the quality of paper, the printing materials used, and other factors like stress, motivation and even the purpose of the handwriting. For the last two decades, researchers have been working hard in the field of Odia character recognition. Despite everything, it remains an open problem in transforming the document into its digitized form even though sophisticated devices scanners, PDAs are available.

In this thesis, attempts have been made to recognize handwritten atomic Odia character and digit. The biggest problem in the process of character recognition arises because we are dealing with handwritten characters. It has got lots of variations in shape and orientation. A massive database is required to train the network with allomorph of samples and use it for recognition. Chapter 2 describes the character set of various *Indic* languages including Odia. Also, it gives a brief description of some existing databases of few popular Indian scripts. Here, two databases namely, ODDB and OHCS for Odia digit and character respectively have been designed. It is very unlikely that the handwritten samples collected from different persons follow a similar pattern.

In chapter 3, Gabor filter array with various scales and orientations has been used as the criteria to obtain the features from the digit images. Before this, each sample in the training set of ODDB dataset is divided into four equal blocks. The Gabor filters are applied to each block, and a convolution operation is performed to get the feature vector. Using this method for all the images present in training generates a feature matrix. Further, a back propagation neural network has been trained taking the inputs from this feature matrix. It has been noticed that with five scales and 12 orientations in each scale, it gives better accuracy. In addition to this, the proposed model is also tested on standard digit databases like MNIST and USPS.

In Chapter 4, a multi-resolution technique has been considered for the recognition of characters of OHCS dataset. In the beginning, the whole dataset is divided into two groups. The structural features have been taken into consideration followed by a perceptron to solve the two class problem. However, the trained network fails in very few cases only. Also, the two-dimensional DOST coefficients are extracted from images of each group and further, PCA has been applied to reduce the feature matrix. Separate BPNN classifiers have been employed in the process of classifying the characters in each group. The experimental results verify that the proposed back propagation neural network along with SVM provides good results.

In Chapter 5, linear support vector machine (SVM) is used instead of a neural network for grouping of characters. Further, the structural features have been explored of each Odia handwritten character in each group. Multiple BPNN classifiers used and an overall accuracy recorded as 98.87% on OHCS database. The proposed scheme only takes 64 features of a character and provides the better result as compared to the previously suggested scheme over OHCS database.

A semi-supervised learning method has been proposed in Chapter 6, which works on the unlabeled dataset. Contrastive Divergence approximation algorithm has been applied to optimize the network parameters. Two experiments have been conducted for the recognition of Odia handwritten character from OHCS database. In the first attempt, the whole dataset is considered. Secondly, the database is divided into two groups and separate DBN networks have been utilized for the classification of characters in each group. Though the accuracy obtained is not better as compared to our previous schemes, but it needs no prior knowledge about the sample under testing.

7.2 Future Work

In this thesis, four different schemes have been proposed for the recognition of handwritten Odia digits and characters. Each of the schemes has been simulated separately, and the performance comparison has been made with other competent schemes. It has been observed that the suggested schemes provide better recognition accuracy as compared to the state of the art methods. The outcomes out of the thesis have brought out some useful observations and also opened up different directions of research on Odia character recognition. Further, dealing with composite character recognition is more challenging and can be taken up as a future direction of research.

Appendix

Samples of a particular Odia character

In Chapter 2, we have introduced a database OHCS v1.0 for handwritten Odia character set. It is further divided into two databases namely ODDB and OHCS. The ODDB contains all digit samples whereas OHCS keeps all character samples. Samples of the character 'ka' are shown in Figure A1 and sample set of handwritten digits are shown in Figure A2. These were used in our experiments.

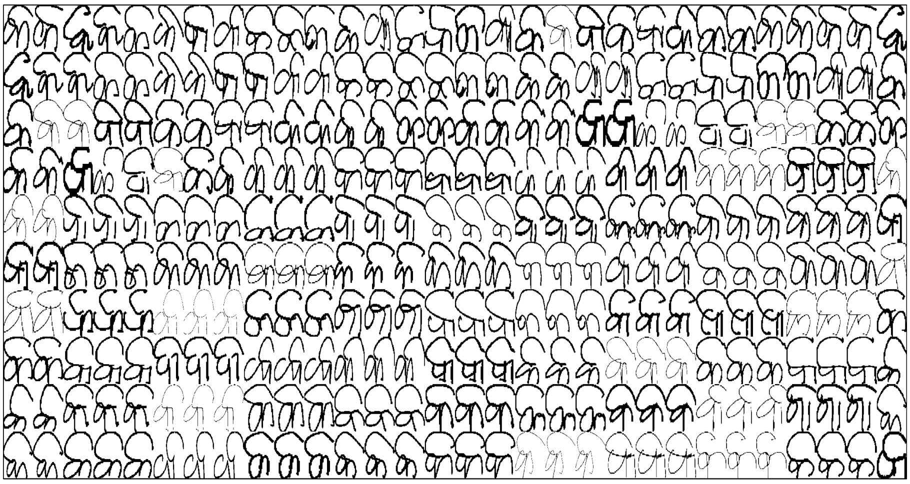

Figure A1: Three hundred sample character images of the letter ka from OHCS dataset

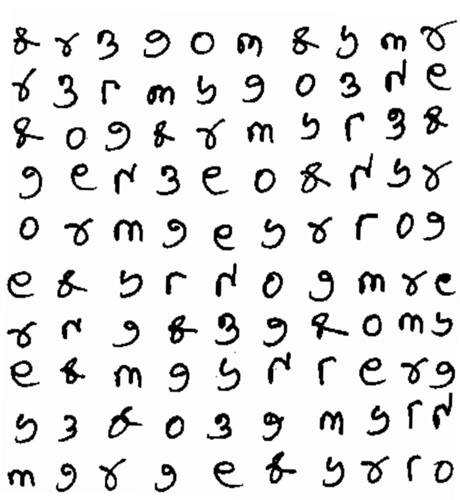

Figure A2: One hundred sample digit images of all ten digits from ODDB dataset

References

[1] L. O'Gorman and R. Kasturi, "Document Image Analysis", *IEEE Computer Society Press Los Alamitos, CA*, Vol. 39, 1995.

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

[3] S. Impedovo, L. Ottaviano, and S. Occhinegro, "Optical character recognition: a survey," in *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 5, pp. 1–24, 1991.

[4] G. Nagy, T. A. Nartker, and S. V. Rice, "Optical character recognition: An illustrated guide to the frontier," in *Electronic Imaging*. International Society for Optics and Photonics, pp. 58–69, 1999.

[5] A. Rocha and S. Goldenstein, "Classifiers and machine learning techniques for image processing and computer vision," Ph.D. dissertation, University of Campinas (Unicamp), Campinas, Brazil, 2009.

[6] S. B. Kotsiantis, "Supervised Machine Learning: A Review of Classification Techniques,", *Informatica*, vol. 31, pp. 249–268, 2007.

[7] T. E. de Campos, B. R. Babu, and M. Varma, "Character recognition in natural images." in *VISAPP (2)*, pp. 273–280, 2009.

[8] N. Arica and F. T. Yarman-Vural, "An overview of character recognition focused on off-line handwriting," in *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews,*, vol. 31, no. 2, pp. 216–233, 2001.

[9] K. S. Fu, *Applications of pattern recognition*, CRC, 1982.

[10] A. Amin, "Off-line Arabic character recognition: the state of the art," *Pattern Recognition*, vol. 31, no. 5, pp. 517–530, 1998.

[11] Y. Cao, S. Wang, and H. Li, "Skew detection and correction in document images based on straight-line fitting," in *Pattern Recognition Letters*, vol. 24, no. 12, pp. 1871–1879, 2003.

[12] G. Louloudis, B. Gatos, I. Pratikakis, and C. Halatsis, “Text line and word segmentation of handwritten documents,” *Pattern Recognition*, vol. 42, no. 12, pp. 3169–3183, 2009.

[13] B. Gatos, N. Papamarkos, and C. Chamzas, “Skew detection and text line position determination in digitized documents,” *Pattern Recognition*, vol. 30, no. 9, pp. 1505–1519, 1997.

[14] N. Nikolaou, M. Makridis, B. Gatos, N. Stamatopoulos, and N. Papamarkos, “Segmentation of historical machine-printed documents using adaptive run length smoothing and skeleton segmentation paths,” *Image and Vision Computing*, vol. 28, no. 4, pp. 590–604, 2010.

[15] Ø. D. Trier, A. K. Jain, and T. Taxt, “Feature extraction methods for character recognition-a survey,” *Pattern Recognition*, vol. 29, no. 4, pp. 641–662, 1996.

[16] M. Nixon and A. Aguado, “Feature extraction & image processing,”, *Academic Press*, 2012.

[17] Y. Yang and J. O. Pedersen, “A comparative study on feature selection in text categorization,” in *Proceedings of the Fourteenth International Conference on Machine Learning*. Morgan Kaufmann Publishers Inc., pp. 412–420, 1997.

[18] I. K. Fodor, “A survey of dimension reduction techniques,” 2002.

[19] Q. Liu and Y. Wu, “Supervised learning,” in *Encyclopedia of the Sciences of Learning*. Springer, pp. 3243–3245, 2012.

[20] C. Cortes and V. Vapnik, “Support-vector networks,” *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[21] D. Hammerstrom, “Neural networks at work,” *Spectrum, IEEE*, vol. 30, no. 6, pp. 26–32, 1993.

[22] A. K. Jain, *Fundamentals of digital image processing*. Prentice-Hall, Inc., 1989.

[23] J. R. Quinlan, “Induction of decision trees,” *Machine learning*, vol. 1, no. 1, pp. 81–106, 1986.

[24] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification*. John Wiley & Sons, 2012.

[25] A. P. James and S. Dimitrijev, “Nearest neighbor classifier based on nearest feature decisions,” *The Computer Journal*, pp. bxs001, 2012.

[26] V. Patil and S. Shimpi, "Handwritten english character recognition using neural network," *Elixir Comp. Sci. & Engg*, vol. 41, pp. 5587–5591, 2011.

[27] C. C. Tappert, C. Y. Suen, and T. Wakahara, "Online handwriting recognition-a survey," in *9th International Conference on Pattern Recognition*, IEEE, pp. 1123–1132, 1988.

[28] ——, "The state of the art in online handwriting recognition," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 12, no. 8, pp. 787–808, 1990.

[29] R. Plamondon and S. N. Srihari, "Online and off-line handwriting recognition: a comprehensive survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 63–84, 2000.

[30] V. Govindan and A. Shivaprasad, "Character recognition—a review," *Pattern recognition*, vol. 23, no. 7, pp. 671–683, 1990.

[31] U. Pal and B. Chaudhuri, "Indian script character recognition: a survey," *Pattern Recognition*, vol. 37, no. 9, pp. 1887–1899, 2004.

[32] T. Akiyama and N. Hagita, "Automated entry system for printed documents," *Pattern Recognition*, vol. 23, no. 11, pp. 1141–1154, 1990.

[33] D. C. Álvarez, F. M. Rodríguez, and X. F. Hermida, "Printed and handwritten digits recognition using neural networks," in *Proceedings of International Conference on Signal Processing and Applications Technology*, 1998.

[34] U. Pal and B. Chaudhuri, "Machine-printed and hand-written text lines identification," *Pattern Recognition Letters*, vol. 22, no. 3–4, pp. 431–441, 2001.

[35] D. V. Sharma, G. S. Lehal, and P. Kathuria, "Digit extraction and recognition from machine printed Gurmukhi documents," in *Proceedings of the International Workshop on Multilingual OCR*. ACM, no. 12, pp. 1–8, 2009.

[36] B. Dhandra, G. Mukarambi, and M. Hangarge, "Zone based features for handwritten and printed mixed kannada digits recognition," in *IJCA Proceedings on International Conference on VLSI, Communications and Instrumentation (ICVCI)*, no. 7, pp. 5–8, 2011.

[37] M. Maloo and K. Kale, "Support vector machine based gujarati numeral recognition," *International Journal on Computer Science and Engineering*, vol. 3, no. 7, pp. 2595–2600, 2011.

[38] S. Rajasekaran and B. Deekshatulu, "Recognition of printed telugu characters," *Computer Graphics and Image Processing*, vol. 6, no. 4, pp. 335–360, 1977.

[39] G. Siromoney, R. Chandrasekaran, and M. Chandrasekaran, "Computer recognition of printed tamil characters," *Pattern Recognition*, vol. 10, no. 4, pp. 243–247, 1978.

[40] B. Chaudhuri, U. Pal, and M. Mitra, "Automatic recognition of printed oriya script," *Sadhana*, vol. 27, no. 1, pp. 23–34, 2002.

[41] U. Pal and B. Chaudhuri, "Automatic recognition of unconstrained off-line Bangla handwritten numerals," in *Advances in Multimodal Interfaces—ICMI 2000*. Springer, pp. 371–378, 2000.

[42] Y. Lecun and C. Cortes, "The MNIST database of handwritten digits," 2009. [Online]. Available: http://yann.lecun.com/exdb/mnist/

[43] J. W. Xu, J. Xu, and Y. Lu, "Handwritten Bangla digit recognition using hierarchical bayesian network," in *3rd International Conference on Intelligent System and Knowledge Engineering*, IEEE, vol. 1. , pp. 1096–1099, 2008.

[44] H. A. Khan, A. Al Helal, and K. I. Ahmed, "Handwritten Bangla digit recognition using sparse representation classifier," in *International Conference on Informatics, Electronics & Vision (ICIEV)*, IEEE, pp. 1–6, 2014.

[45] S. Basu, R. Sarkar, N. Das, M. Kundu, M. Nasipuri, and D. K. Basu, "Handwritten bangla digit recognition using classifier combination through ds technique," in *Proceedings of the First International Conference on Pattern Recognition and Machine Intelligence*, ser. PReMI'05. Berlin, Heidelberg: Springer-Verlag, pp. 236–241, 2005.

[46] S. Basu, N. Das, R. Sarkar, M. Kundu, M. Nasipuri, and D. K. Basu, "An mlp based approach for recognition of handwritten 'Bangla' numerals," *arXiv preprint arXiv:1203.0876*, 2012.

[47] T. Hassan and H. A. Khan, "Handwritten bangla numeral recognition using local binary pattern," in *International Conference on Electrical Engineering and Information Communication Technology (ICEEICT)*, IEEE, pp. 1–4, 2015.

[48] N. Das, S. Pramanik, S. Basu, P. K. Saha, R. Sarkar, M. Kundu, and M. Nasipuri, "Recognition of handwritten Bangla basic characters and digits using convex hull based feature set," *arXiv preprint arXiv:1410.0478*, pp. 380–386, 2014.

[49] K. Roy, T. Pal, U. Pal, and F. Kimura, "Oriya handwritten numeral recognition system," in *Eighth International Conference on Document Analysis and Recognition*. IEEE, pp. 770–774, 2005.

[50] T. K. Bhowmik, S. K. Parui, U. Bhattacharya, and B. Shaw, "An hmm based recognition scheme for handwritten oriya numerals," in *9th International Conference on Information Technology*. IEEE, pp. 105–110, 2006.

[51] P. K. Sarangi, A. K. Sahoo, and P. Ahmed, "Recognition of isolated handwritten oriya numerals using hopfield neural network," *International Journal of Computer Applications*, vol. 40, no. 8, pp. 36–42, 2012.

[52] J. Panda, M. Panda, A. Samal, and N. Das, "Odia handwritten digit recognition using single layer perceptron," *International Journal Of Electronics And Communication Engineering & Technology*, vol. 5, no. 4, pp. 80–88, 2014.

[53] K. S. Dash, N. Puhan, and G. Panda, "A hybrid feature and discriminant classifier for high accuracy handwritten odia numeral recognition," in *IEEE Region 10 Symposium*, IEEE, pp. 531–535, 2014.

[54] T. K. Mishra, B. Majhi, P. K. Sa, and S. Panda, "Model based odia numeral recognition using fuzzy aggregated features," *Frontiers of Computer Science*, vol. 8, no. 6, pp. 916–922, 2014.

[55] P. Pujari and B. Majhi, "A comparative study of classifiers on recognition of offline handwritten odia numerals," in *International Conference on Electrical, Electronics, Signals, Communication and Optimization (EESCO)*. IEEE, pp. 1–5, 2015.

[56] K. Dash, N. Puhan, and G. Panda, "Non-redundant stockwell transform based feature extraction for handwritten digit recognition," in *International Conference on Signal Processing and Communications (SPCOM)*, pp. 1–4, July 2014.

[57] A. Amin and W. Wilson, "Hand-printed character recognition system using artificial neural networks," in *Proceedings of the Second International Conference on Document Analysis and Recognition*, IEEE, pp. 943–946, 1993.

[58] B. S. Jeng, T.-M. Shieh, C.-S. Miou, C.-J. Lee, B.-S. Chien, Y.-H. Hu, and G.-H. Chang, "Automated entry system for chinese printed documents," *Image and Vision Computing*, vol. 13, no. 10, pp. 745 – 754, 1995.

[59] H. P. Chiu and D.-C. Tseng, "Invariant handwritten chinese character recognition using fuzzy min-max neural networks," *Pattern Recognition letters*, vol. 18, no. 5, pp. 481–491, 1997.

[60] M. S. Khorsheed, "Off-line arabic character recognition–a review," *Pattern analysis & applications*, vol. 5, no. 1, pp. 31–45, 2002.

[61] A. M. Elgammal, M. Ismail *et al.*, "Techniques for language identification for hybrid arabic-english document images," in *Sixth International Conference on Document Analysis and Recognition*. IEEE, pp. 1100–1104, 2001.

[62] H. A. Al-Jamimi and S. A. Mahmoud, "Arabic character recognition using gabor filters," in *Innovations and Advances in Computer Sciences and Engineering*. Springer, pp. 113–118, 2010.

[63] J. H. AlKhateeb, O. Pauplin, J. Ren, and J. Jiang, "Performance of Hidden Markov Model and dynamic Bayesian network classifiers on handwritten Arabic word recognition," *knowledge-based systems*, vol. 24, no. 5, pp. 680–688, 2011.

[64] I. Bazzi, R. Schwartz, and J. Makhoul, "An omnifont open-vocabulary ocr system for english and arabic," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 6, pp. 495–504, 1999.

[65] S. Mori, K. Yamamoto, and M. Yasuda, "Research on machine recognition of handprinted characters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 4, pp. 386–405, 1984.

[66] E. Borovikov, "A survey of modern optical character recognition techniques," *arXiv preprint arXiv:1412.4183*, 2014.

[67] I. K. Sethi and B. Chatterjee, "Machine recognition of constrained hand printed devanagari," *Pattern Recognition*, vol. 9, no. 2, pp. 69–75, 1977.

[68] R. Sinha and H. Mahabala, "Machine recognition of devanagari script," *IEEE Trans. Syst. Man Cybern*, vol. 9, no. 8, pp. 435–441, 1979.

[69] S. D. Connell, R. Sinha, and A. K. Jain, "Recognition of unconstrained online devanagari characters," in *15th International Conference on Pattern Recognition*, vol. 2. IEEE, pp. 368–371, 2000.

[70] P. Khanale and S. Chitnis, "Handwritten devanagari character recognition using artificial neural network," *Journal of Artificial Intelligence*, vol. 4, no. 1, pp. 55–62, 2011.

[71] A. Majumdar, "Bangla basic character recognition using digital curvelet transform," *Journal of Pattern Recognition Research*, vol. 2, no. 1, pp. 17–26, 2007.

[72] U. Bhattacharya, M. Shridhar, S. K. Parui, P. Sen, and B. Chaudhuri, "Offline recognition of handwritten Bangla characters: an efficient two-stage approach," *Pattern Analysis and Applications*, vol. 15, no. 4, pp. 445–458, 2012.

[73] S. Basu, N. Das, R. Sarkar, M. Kundu, M. Nasipuri, and D. K. Basu, "Handwritten bangla alphabet recognition using an mlp based classifier," *arXiv preprint arXiv:1203.0882*, 2012.

[74] S. Rajashekararadhya and P. V. Ranjan, "Neural network based handwritten numeral recognition of kannada and telugu scripts," in *IEEE Region 10 Conference, TENCON*, IEEE, pp. 1–5, 2008.

[75] A. K. Pujari, C. D. Naidu, M. S. Rao, and B. Jinaga, "An intelligent character recognizer for telugu scripts using multiresolution analysis and associative memory," *Image and Vision Computing*, vol. 22, no. 14, pp. 1221–1227, 2004.

[76] U. Bhattacharya, S. Ghosh, and S. K. Parui, "A two stage recognition scheme for handwritten tamil characters," in *Ninth International Conference on Document Analysis and Recognition*, vol. 1. IEEE, pp. 511–515, 2007.

[77] N. Shanthi and K. Duraiswamy, "A novel svm-based handwritten Tamil character recognition system," *Pattern Analysis and Applications*, vol. 13, no. 2, pp. 173–180, 2010.

[78] J. R. Prasad, U. Kulkarni, and R. S. Prasad, "Offline handwritten character recognition of gujrati script using pattern matching," in *3rd International Conference on Anti-counterfeiting, Security, and Identification in Communication*. IEEE, pp. 611–615, 2009.

[79] A. A. Desai, "Gujarati handwritten numeral optical character reorganization through neural network," *Pattern Recognition*, vol. 43, no. 7, pp. 2582–2589, 2010.

[80] M. Kumar, M. Jindal, and R. Sharma, "k-nearest neighbor based offline handwritten gurmukhi character recognition," in *International Conference on Image Information Processing (ICIIP)*. IEEE, pp. 1–4, 2011.

[81] K. S. Siddharth, M. Jangid, R. Dhir, and R. Rani, "Handwritten Gurmukhi character recognition using statistical and background directional distribution." in *International Journal on Computer Science and Engineering (IJCSE)*, vol. 3, no. 06, pp. 2332–2345, 2011.

[82] M. Kumar, R. K. Sharma, and M. K. Jindal, "Offline handwritten gurmukhi character recognition: Study of different feature-classifier combinations," in *Proceeding of the Workshop on Document Analysis and Recognition*, ser. DAR '12. NY, USA: ACM, pp. 94–99, 2012.

[83] A. Aggarwal and K. Singh, "Handwritten gurmukhi character recognition," in *International Conference on Computer, Communication and Control (IC4)*, pp. 1–5, Sept 2015.

[84] K. Verma and R. Sharma, "Performance analysis of zone based features for online handwritten gurmukhi script recognition using support vector machine," in *Progress in Systems Engineering*, Springer, pp. 747–753, 2015.

[85] S. Mohanty, "Pattern recognition in alphabets of oriya language using kohonen neural network," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 12, no. 07, pp. 1007–1015, 1998.

[86] U. Pal, T. Wakabayashi, and F. Kimura, "A system for off-line oriya handwritten character recognition using curvature feature," in *10th International Conference on Information Technology*. IEEE, pp. 227–229, 2007.

[87] S. Meher and D. Basa, "An intelligent scanner with handwritten odia character recognition capability," in *Fifth International Conference on Sensing Technology (ICST)*. IEEE, pp. 53–59, 2011.

[88] B. Kumar, N. Kumar, C. Palai, P. K. Jena, and S. Chattopadhyay, "Optical character recognition using ant miner algorithm: a case study on Oriya character recognition," *International Journal of Computer Applications*, vol. 61, no. 3, pp. 17–22, 2013.

[89] [Online].Available:http://www.census2011.co.in/census/state/orissa.html

[90] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," *10th European Conference on Machine Learning Chemnitz, Germany*, Springer, vol. 1398, pp. 137–141, 1998.

[91] J. A. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural processing letters*, vol. 9, no. 3, pp. 293–300, 1999.

[92] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, vol. 5, no. 4, pp. 115–133, 1943.

[93] F. Rosenblatt, *The perceptron, a perceiving and recognizing automaton Project Para*. Cornell Aeronautical Laboratory, 1957.

[94] J. A. Anderson, "A simple neural network generating an interactive memory," *Mathematical Biosciences*, vol. 14, no. 3, pp. 197–220, 1972.

[95] T. Kohonen, "Correlation matrix memories," *IEEE Transactions on Computers*, vol. 100, no. 4, pp. 353–359, 1972.

[96] G. A. Carpenter and S. Grossberg, “A massively parallel architecture for a self-organizing neural pattern recognition machine,” *Computer Vision, Graphics, and Image Processing*, vol. 37, no. 1, pp. 54–115, 1987.

[97] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, “Backpropagation applied to handwritten zip code recognition,” *Neural Computation*, vol. 1, no. 4, pp. 541–551, 1989.

[98] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Cognitive modeling*, vol. 5, no. 3, pp. 213–222, 1988.

[99] Y. Lecun and C. Cortes, “The MNIST database of handwritten digits.” [Online]. Available: http://yann.lecun.com/exdb/mnist/

[100] M. Lichman, “UCI machine learning repository,” 2013. [Online]. Available: http://archive.ics.uci.edu/ml

[101] U. Bhattacharya and B. B. Chaudhuri, “Databases for research on recognition of handwritten characters of indian scripts,” in *Eighth International Conference on Document Analysis and Recognition*, IEEE, pp. 789–793, 2005.

[102] ——, “Handwritten numeral databases of indian scripts and multistage recognition of mixed numerals,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 3, pp. 444–457, 2009.

[103] J.-S. Lee, “Digital image enhancement and noise filtering by use of local statistics,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-2, no. 2, pp. 165–168, March 1980.

[104] B. Su, S. Lu, and C. L. Tan, “Robust document image binarization technique for degraded document images,” *IEEE Transactions on Image Processing*, vol. 22, no. 4, pp. 1408–1417, 2013.

[105] A. Amin and S. Fischer, “A document skew detection method using the hough transform,” *Pattern Analysis and Applications*, vol. 3, no. 3, pp. 243–253, 2000.

[106] J. Benesty, J. Chen, Y. A. Huang, and S. Doclo, “Study of the wiener filter for noise reduction,” in *Speech Enhancement*. Springer, pp. 9–41, 2005.

[107] W. K. Pratt, “Generalized wiener filtering computation techniques,” *IEEE Transactions on Computers*, vol. 100, no. 7, pp. 636–641, 1972.

[108] MATLAB, *version 8.3.0.532 (R2014a)*. Natick, Massachusetts: The MathWorks Inc., 2014.

[109] R. C. Gonzalez and R. E. Woods, *Digital Image Processing (3rd Edition)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2006.

[110] T. P. Weldon, W. E. Higgins, and D. F. Dunn, "Gabor filter design for multiple texture segmentation," *Optical Engineering*, vol. 35, no. 10, pp. 2852–2863, 1996.

[111] L. Shen and L. Bai, "A review on Gabor wavelets for face recognition," *Pattern Analysis and Applications*, vol. 9, no. 2-3, pp. 273–292, 2006.

[112] D. Gabor, "Theory of communication. part 1: The analysis of information," *Journal of the Institution of Electrical Engineers-Part III: Radio and Communication Engineering*, vol. 93, no. 26, pp. 429–441, 1946.

[113] J. G. Daugman, "Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters," *JOSA A*, vol. 2, no. 7, pp. 1160–1169, 1985.

[114] M. Haghighat, S. Zonouz, and M. Abdel-Mottaleb, "Identification using encrypted biometrics," in *International Conference on Computer Analysis of Images and Patterns*. Springer, pp. 440–448, 2013.

[115] N. Otsu, "A threshold selection method from gray-level histograms," *Automatica*, vol. 11, no. 285-296, pp. 23–27, 1975.

[116] Y. Liu and S. N. Srihari, "Document image binarization based on texture features," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 5, pp. 540–544, 1997.

[117] Y. Wang and J. Orchard, "The discrete orthonormal stockwell transform for image restoration," in *16th IEEE International Conference on Image Processing (ICIP)*. IEEE, pp. 2761–2764, 2009.

[118] R. G. Stockwell, L. Mansinha, and R. Lowe, "Localization of the complex spectrum: the s transform," *IEEE Transactions on Signal Processing*, vol. 44, no. 4, pp. 998–1001, 1996.

[119] R. G. Stockwell, "A basis for efficient representation of the s-transform," *Digital Signal Processing*, vol. 17, no. 1, pp. 371–393, 2007.

[120] Y. Wang and J. Orchard, "Fast discrete orthonormal Stockwell transform," *SIAM Journal on Scientific Computing*, vol. 31, no. 5, pp. 4000–4012, 2009.

[121] H. Freeman, "On the encoding of arbitrary geometric configurations," *IRE Transactions on Electronic Computers*, no. 2, pp. 260–268, 1961.

[122] A. Manjunath and K. Hemantha, "Principal component analysis and generalized regression neural networks for efficient character recognition," in *First International Conference on Emerging Trends in Engineering and Technology*. IEEE, pp. 1170–1174, 2008.

[123] U. Pal, N. Sharma, T. Wakabayashi, and F. Kimura, "Handwritten numeral recognition of six popular indian scripts," in *Ninth International Conference on Document Analysis and Recognition*,IEEE, vol. 2. , pp. 749–753, 2007.

[124] K. Pearson, "Principal components analysis," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 6, no. 2, p. 559, 1901.

[125] H. Hotelling, "Analysis of a complex of statistical variables into principal components." *Journal of educational psychology*, vol. 24, no. 6, pp. 417–441, 1933.

[126] ——, "Simplified calculation of principal components," *Psychometrika*, vol. 1, no. 1, pp. 27–35, 1936.

[127] R. Salakhutdinov and I. Murray, "On the quantitative analysis of deep belief networks," in *Proceedings of the 25th international conference on Machine learning*. ACM, pp. 872–879, 2008.

[128] Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle *et al.*, "Greedy layer-wise training of deep networks," *Advances in neural information processing systems*, vol. 19, pp. 153–160, 2007.

[129] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[130] A. Fischer and C. Igel, "An introduction to restricted boltzmann machines," in *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, Springer, pp. 14–36, 2012.

[131] G. E. Hinton , S. Osindero and Y. W. Teh, "A fast learning algorithm for deep belief nets," in *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[132] A. R. Mohamed, G. Dahl, and G. Hinton, "Deep belief networks for phone recognition." in *Nips workshop on deep learning for speech recognition and related applications*, vol. 1, no. 9, pp. 1–9, 2009.

[133] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, pp. 609–616, 2009.

[134] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Computation*, vol. 14, no. 8, pp. 1771–1800, 2002.

[135] H. S. Cruz, E. Bribiesca, and R. M. R. Dagnino, "Efficiency of chain codes to represent binary objects," *Pattern Recognition*, vol. 40, no. 6, pp. 1660–1674, 2007.

[136] D. Padhi, "Novel Hybrid approach for Odia Handwritten Character Recognition System," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 2, no. 5, pp. 150–157, 2012.

Dissemination

Journal Articles

1. **Ramesh Kumar Mohapatra**, Banshidhar Majhi and Sanjay Kumar Jena. A Two-Stage Classification Approach for the Recognition of Handwritten Atomic Odia Character. *Int. J. Applied Pattern Recognition, Inderscience Publishers*,Vol. 3, No. 4, pp. 326–345, 2016.

2. **Ramesh Kumar Mohapatra**, Banshidhar Majhi and Sanjay Kumar Jena. Classification of Handwritten Odia Basic Character using Stockwell Transform. *Int. J. Applied Pattern Recognition, Inderscience Publishers*,Vol. 2, No. 3, pp. 235–254, 2015.

Conferences

1. **Ramesh Kumar Mohapatra**, Banshidhar Majhi, and Sanjay Kumar Jena. Classification performance analysis of MNIST Dataset utilizing a Multi-resolution Technique. In *International Conference on Computing, Communication and Security* , pp. 1–5, IEEE, 2015.

2. **Ramesh Kumar Mohapatra**, Tushar Kanti Mishra, Sandeep Panda, and Banshidhar Majhi. OHCS : A Database for Handwritten Atomic Odia Character Recognition. In *Proceedings of Fifth National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG)*, pp. 1–4, IEEE, 2015.

Article under preparation

1. **Ramesh Kumar Mohapatra**, Banshidhar Majhi, and Sanjay Kumar Jena. A semi-supervised learning based handwritten Odia character Recognition.

2. **Ramesh Kumar Mohapatra**, Banshidhar Majhi, and Sanjay Kumar Jena. Design and Analysis of Handwritten Odia Digit Recognition utilizing Gabor Filter Array.

3. **Ramesh Kumar Mohapatra**, Banshidhar Majhi, and Sanjay Kumar Jena. A GUI model Designed for the Evaluation of Simple Odia Handwritten Arithmetic Equation.

Index