Optical Character Recognition of Machine Printed Dogri Language Documents

A Thesis

Submitted in fulfilment of the

requirement for the award of the degree of

Doctor of Philosophy

Submitted by

Khushneet Jindal
(Registration No. 951211001)

Department of Computer Science and Engineering
Thapar Institute of Engineering and Technology
Patiala-147 004, Punjab, India

July, 2018

Dedicated to

My Parents

&

My Family

Certificate

I hereby certify that the work which is being presented in this thesis entitled, "**Optical Character Recognition of Machine Printed Dogri Language Documents**", in fulfilment of the requirements for the award of the degree of **Doctor of Philosophy** submitted in the Computer Science and Engineering Department, Thapar Institute of Engineering and Technology, Patiala, is an authentic record of my own work carried out under the supervision of Dr. Rajiv Kumar, Associate Professor, Computer Science and Engineering Department, Thapar Institute of Engineering and Technology (TIET), Patiala. The ideas and references cited herein have been duly acknowledged.

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

Khushneet Jindal

(Khushneet Jindal)
Registration No. 951211001

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

Rajiv Kumar

Dr. Rajiv Kumar Sharma

Associate Professor
Computer Science and Engineering Department
Thapar Institute of Engineering and Technology, Patiala-147004
Punjab, INDIA
Supervisor

Acknowledgements

First and foremost, I express my deep and sincere gratitude to my supervisor Dr. Rajiv Kumar Sharma, Associate Professor, Department of Computer Science and Engineering, TIET, Patiala for their expert guidance, valuable suggestions, support and continuous encouragement throughout the period of my research work. The imperative comments, rendered by him during the discussions are deeply appreciated.

My heartfelt thanks go to Dr. N.Tejo Parkash (Professor, SEE, TIET, Patiala), Dr. Sunil Singla (Associate Professor, EIED, TIET, Patiala), Dr. Amit Kumar (Associate Professor, SoM, TIET, Patiala), Dr. Raj Kumar Gupta (Professor, CED, TIET, Patiala), Dr. Rajesh Kumar Gupta (Associate Professor, Central University, Bathinda), Dr. Harish Garg (Assistant Professor, SoM, TIET, Patiala), Dr. Ankur Rana (Programmer, Punjabi University, Patiala) and Vasu Bhushan (nephew) for their helpful and valuable suggestions. Also, I am thankful to Dr. S.S. Bhatia (DoAA, TIET, Patiala), Dr. R.K. Sharma (DoFA, TIET, Patiala), Dr. Rajesh Kumar (Professor, CSED, TIET, Patiala), Dr. A.K Lal (Professor, SoM, TIET, Patiala), Dr. Sharad Saxena (Associate Professor, CSED, TIET, Patiala) and Dr. Maninder Singh (Head, CSED, TIET, Patiala) for their continuous feedback and moral support during my work.

For my wife Bably, I find it difficult to express my appreciation because it is so boundless. She has given up so much to make my career a priority, she has been with me through the ups and downs of my entire Ph.D. work. Without her love and support this thesis would have never been possible. I owe precious time of my lovely kids Pinank and Nivedita (Pihu), for my research work, which otherwise, I have enjoyed

with them. I cannot bring back those days but now, I will give them maximum of my time.

I am forever grateful to my parents, who mean world to me. I pay regards from the core of my heart to my mother Mrs. Kamla Devi and father Mr. Krishan Lal Jindal for their love and blessings. I cannot forget to remember timely help and continuous guidance of my elder brother Mr. Lalit Jindal. Also, I never forget to remember my grandfather late Shri. Bant Ram ji, from whom I learned a lot. I inherited his spirit of hardworking and never give up approach. Today, what I am is due to his blessings only.

I am always thankful to late Prof. Gursewak Singh, former Vice-Chancellor of Punjabi University, Patiala for his blessings and support. Also, my sincere thanks to Dr. Manmohan Singh, DM Cardiology, Sehat Medicare, Patiala for his blessings, guidance and endless support.

Last but not the least, I am ever grateful to God, the Creator, Guardian and to whom I owe my very existence. Thank you God for the numerous blessings bestowed upon me in every aspect of my life.

I am highly thankful to Prof. (Dr.) Parkash Gopalan, Director, Thapar Institute of Engineering and Technology, Patiala, Prof. (Dr.) R.S. Kaler, Thapar Institute of Engineering and Technology, Patiala, Dr. Rafat Siddique, Dean of Research & Sponsored Projects, Thapar Institute of Engineering and Technology, Patiala and Dr. Gurbinder Singh, Registrar, Thapar Institute of Engineering and Technology, Patiala for providing me the opportunity to pursue my course work and research.

Patiala (**Khushneet Jindal**)

July, 2018

Abstract

Optical character recognition (OCR) is a technology used for the digitization of printed historical documents, books, magazines, manuscripts etc., in order to preserve them from deterioration. In India, there are a number of language groups, the major ones being the Indo-Aryan languages, spoken by most of the Indians. One such language is Dogri which is written using Devanagari script and is one of the important Indian language used in the border areas of the North of India. The present research work is an exclusive attempt towards the design and development of an OCR for recognition of machine printed Dogri language documents. A new dataset of Dogri language characters has been prepared as standard dataset for Dogri language OCR was non-existent. The new dataset consists of around 87000 character images collected from old books, magazines, newspaper and synthetically generated data. A novel shape based algorithms have been proposed for the segmentation of lines, words, characters and modifiers for the printed Dogri language documents. The proposed algorithms mainly focused upon the structure of character by retaining the header line (Shirorekha) during segmentation which is importantly required to minimize the loss of structural information. It helps in minimizing the chances of under-segmentation or over-segmentation. A segmentation accuracy of around 99.46% at character level has been achieved using proposed algorithms. The results showed that the proposed algorithms not only successfully resolves identified shortcomings that occur due to structural loss, but are also time efficient than the other methods. Moreover, when the proposed algorithms were applied on the pre-detected words of Devanagari script based natural scene images, it has been found that there was an enhanced accuracy of 36.34% at

character level with 56% lesser processing time than the method in vogue. The proposed algorithms successfully segments almost all the cases where existing algorithms failed to segment, under or over segment the text image.

For the recognition of characters initially shape oriented features have been extracted using Discrete Cosine Transformation (DCT), Gradient and Zernike Moments feature extraction techniques. The performance of these techniques is evaluated in terms of attributes and length of features. The effectiveness of shape based features has been analysed in recognition stage using various combinations classification techniques. Around 200 features have been extracted in zig-zag manner from each of the image of size 32x32.

The characters recognition has been performed using various combinations of extracted features and Multilayer perceptron neural networks (MPNN), Support Vector Machines (SVM) & *k*-Nearest Neighbors (*k*-NN) classification techniques. For experimentation, the datasets was partitioned in the ratio of 75:25 i.e, 75% data has been used for training and remaining 25% for testing the classifier. The proposed character recognition system has achieved an impressive accuracy of 98.56% to 99.10% (depending upon the classifier used) the best reported till date. Further, the maximum character recognition accuracy of 99.10% was also achieved with the combination of Gradient features and Support vector machine.

Then, a dictionary based post-processing technique has been applied for the correction of errors left by the classification stage. A corpus containing around fifty lac words of Dogri and Hindi language text has been formulated from online books, documents, magazines, newspaper etc. The output of post-processor has been manually

checked at character level on five Dogri language documents and the results were matched with the actual printed document.

Finally, chapter six presents the inferences drawn from the results of the various experiments carried out in this work. Also, some future research directions on the line of this work are discussed briefly.

Contents

List of Figures

List of Tables

Chapter 1
Introduction

Early human civilizations used stones, wood, as well as leaves of banana and palm as surfaces for writing and storing information. This helped in transfer of valuable historical information from one generation to another. Over time, the advancement in writing material for information storage continued leading to preserving information on papers. Over the past few decades, with more advancement of technology, the information is getting stored on computers in digital form, for its long term usage and availability. There are a number of advantages of storing information in digital format. One of the advantages of digital storage is that a large number of people can have access to this digital information with the click of a button. Other benefits of information digitization include paper saving, huge volume of data storage in a small storage device, in addition to easier and wider access.

Huge information is available in the form of old books, documents, historical information etc., which is required to be digitized for its effective access. Now, the question arises that how to digitize historic information which is available in the form of paper or pictures for its effective access and reliable storage. A common solution for this problem is to manually digitize (by typing) the data through data entry operation. However this task is time consuming and requires lot of human effort. Another effective solution to accomplish this task is by scanning or capturing the pictures of the documents and then converting those images into digital format using image processing and character recognition techniques. A revolutionary technique known as Optical Character Recognition (OCR) can be used for transforming

document pictures containing printed or handwritten text into digital format. OCR is globally used for the conversion of machine printed books, magazines and documents into digital format. It minimizes the human effort, takes less time and much lesser storage space. The converted digitized text can be edited, searched and accessed on-line, widely accessed on-line, and can be utilized in applications like the translation by computer, text into speech and analyzing large amount of data in a database etc., within less storage space. A number of OCR systems have already been implemented in various languages, including Indian languages. However, there is a large volume of text documents in diverse variety of languages that are required to be digitized for wider access and longtime availability. One such Indian language is Dogri and as per the information available, there is no recognition system that can exclusively, accurately and efficiently digitize the machine printed Dogri language documents. In light of this, the proposed work is focused on the design and development of an OCR for machine printed Dogri language documents.

1.1 Preamble

OCR is an application software, which can convert machine printed or handwritten document images into digitally editable formats. It is a language specific application software which helps to digitizing documents in a short span of time and with less human effort. The input to an OCR system is a scanned or camera captured image, which is pre-processed to remove anomalies like background noise, skewness, multiple gray levels, etc. The preprocessed image is then segmented and the features are extracted from sub images, which are later classified by the recognition system. The output produced is a digital text that can be edited, searched and stored on a media. In the early days, OCR was available for very few languages such as English

and others that are of European origin. However, within a few years, it has become a very popular medium of digitization. The main reasons behind its popularity are the savings in terms of effort, time and cost. Now, a day, OCR is being used all over the world for many languages. The overview of an OCR is shown in Figure 1.1.

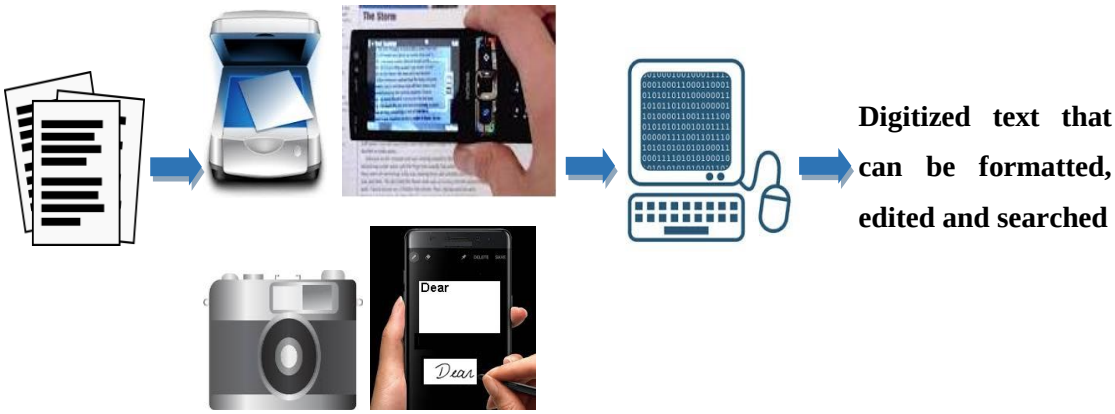


Figure 1.1 Overview of an OCR system

OCR technique has a number of applications being used in daily life, such as automated processing of bank cheques and postal mails, office automation (data entry of text documents into electronic format) and various applications for smart phones (vcard, city name and phone number scanner), machine translation, text to speech for the blind and visually impaired users.

1.2 Historical Background of Character Recognition Scripts

In 1929 an Austrian engineer Gustav Tauschek invented OCR (a mechanical device known as reading machine), using template matching with a photodetector (Handley, 1998). Tauschek was the first person to obtain a patent in Germany for the OCR machine. Later, Paul Handel obtained a patent (US patent number US1915993 A) on OCR (Paul, 1933), followed by grant of US patent (1935) to Tauschek.

In late 1953, a cryptanalyst David H. Shepard at US armed forces, in association with Harvey Cook, invented a reading machine known as GISMO (Shepard et al.,

1953) and obtained US patent number 2,663,758. Thereafter, a corporation known as Intelligent Machines Research (IMR) was started and designed world's first several commercially operational character recognition system.

A postal service of the United States has started using OCR since 1965 to sort mail letters based on the technique primarily invented by Jacob Rabinow. In Europe, the British General Post Office was first to start using OCR technology. The United Kingdom and Canada had also started using OCR technology in banks for the bill payment system, during the same period.

Based on the phases of technology advancements, the OCR systems can be broadly categorized into four generations:

1.2.1 First Generation OCR: The first generation OCR systems supported few particular fonts and numerals only that were specially designed characters in machine readable format. In early 60's, companies like IBM initiated research for development of more advanced commercial OCR systems known as the IBM 1418 (Leimer, 1962), designed to read a specific IBM font number 407. The machine used to recognize a character by comparing the character image against already stored template images for each character of all available fonts.

1.2.2 Second Generation OCR: In the middle of 60's, the evolution of second generation OCR's took place that supported more number of fonts, along with recognition support for hand written characters. During this time IBM developed second generation OCR that was popularly known as IBM 1287 (Eikvil, 1993). In 1968, American National Standards Institute (ANSI) designed a new type of fixed-width font OCR-A (Bigelow, 2013). The OCR-A character set was introduced for

standardization, easy readability for humans and optical recognition by computers. The character set of OCR-A is shown in Figure 1.2.

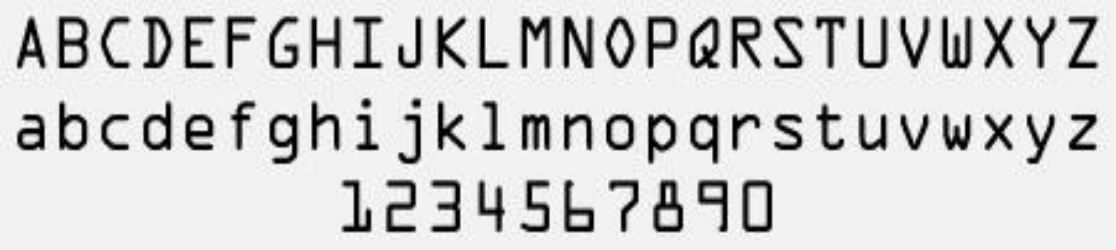


Figure 1.2 OCR-A Digits, unaccented capital and small letters

In continuation to standardization, a European fixed-width font (OCR-B) was developed in 1968 by Adrian Frutiger (Bigelow, 2013). The OCR-B was specially designed for the banking sector (It contains all ASCII symbols and some specially designed symbols for the banks) and gained global acceptance in the year 1973. The character set of OCR-B is shown in Figure 1.3.

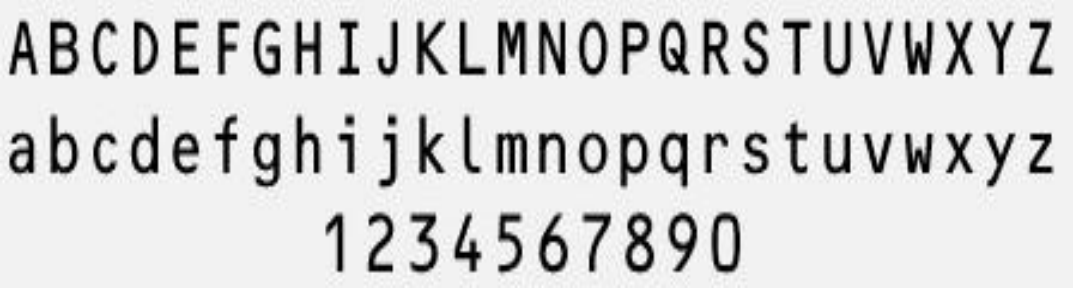


Figure 1.3 OCR-B Digits, unaccented capital and small letters

1.2.3 Third Generation OCR: The third generation OCR research focused upon quality improvements, handwritten text, inclusion and support for new fonts. Most of the people (especially big organizations) understood the importance and usefulness of OCR technology. The actual commercial usage of the technology triggered during this period. Countries like the UK and Canada started commercial use of recognition systems for bill payments in 1965 and postal mail system in 1971. In continuation of OCR technology advancement, during 1974, Ray Kurzweil developed first omni-font

OCR system capable of recognizing printed text in any standard font. During that period, most of the research was carried out on Latin-script, with satisfactory accuracy rate on standard fonts with fixed shape characters.

1.2.4 Fourth Generation OCR: The fourth generation OCR technology focused upon complex documents containing text, images, graphs, colored content, tables, low quality noisy pictures, unconstrained handwritten characters and mathematical symbols being used in today's world. Among the commercial products, postal address readers, and reading aids for the blind were getting available in the market, that were based on OCR's of this generation.

Since then, number of OCR software's have been worked upon by the researchers for the digitization of machine printed (typewritten, faxed, printed etc.) and handwritten (free hand individual writing) text documents. However, the reported recognition accuracy levels still require greater improvements and efficiency. The recognition of characters in many Indian scripts still continues to be an active area of research (Casey et al., 1996; Pal et al., 2004; Bag and Harit, 2013; Jayadevan et al., 2011; Pal et al., 2012; Bag et al., 2013). Thus researchers are showing more interest in accessing digital content, recognition of huge volume data, language modeling and indexing services etc.

1.3 Types of OCR

Depending upon the type of input, the OCR systems can be grouped into two major categories, a) Machine Printed Character Recognition and b) Handwritten Character Recognition as shown in Figure 1.4.

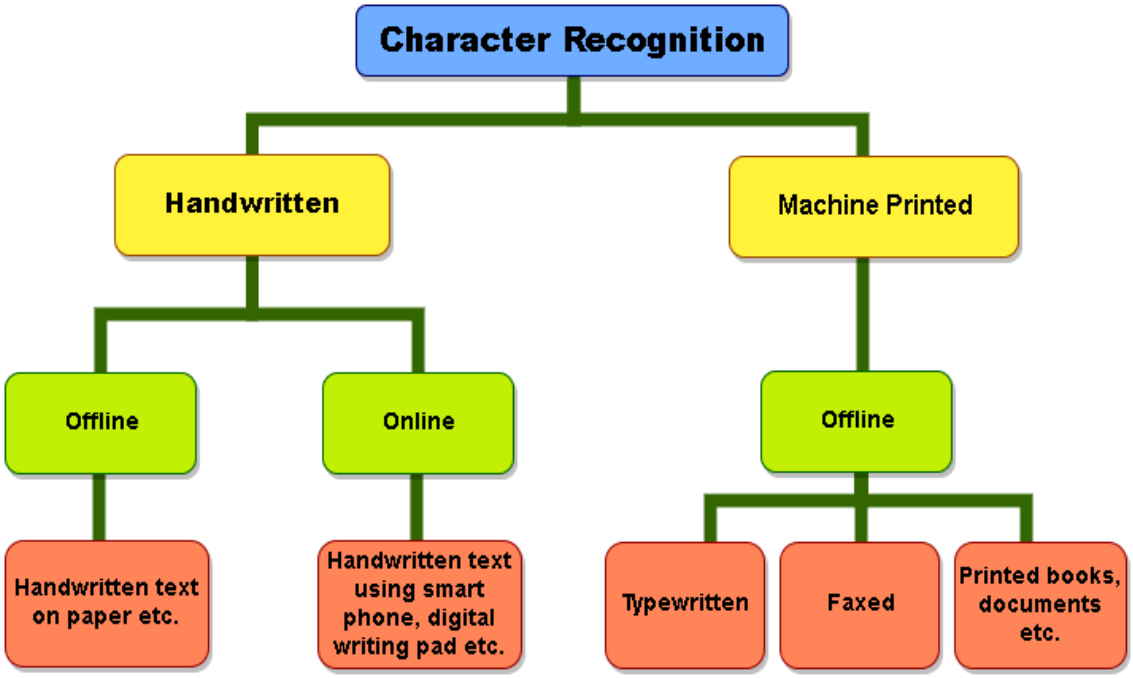


Figure 1.4 Types of OCR systems

If the images of input document are in the form of machine print, then it comes under the category of machine printed character recognition. If the input document images are in handwritten form, then it is known handwritten character recognition. OCR software enables the possibility of identifying characters on the image, further into words and finally words into sentences. Hence, digital access and editing of the content in the actual document is possible.

1.3.1 Machine Printed Character Recognition

The machine printed character recognition works in offline mode only and can recognize text from typewritten, faxed, magazines, newspaper, books and printed documents. The offline system takes input as scanned or camera captured images and convert them into digital text using recognition algorithms as shown in Figure 1.5. Developing an efficient and accurate offline OCR system is complex, time consuming and daunting task due to challenges like image skewness, noise, gray background, touching, broken, heavily printed, improper segmentation and misclassifications.

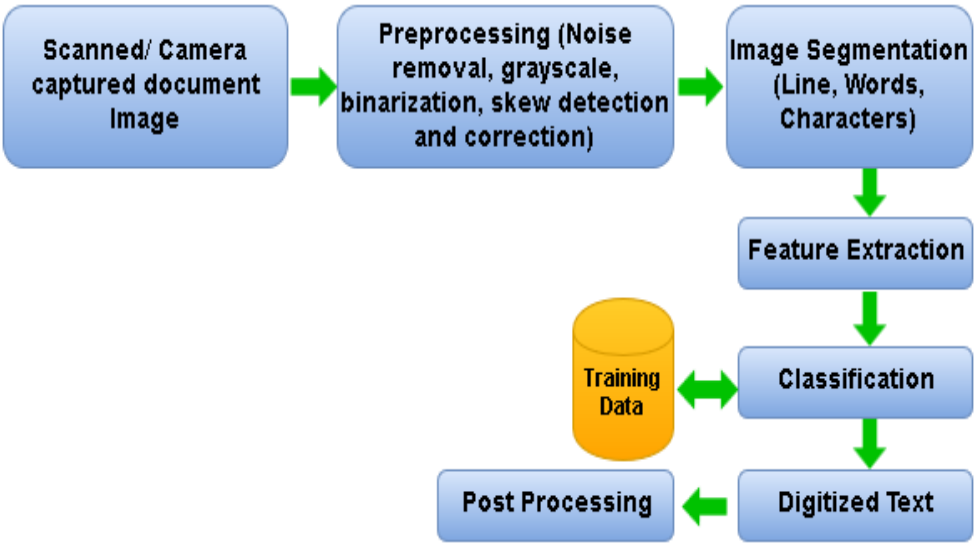


Figure 1.5 Working of offline OCR

1.3.2 Handwritten Character Recognition (HCR)

Before the invention of smart devices, the handwritten character recognition was performed using offline mode only. In offline mode, the scanned image of handwritten text is processed for the recognition of text. The procedure of handwritten character recognition is similar to machine printed character recognition. The online handwritten character recognition came into existence in the early 1980's (Vashishtha et al., 2014) with the invention of new hardware devices that can capture user input as handwritten text with the help of a digital pad and pen as shown in Figure 1.6. Nowadays, almost all the digital devices like smart phones, laptops, desktop LEDs, digital panels etc., are equipped with the touch screen technology. The digital touch screen helps a user to input handwritten text that can be dynamically recognized.

Figure 1.6 Illustration of online HCR

When someone starts writing on a digital touch screen, then the device captures the co-ordinates of the moment of pen (stylus). As soon as the user stops writing (pause time interval) by lifting writing pen, then the device processes the captured data to recognize the user input and accordingly converts user input into digital text.

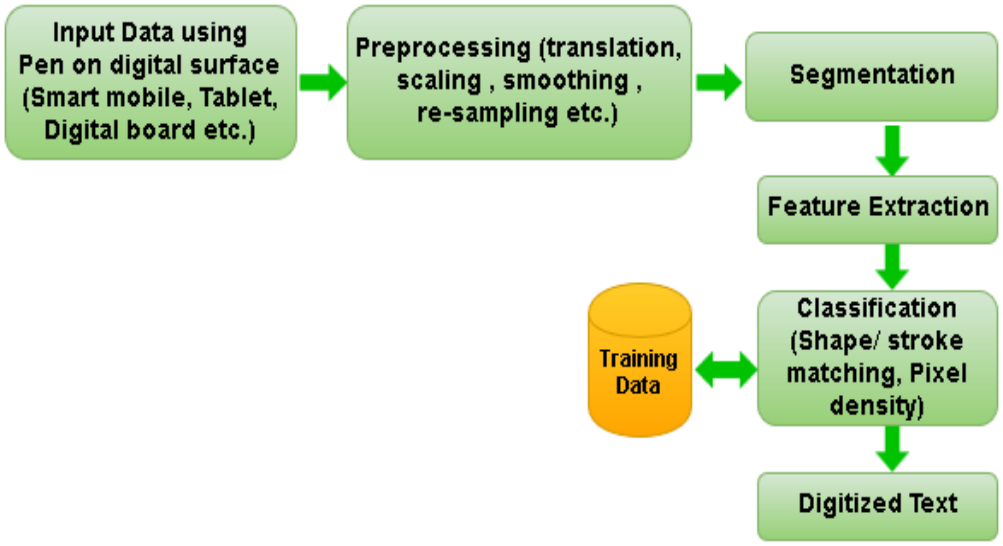


Figure 1.7 Working of online OCR

The online HCR system is less complex and easy to implement in comparison with the offline OCR system. The online system uses one dimensional data only instead of two dimensional images used in offline system. The working of an online character recognition system is shown in Figure 1.7.

1.4 Basic Stages of Machine Printed Character Recognition

The machine printed, optical character recognition system is divided into various stages. The working of all the stages are explained in the following sub-sections.

1.4.1 Conversion of Input Data into Digital Form

Any optical character recognition system requires any input document for digitization. At this stage, the input text document (printed book, magazine, historical document etc.) is scanned using a scanning device and the image file (BMP/ JPG/

PNG etc.) is stored on a computer. One can also make use of camera clicked document image (using a smartphone, tab, camera etc.) as the input data. During scanning, the quality of an image is measured in dots per inch (DPI) and it has great impact on image quality and overall digitization process. For OCR image processing, the effective data extraction optimum level of image resolution is 300 dpi (Shafait et al., 2008).

Generally, at this stage, the manual data are converted to a digital image and process it using a computer. The digitized image file is then forwarded to the pre-processing stage for further processing.

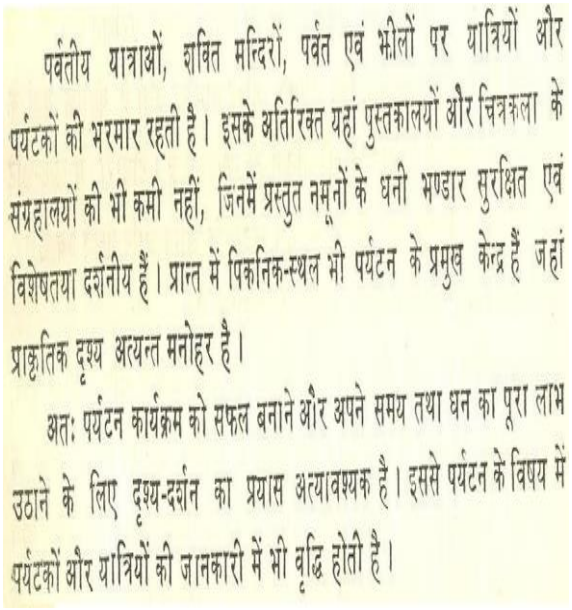
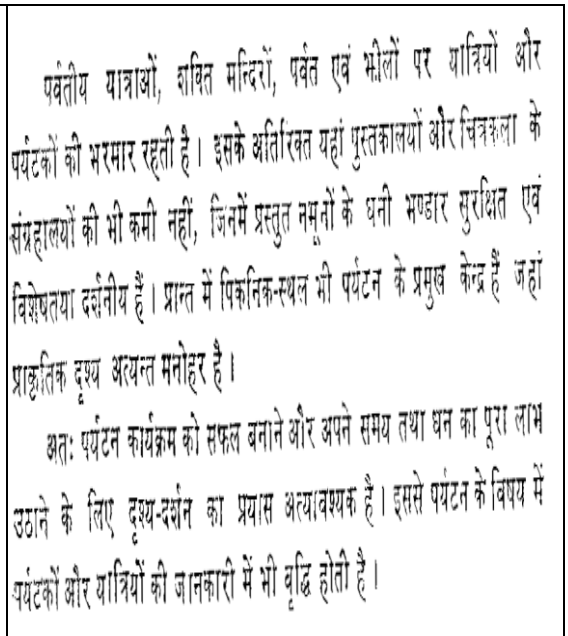
1.4.2 Pre-processing

Pre-processing is a stage that involves image cleaning, enhancement and smoothing. Pre-processing of the scanned document image is required to remove/ minimize various types of ambiguities occurring due to scanning and paper conditions. This stage is used to produce improvised image that can be accurately and efficiently processed by a character recognition system. The digital image pre-processing involves the following steps:

1.4.2.1 Image Thresholding and Binarization

Generally, an OCR system takes digitized grayscale image input. In image pre-processing, the first step is the transformation of the input image document into the grayscale image document. This transformation process is known as Thresholding. The grayscale conversion process removes unwanted pixel intensities (different gray levels). After that the digital image remains with two pixel intensity values (Black and White) only as shown in Table 1.1.

Table 1.1 Input digital image

	
(a) Original image	(b) Gray-scaled image

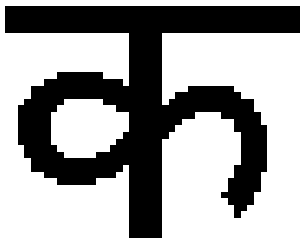
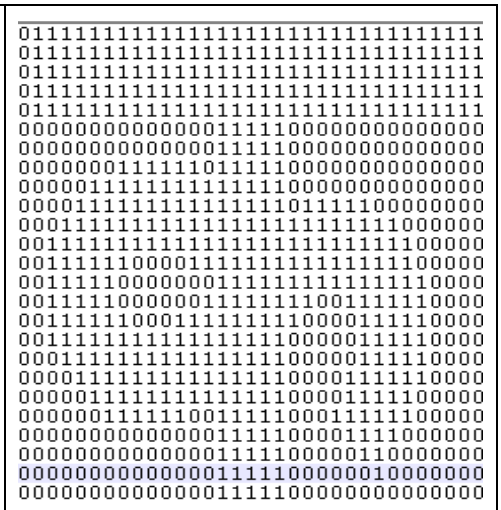
Thresholding technique is used to select the candidate pixels and to ignore the non-required pixels. In order to convert a digital image into grayscale, a threshold constant value is selected. All the pixel intensities with value lesser than the threshold value are converted into black color and the pixel intensities having a value greater than the threshold are converted into a white color.

Selection of threshold value has great impact on binarization process. Selecting a high threshold will result into removing some of the character strokes and low threshold will result into unclear boundary areas of characters. Generally, two popular methods of image thresholding are in use (Sauvola and Pietikainen; 2000; Sezgin and Sankur, 2004; Yan et al., 2005; Shafait et al., 2008; Toh and Isa, 2010; Chandler, 2013; Horng et al., 2014). In the global threshold method, only one threshold constant value is selected for the entire image area. It works by estimating the image background level measured by using the intensity histogram, but this technique is not

suitable for degraded quality images because cases may exist where some portion of the image is very dark or light. Whereas, the local (adaptive) technique divides the whole image into sectors and process each sector individually with sector wise threshold value. In this way, the dark areas of the image are processed separately than the areas having lighter color intensity. The thresholding technique not only improves the digital image, but also reduces the size of digital images. This helps in lesser data processing during the character recognition process.

After thresholding, the grayscale document image is transformed into a binary image. The document image binarization is a challenging task due to different gray levels and noise. A binary image is a digital image in which all pixel intensities are assigned the value 1 (black) or 0 (white) as shown in Table 1.2. This process is termed as binarization. Image binarization helps in applying mathematical operations like image scaling, transformations and feature extraction etc.

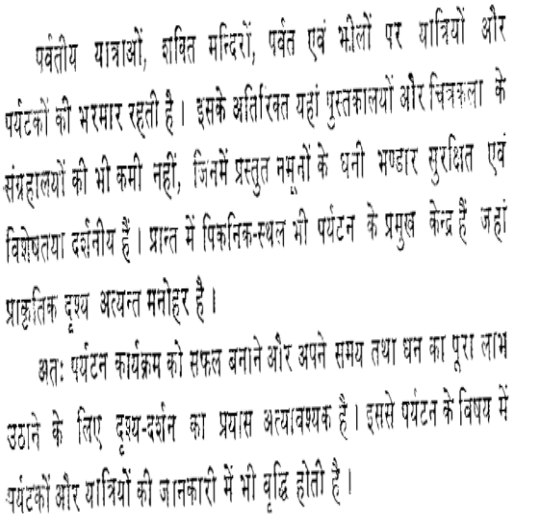
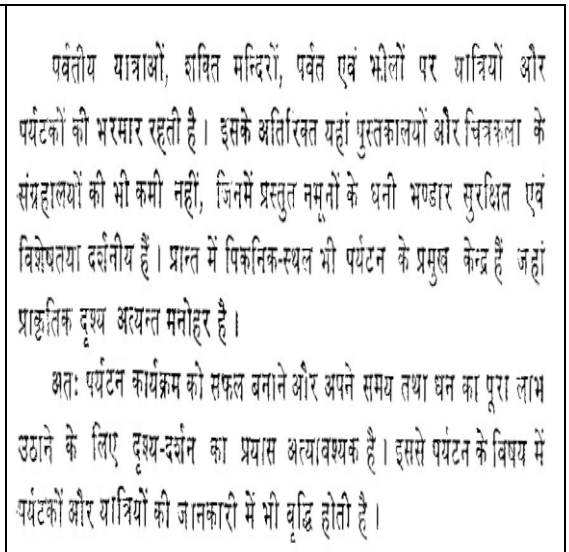
Table 1.2 Image binarization

	
(a) Original image	(b) Binary image

1.4.2.2 Skew Detection and Correction

Skew correction techniques are used to align the text in document image to improve recognition results. When a document is scanned using a scanner, then accidently it can be improperly fed by user or rolled by the scanner. This leads to some tilt or slope in the scanned image as shown in Table 1.3(a). This tilt or slope in the image is termed as skewness. The skewness can also occur in the document image clicked using a camera. If a character recognition system proceeds with the skewed image, then the overall performance/ result will be unsatisfactory. It is very important that the document image should be straight/ aligned horizontally and vertically before further processing. It helps in obtaining higher results by minimizing the errors caused by skewed text. Some of the techniques for identifying and estimating the skew angle are Hough transform, projection profiles, cross-correlation, and nearest-neighbor (Farrow et al., 1994; Chaudhuri and Pal, 1997; Kapoor et al., 2004; Shafait et al., 2008; Kavallieratou et al., 2002; Saragiotis and Papamarkos, 2008). The de-skewed image is shown in Table 1.3 (b).

Table 1.3 Skew correction

	
(a) Before	(b) After

1.4.2.3 Noise Reduction

The process of noise reduction is carried out to improve the quality of the scanned document. During this process different type of unwanted noise types like shadow, dark areas, salt and pepper noise etc. are removed/ minimized (Singh et al., 1995; Likforman-Sulem et al., 2011; Kim et al., 2011; Chandler, 2013). The regions of the binary image are checked to distinguish printed text from the background color. The overall process reduces the data size and helps in properly extracting the shape information of the text. There are different techniques available for image noise reduction. One such commonly used technique is of low pass filter which utilizes the vector point's data representing the text and then make use of 3x3 windows for traversing the text image pixel by pixel. The existence of each pixel is analyzed, based on the value of its 8-neighboring pixels (Moazzam et al., 2016; Irum, 2014).

1.4.3 Segmentation

Segmentation technique divides a digital image into sub regions based upon a set of rules, i.e., document images containing paragraphs, lines and words are segmented into individual characters (Casey et al.,1996; Garain et al., 2002; Kompalli et al., 2009; Nikolaou et al., 2010; Grafmller, 2013; Bag, 2013) as shown in Figure 1.8.

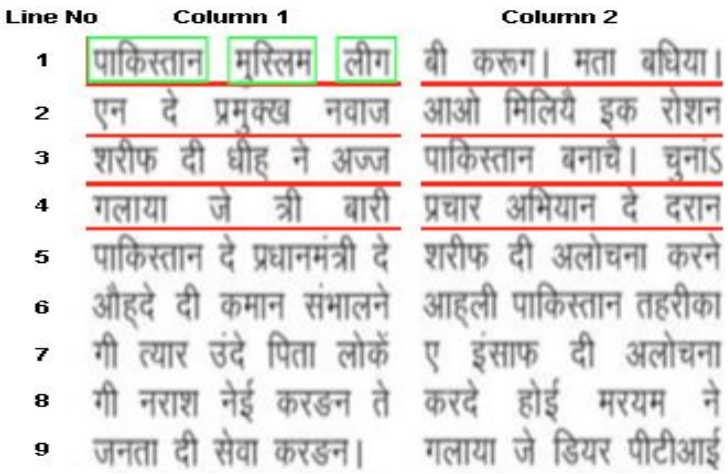


Figure 1.8 Candidate lines (red color) and words (green color) for segmentation

The accuracy of character segmentation has significant impact on character recognition results. Generally, one segmentation error causes at least single recognition error (Bansal, 2002). There are three major strategies for segmentation:

1.4.3.1 Classical Method: In this method an image is partitioned into meaningful components having a character like properties.

1.4.3.2 Recognition Based Segmentation: It analyses the image for the attributes that match to the properties of the alphabet.

1.4.3.3 Holistic Method: This method does not segment the word into characters. It only tries to recognize the complete word.

The combinations of two or more of the above strategies are known as hybrid approach (Bansal, 1999).

1.4.4 Feature Extraction

It is the technique of extracting relevant pixels in a binary image that have some distinctive characteristics. The extracted feature data set represents distinct pixel information for unique identification of objects. The main goal is to maximize the recognition rate with the least amount of data. This stage reduces the amount of data by extracting relevant information from the input data. Feature extraction can be divided into two categories global and structural features (Lehal et al., 1999; Guyon et al., 2003; Srinivas et al., 2008; Siddiqi et al., 2010; Singh et al., 2011)

1.4.4.1 Statistical Features

These are basic features of a character and can be identified with a very less effort. In this category features are extracted based upon the statistically distributed data points. These types of features have resistance towards variation in shape, size and style. Some of the statistical techniques are:

- **Zoning:** In this technique, the whole character matrix (rectangular boundary area of the character) is divided into small regions (zones). The strength of black pixels in each region (zone) is calculated, which is used as feature vectors.
- **Moments:** It is the statistical measure of the pixel density towards the center of gravity of the character. A character can be identified by measuring the distance between pixels of character matrix and the centroid of the character. This concept is known as central moments.
- **n-tuples:** It is a technique in which features are selected on the basis of the position of black or white pixels in a character image.
- **Projection Histogram:** A character is inspected for the number of black pixels in the particular area in both directions i.e., vertically and horizontally. The projection histograms can be in any direction i.e., left or right diagonal, vertical or horizontal.
- **Crossings and distances:** In this technique features are selected by inspecting character shape crossing count by vectors towards particular directions and in case of distance technique, some lengths towards the vectors junctions of the character shape is measured.

1.4.4.2 Structural Features

In this category, the features that describe the topological and geometric structure of the character shape are extracted. It deals with the curvature portions (concave and convex shape area), total quantity of character endpoints and holes, joints, intersection between the lines, loops and bounding box of the character. Features identified with this technique are highly isolated from noise and different styles. The main drawback of this technique is of more overheads.

1.4.5 Classification

It is the technique of character identification and its grouping with correct character class. It decides about the class membership of some pattern by comparison of the feature vectors of the candidate character. The classification stage makes comparative analyses of the extracted features in order to identify the text segment based upon certain rules. There are a number of techniques which are followed by researchers for classification purpose. Some of the commonly used techniques are statistical methods, template matching, artificial neural networks, binary classifier trees and kernel methods (Mill and Inoue 2004; Kompalli et al., 2009; Wang et al., 2010; Singh et al., 2011; Fu et al., 2011; Jayadevan et al., 2011; Ramteke et al., 2013; Kale et al., 2013; Singla et al., 2014; Do et al., 2016).

1.4.5.1 Statistical Methods

It determines the category of the given pattern in which it belongs. These methods belong to an auto trainable category. A measurement vector is prepared by preparing a set of numbers of observations and measurements. One of the methods is k-NN which makes comparison of an unknown symbol against the collection of predefined symbols which are already marked with unique class identifies in the training stage (Bansal, 1999; Pal et al., 2012; Aharrane et al., 2015). On the basis of comparison, a pattern is said to be identified if it matches with the closest distance of predefined pattern. Another commonly used statistical method is the Bayesian classification method in a pattern, which is matched against unknown samples and assigns them to a class which has the maximum similarity.

1.4.5.2 Structural Methods

These methods are commonly used for classification of handwritten texts in which input patterns are classified by analyzing the components of a character and the relation between these components. The pattern characteristics can be derived from actual shape of the character that are difficult to quantify and the class membership is decided by the relationship between the characteristics (Kumar et al., 2014).

1.4.5.3 Template Matching

In this technique, a prototype or carbon copy of the pattern that is to be compared with the desired one is available in the database of patterns. The sample is superimposed on a predefined template pattern and is assigned to a class on successful match. During comparison, size and style of the pattern is not considered, therefore, it is considered as one of the easier techniques to implement (Gaikwad et al., 2013).

1.4.5.4 Artificial Neural Networks

Basically, it is the composition of closely connected elements which are known as neurons. The training of a neural network is automatically self-regulated by a set of examples and from applications than can learn bigger databases. Commonly used neural networks are feed-forward network, multilayer perception, learning vector quantization (LVQ), radial-basis function (RBF) networks, convolutional neural network and vector quantization (VQ) networks (Singh et al., 1990; Mani et al., 1997; Frias-Martinez et al., 2006).

1.4.5.5 Kernel Methods

The most popular kernel methods are support vector machines (SVM), kernel fisher discriminant analysis and kernel principal component analysis. SVM is a combination of supervised learning algorithms that are applied to the feature set for

pattern classification. There are various kernel algorithms available for SVM like polynomial kernel, linear kernel etc. The SVM initially needs to be trained on some part of the data to be classified and then testing is done on the remaining data set. Based upon the training set SVM predicts the target values of the test data (Frias-Martinez et al., 2006; Al-Boeridi et al., 2015).

1.4.6 Post-processing

It is the process of correcting left out errors in the recognition stage resulting due to contextual information (Perez-Cortes et al., 2000; Lehal et al., 2002; Hu et al., 2011). The string produced by classification process can be refined further using the post processing technique. It aims at making improvement in the input received after classification stage. There are possibilities of error generation during segmentation and classification stages. To correct those errors, the dictionary based approach is used. Semantic relations between words can be used to select the most appropriate word from probable words in order to improve the overall recognition rate. Context may be operative at the word, sentence and semantics level. The final output of the system is Unicode based characters obtained after the mapping. The dictionary look-up method is used during post-processing and it operates at the word level. The example shown in Figure 1.9 presents a clear picture of the post processing technique.

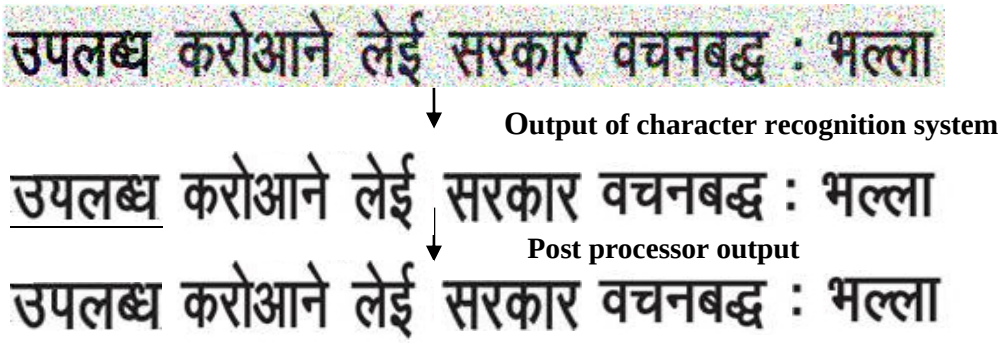


Figure 1.9 OCR result improvement using the post processing technique

1.5 Need for Research

With the advancement in technology, it is possible to digitize and preserve the information available in the form of historical documents and books. Such type of digitization systems with satisfactory accuracy levels are available for Roman, Chinese, Arabic and Japanese scripts. However, papers reporting digitization work for Indian scripts (like Gurumukhi, Bangla, Devanagari) are limited with reported recognition systems working on good quality text images only (Chaudhuri and Pal, 1997a; Chaudhuri and Pal, 1997b; Bansal, 1999; Lehal, 2001; Kompalli et al., 2009).

The work done by researchers specifically on Devanagari script gives average results on good quality documents, some specific fonts, languages and document conditions. It has been analyzed that the presently available recognition system for Devanagari script are not robust and specifically cannot fully handle scanned documents of Dogri language documents. In addition, it was noted during the study that no software is available that can exclusively digitize machine printed image documents of Dogri language. In light of the identified limitations, the need for a character recognition system for digitization of Dogri language text was identified, to enable larger user base to get access to Dogri literature and historical documents.

1.6 About Dogri Language

The Dogri language is part of an Indo-European group of languages and it is a member of the Western Pahari language group (Das, 2010; Dubey et al., 2011). It is used in the regions of North India like Jammu and Kashmir, boarder areas of Himachal Pradesh and Punjab. It is also written and spoken in some parts of Pakistan. Dogri is an important language having millions of people speaking this dialect in the border areas of India and Pakistan. It has been taught since 1980 at University level

primarily in Jammu area and at the primary school level. It is being taught as a second language at Northern Regional Language Centre, Patiala, Punjab since 2006 (Das, 2010). The people who speak Dogri are known as Dogras and the region to which they belong is termed as Duggar. In December 2003, Indian government gave official status to Dogri language and was included (Ninety-second Amendment) in the 8th schedule of the constitution of India (The Constitution Act, 2003).

After its origin, Dogri language was written using the Takri script and it gradually included a number of words from Arabic, Persian and English. During the 20th century, Dogri literature flourished in diverse spheres of poetry, novels, short stories and plays. Dogri is written using the same script as Hindi i.e., Devanagari script, but its character set contains thirty eight segmental, ten English digits and five supra segmental phonemes written from left to right. Segmental phonemes are grouped into two broad categories i.e., ten vowel and twenty eight consonant phonemes. In addition to these, Dogri contains vowel modifiers known as a Matra symbol (which can be positioned to the left, right, above, or at the bottom of a character or conjunct), pure-consonant (also called half-letters), which when joined with other consonants produces conjuncts. The basic set of consonants, vowels and numerals are illustrated in Table 1.4 (Das 2010; Dubey et al., 2011; Dubey et al., 2015; Kour et al., 2016).

Table 1.4 Dogri language (a) Consonants, (b) Vowels, (c) Vowel sign, (d) Numerals and symbols

(a)	(b)	(c)	(d)
क ख ग घ ङ च छ ज झ ञ ट ठ ड ढ ण त थ द ध न प फ ब भ म य र ल व श स ह ड़ ढ़	अ आ ऊ ए इ ई ओ औ ऐ उ	ा, ि, ी, ु, ू, ृ, ॄ, ॢ, ॣ, े, ै, ो, ौ, ँ, ं, ः	**1 2 3 4 5** **6 7 8 9 0** ऽ (extra-long vowel)

In addition to the characters shown in Table 1.4, Dogri language contains many compound characters which are composed by combining two or more characters or half characters. Almost, all the characters have a horizontal line on the top called Shirorekha (also known as a header line). The language has its own dictionary and grammar (Das, 2010; Dubey et al., 2011). Figure 1.10 shows a sample image document of Dogri language (Jammu Prabhat Newspaper).

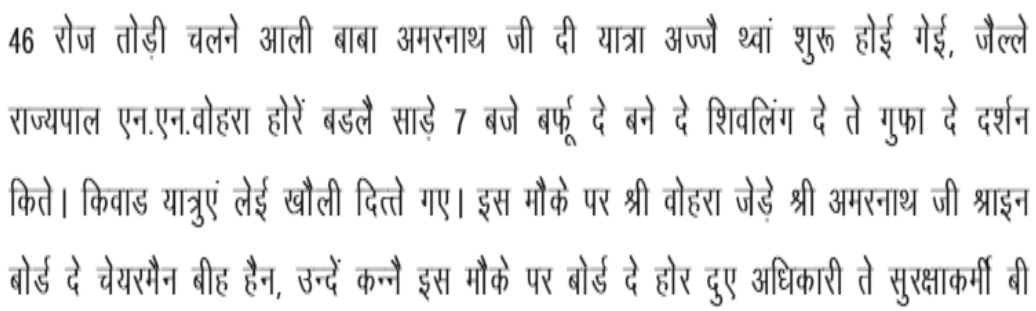

46 रोज तोड़ी चलने आली बाबा अमरनाथ जी दी यात्रा अज्जै थ्वां शुरू होई गेई, जैल्ले राज्यपाल एन.एन.वोहरा होरें बडलै साड़े 7 बजे बर्फ़ू दे बने दे शिवलिंग दे ते गुफा दे दर्शन किते। किवाड यात्रुएं लेई खौली दित्ते गए। इस मौके पर श्री वोहरा जेड़े श्री अमरनाथ जी श्राइन बोर्ड दे चेयरमैन बीह हैन, उन्दें कन्नै इस मौके पर बोर्ड दे होर दुए अधिकारी ते सुरक्षाकर्मी बी

Figure 1.10 Sample image text from Dogri language newspaper

Dogri has a unique set of characters/ symbols, which are not used in other languages written using Devanagari Script. Figure 1.5 shows exclusive Dogri words.

Table 1.5 Commonly used character and word formations of the Dogri Language

S.No.	Character	Words (Meaning)		
1.	ञ	संञ (Dusk)	ञ्याना (Child)	सोआञना (Tree)
2.	ङ	चैङन (Tomato)	ङूर (Grapes)	ङूठ (Thumb)
3.	"ऽ"	लोऽ (Light)	भ्राऽ (Brother)	ग्रांऽ (Village)
4.	'	रव'ल्ल (Leather)	ग'ल्ल (Cheek)	फ'ड़ (Exaggerate)

A large volume of Dogri language data is present in the form of machine printed documents, books and manuscripts. If someone needs to search text out of image documents, then one has to search it manually by reading each page till the required

text is obtained. Moreover, accessing of data in such a way is limited to few users. These document images can therefore be digitized with the use of an imaging device. System is required to be developed to perform reading, searching and other file operations on image document. Currently, number of commercial OCR software in some popular languages are available but not much work is done so far on OCR for regional languages like Dogri, due to certain inherent limitations and challenges.

1.7 Challenges in Dogri Character Recognition

There are a number of factors which make the implementation of Dogri OCR a complex and challenging task. The main factors which play important role in Dogri character recognition are:

a) A character has different shapes, style, size and non-standard font set. Image variations of Dogri character क are shown in Figure 1.11.

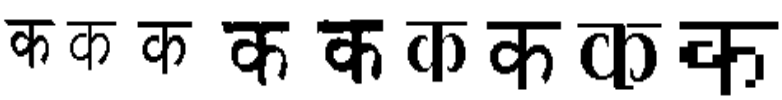
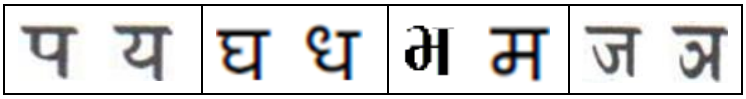


Figure 1.11 Dogri language character images in different font, style and size

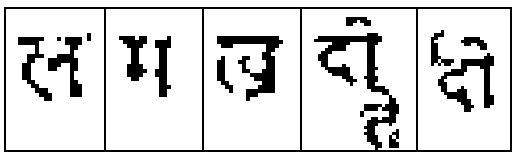
b) Similar looking characters increase the complexity as shown in Table 1.6.

Table 1.6 Dogri language similar looking character images



c) Complexities and problems involved in touching, overlapped and broken characters as illustrated in Table 1.7.

Table 1.7 Dogri language broken, overlapped and touching character images



d) Segmentation and classification of conjunct/ half consonants/ diacritic characters. Few of such compound character images are shown in Figure 1.12. There are more than seven hundred compound character formations.

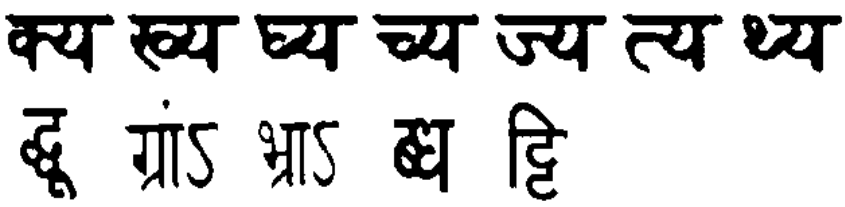


Figure 1.12 Dogri language compound character images

e) For an example, a paragraph containing touching, broken and overlapped lines, words and characters is shown in Figure 1.13.

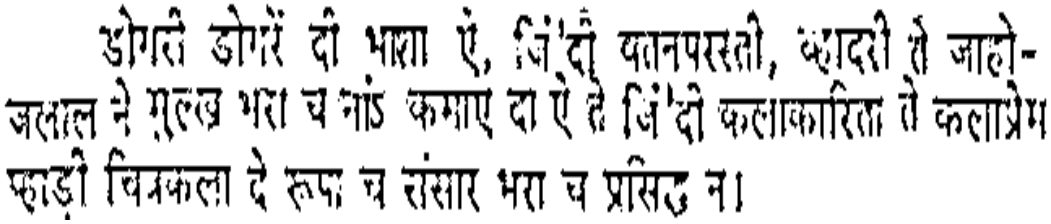


Figure 1.13 Dogri language text image with broken, overlapped and touching characters

f) Unavailability of benchmarking database.

The objective of this research was, therefore, to design and develop new algorithms for the recognition of printed Dogri language documents. In addition, there was a need to develop software that can convert machine printed document images written in Dogri language into a digitally editable form. During this process, special focus was required to recognize conjunct/ half consonants/ diacritic characters, as these increase the error count during the segmentation stage. Algorithm(s) were required to deal with this category of characters to obtain higher recognition accuracy. Also, post processing techniques like dictionary lookup and statistical methods were used to be explored and deployed for accuracy improvement.

1.8 Assumptions

The following assumptions were made during the development of algorithms for Dogri language character recognition system:

a) Pre-processed documents are considered for the experimental work i.e., noise cleaning, background gray color removal, slant and skew detection and corrections have been made using existing techniques.

b) Printed documents with single column text material were considered i.e., documents with non-text matter was filtered out like pictures, graphs, tables etc.

1.9 Objectives of the Research Work

The objectives of the work carried out were as follows:

a) To study various existing techniques of machine printed Optical Character Recognition;

b) To analyze different algorithms used in the field of Optical Character Recognition;

c) To design and develop recognition algorithm for machine printed Dogri Language Optical Character Recognition; and

d) To verify and validate the developed algorithm. Finally, comparison of the developed algorithm was carried out with the existing algorithms.

This thesis details a complete character recognition system for the recognition of machine printed Dogri language documents. During the study of existing segmentation algorithm(s), it is explored that the existing algorithms were not suitable and less accurate for the segmentation of Dogri language characters. Therefore, new algorithm(s) were designed and developed for the segmentation of Dogri language characters, laying exclusive emphasis to the character shape and ensuring that

segmentation, character shape is not altered i.e., whole of the character is segmented without any structure loss.

1.10 Layout of the Study

The layout of this thesis is as per following details:

a) The literature survey of all the stages of character recognition system has been done in detail. Almost, all the Indian scripts, English, Arabic and Urdu scripts were covered in the survey. Various types of issues associated with preprocessing, segmentation, feature extraction and classification have been examined and analyzed.

b) A dataset of around two lakh Dogri language characters has been prepared from old books, magazines, newspaper and other documents. For the experimental work, two different datasets (Dataset-I and Dataset-II) have been prepared. The first dataset contained characters without header lines, and the second dataset contained the same characters with header lines. The recognition results obtained from these datasets were compared to the overall recognition accuracy of the characters segmented using the proposed shape-based algorithm with existing techniques.

c) New segmentation algorithms have been proposed for the separation of lines, words and characters of machine printed Dogri language documents.

d) The proposed algorithms were also tested on Hindi language documents that showed higher recognition accuracy.

e) Shape oriented features were selected using different feature extraction techniques like Discrete Cosine Transformation (DCT), Gradient and Zernike Moments.

f) For the classification of characters using extracted feature file, different classifiers such as Multilayer perceptron neural networks (MPNN), Support Vector Machines (SVM) and *k*-Nearest Neighbor (*k*-NN) were used.

g) Finally, dictionary based post processing technique has been applied for the correction of errors left by the character recognition engine.

The objectives in this research work were achieved by a comprehensive study of various existing techniques used in the development of character recognition systems for various Indian and other scripts. This work is an attempt for the design and development of a character recognition system for Dogri language that can be extended to other Indian languages in which characters have header line.

Chapter 2

Literature Survey

This chapter describes the work carried out by the various researchers in the field of OCR of various languages.

2.1 Pre-processing Techniques

Trier and Taxt (1995) compared different local thresholding (adaptive) techniques used for the binarization of gray scale images. They proved that Niblack's method outperforms and was among the fastest binarization methods. They also concluded that high resolution scanned images gives better binarization results. On the other hand, low quality documents cause poor results despite of using the best binarization method.

Fast and Allen (1997) invented a technique of image enhancement by identifying and cleaning of scanned images containing text. The image enhancement process includes skew detection and removal, connected lines, reverse printing, background noise, different gray level printing and the image processing was accomplished using run-length coding. Their technique was granted a US patent in the year of 1997.

Sauvola and PietikaKinen (2000) worked upon an adaptive document image binarization technique which deals with the challenges caused by noise (salt and paper etc.), various gray levels and different types of degradations. Their algorithm ascertains the local threshold (adaptive) value for each of the pixels. They tested the system with

images which include various kinds of document types with different degradation levels. Although the results obtained using proposed algorithm was satisfactory, but the algorithm takes more computation time in comparison to existing techniques.

Egmont-Petersen et al., (2002) presented a review of image processing applications which are based upon the method of neural networks. The authors discussed the advancements and future possibilities for the neural networks in the preprocessing stage of the character recognition system. They concluded that the neural networks can play an important role in the image processing as image enhancement, non-linear regression modules, non-supervised extraction of features or in the classification stage.

Yan et al., (2005) proposed a local thresholding technique with the help of variable neighborhood processing. Authors, also proposed a systematic technique for the computation of the relevant coefficient of the mean and variance, which was based upon the global data of the image. Their experimentation setup gave higher accuracy segmentation results on scanned document images and oil sand images in comparison with the existing thresholding techniques.

Bieniecki et al., (2007) worked upon preprocessing techniques to improve the character recognition results on the digital images captured using a camera. Authors experimented using FineReader 7.0 application software as the classification utility for back-end. They proposed some techniques for character recognition systems.

Saragiotis and Papamarkos (2008) worked upon a method of text skew detection and correction. They estimated a local skew angle for each text area and there is an independent skew correction in vertical/ horizontal directions. Their algorithm gives

higher accuracy on scanned documents which are clean and having properly separated text. The results are not good on scanned documents having degraded text, complex and unclear backgrounds. The method also depends heavily upon binarization technique.

Shafait et al., (2008) proposed a new method to compute the threshold value for faster local binarization techniques. The authors make use of the integral images for the computation of mean and variance value in time independent local windows. The results obtained by proposed method are similar to the threshold function of Sauvola but the time taken was equivalent to that of global binarization techniques like Otsu. The method can be integrated with other techniques that make use of local mean and variance like Niblack's binarization technique

Alginahi (2010) presented review of several preprocessing techniques which are widely used in the area of character recognition. Author pointed out that despite of existing preprocessing techniques, still there is a need for the development of new preprocessing methods because most of the existing techniques are application specific.

Pande and Dhami (2010) analyzed the occurrence frequencies of the letters of Hindi language character set in a text document including their occurrence at the word's initial position using Zipf's law (Zipf, 2016). Authors validated the models by comparing the observed and theoretical frequencies for different texts collected from various sources. The occurrence frequency of various letters of Hindi language in a typical page is analyzed and calculated.

Al-Khaffaf et al., (2012) explored the advancement of reconstruction state of Decapod's English font. To determine the changes of glyph shape, the analysis of Potrace

technique and its attributes are done. The authors showed and compare the visual results of Decapod's font reconstruction with the Adobe clearscan. The details of reconstructing the font using both of the methods are presented. The results showed the ability of font reconstruction of both methods for some synthetic book typeset each time with three serif and three sans-serif fonts. The Decapod successfully created a reusable TTF font for both typefaces, which is integrated into the PDF document.

Alihodzic et al., (2014) worked upon improvement in the standard bat algorithm, which is part of the recent swarm intelligence technique and is used in the multilevel image thresholding. The modifications include a few elements of the artificial bee colony algorithm and technique of differential evolution. The modified bat algorithm showed better results in comparison with existing algorithms.

Li et al. (2014) presented a new technique of impulse noise filtering for gray scale images using neuro-fuzzy network. Their proposed filter operates in two modes training and testing. The experimental results showed that proposed impulse noise filter outperforms other conventional filters.

Mesquita et al., (2014) proposed a new technique for image cleaning and binarization to deal with various types of degradations. In order to elaborate the text area, the absolute difference of document image from the background is calculated and used to de-noise the image. The integrated technique of k-means clustering and the thresholding algorithm of Otsu's is deployed for image binarization. Two different datasets with scanned images of printed and handwritten documents are used. The proposed technique

gave satisfactory results when experimented with said datasets. The authors conclude that there is need to improvise the grouping scheme of Otsu's and k-means algorithms.

Wang et al., (2014) developed a new data mining technique for the diagnosis of noise type and designed a novel fuzzy filter design for the quality improvement of noisy images. The traditional filtering methods for image restoration are not much effective in cases where the noise type information is unavailable and images having mixed noise. The proposed technique fails in the cases where some pixel position values becomes zero.

Tu et al., (2015) introduced a two-stage technique for de-noising an image which is inspired by the theory of statistical learning. The authors presented a framework in which the variety of noise and category are initially judged using support vector machine (SVM), and after that the same data is deployed in the proposed de-noising technique for performance improvement i.e., image de-noising. The effectiveness of the technique has been shown using comparative analysis with the existing techniques.

2.2 Segmentation Techniques

Lu (1995) presented an overview for the segmentation of machine printed image documents. They analyzed that the broken, overlapped, touching and bleeding characters were mainly contributing in the majority of errors during the development of a character recognition system. Various methods for the segmentation of different fonts, degraded and broken characters were also discussed on the basis of features and recognition results.

Casey and Lecolinet (1996) presented survey on the segmentation techniques used in the field of character recognition. The main goal of the review is to explore and discuss

the advancements of the developed techniques. The authors categorized the segmentation techniques into four groups. a) classical technique in which an input image is divided into sub-images for the purpose of classification. b) the image is explicitly segmented and classified by pre-specified windows, or the image is implicitly segmented using the classification of sub-sets of spatial features, gathered from the full image. c) hybrid technique made from first two. d) finally, the holistic technique tries to classify the whole image of the character as a single unit instead of segmenting the image. The authors concluded that more extensive experimentation is required on larger datasets.

Amin (1998) presented a comprehensive analysis of the machine printed image documents, including handwritten images in Arabic language. The main motive of the study was to exclusively discuss the advancements of the Arabic text classification system. It was concluded that the segmentation phase needs to be explored and algorithms need to be tested on larger databases. For optimized and accurate results, the authors suggested to devise advanced methods like fuzzy logic and neural network.

Grain and Chaudhuri (2002) presented a method which works on the principle of fuzzy multifactorial analysis for the detection and separation of connected characters in printed document images of Devanagari and Bangla scripts. The authors pointed out that most of the recognition errors occur due to inefficient connected character segmentation process. The experimental results showed satisfactory improvement in the classification and it can be further improvised with the increase of calculations. Their experimental setup achieved an accuracy of around 79.64% on Devanagari and 78.99% on Bangla documents. The method can be replicated on other character recognition systems.

Pal and Chaudhuri (2002) introduced a new technique based on fuzzy multifactorial analysis for the detection and separation of the connected characters in the machine printed Devanagari and Bangla scripts. The authors developed a technique for the selection of possible cut regions for separation of connected characters. The experimental results showed significant improvement in the segmentation process with an accuracy of around 79.64 for Devanagari and 78.99% for Bangla script respectively. The proposed approach works only with discussed scripts and could also be applied to other regional languages with suitable modifications.

Pal et al., (2003) worked upon recognition-free approach for segmentation of touching numerals using the water reservoir concept. The technique was used for unconstrained handwritten image documents to get features. The best eligible cutting point was obtained by checking of the reservoir outer area, connected location and unique features of the connected characters. Finally, morphological structural features were combined to generate the separation path for segmentation. The drawback of the presented scheme was that it handles only two digit touching components. There are still possibilities for the development of a generic system that could be handled connected patterns of more than two numerals.

Louloudis et al., (2008) worked upon an algorithm for the identification of lines that contains text in handwritten scanned documents. The algorithm was implemented in three steps a) Image cleaning and Binarization, connected component detection and extraction, separation of the touching component area into three parts and the approximation of a character average height, b) detection of probable text lines was done

using a block-based Hough transform; and c) Then the correction of probable cut position, identification of the lines that contain text and are left un-segmented during previous step. The efficiency of the technique was evaluated on a consistent and particular evaluation methodology. The technique performed better than existing methods for the search of possible text lines for specific handwritten documents only.

Kompalli et al., (2009) proposed a novel classification oriented segmentation technique for Devanagari script character recognition system with the help of hypothesize and test model. The authors presented a stochastic model for the classification of words, which combines classifier points, language composition rules, and the characters n-gram statistical information. After that the services of post-processing techniques like word n-grams/ grammar models are deployed to improvise the classification output. The Hindi language dataset has used for implementation of the language model. The corpus dataset from other languages which are sharing the Devanagari script needs to be explored.

Saeed and Albakoor (2009) proposed a segmentation algorithm which was based upon region growing method for the machine printed and handwritten text documents. The authors considered three categories of Arabic language letter fonts for typewritten texts; Simplified, Arabic Transparent and Arabic Arial. Their experimentation work results in the recognition accuracy rate of around 93% for Arabic machine printed text. To achieve high recognition rate in the case of handwritten text, existing preprocessing techniques like the correction of tilt and slant of the text rows were deployed. They also suggested that the work on hand written documents containing multiscript texts with larger dataset size could be extended using neural networks.

Liu et al., (2010) worked upon a semi-supervised learning framework for the segmentation of Manhattan-layout scanned documents with high noise levels. The method was implemented in three steps: a) seed filling algorithm was used during the initial segmentation phase; b) loop based grouping technique which utilizes the details of projection profiles for estimating the vertical border of the document contents; and c) an interior document-content noise removal which make use of the online training and classification. The algorithm was tested on two distinct datasets and satisfactory results were obtained for document and text-line level segmentation. The work could be enhanced using the algorithm of layout estimation for processing more complex documents like newspapers without manhattan layouts.

Nikolaou et al., (2010) worked upon a novel technique for the segmentation of text lines, word and characters from the low quality machine printed scanned documents. The proposed algorithm successfully segments most of the cases in which text is of varying size, document with graphic-text areas and skewed, connected and overlapped lines. The robustness and effectiveness of the proposed technique were illustrated on the low quality machine-printed documents.

Sharma and Dhiman (2010a) analyzed the problems incurred in the document image segmentation in case of handwritten documents of Gurmukhi script. They observed that the segmentation of typewritten image text was less complex than that of handwritten image text. The problem becomes more challenging when handwritten texts are written in 2D scripts and character that occur in a word will have similar pixel values

in the horizontal direction. They suggested that a hybrid approach could be developed with classical and recognition based approach for higher segmentation accuracy.

Sharma and Dhiman (2010b) discussed various challenges in the segmentation of handwritten Gurmukhi script image documents. The authors analyzed that the unique structural properties of the characters play a vital role in decision making of the segmentation stage. In addition to that compound characters and writing style also affect the segmentation process. To handle these kind of problems there is need to design a hybrid technique using classical and recognition based methods, so as to obtain more accurate segmentation results.

Jayadevan et al., (2011) presented a study of different techniques of offline character recognition of Devanagari script. A number of techniques related to feature extraction and recognition were compared. Also, for character recognition, classifier combinations were explored. They analyzed that most of the recognition errors are due to incorrect character segmentation. It was also observed that exclusive keyboards should be used for Devanagari text, because the character set in Devanagari script is larger than in Latin script. Also, more concentration is required to be paid on correct segmentation of lines into words to characters to handle and decrease the recognition errors.

Kumar and Singh (2011) presented a new algorithm for the segmentation of handwritten image documents into individual characters of Gurmukhi script using reverse engineering approach. The algorithm was applied to various handwritten documents and it showed satisfactory results. There are few cases in which the proposed algorithm fails to segment the images because of structural similarities of the characters.

Chen et al., (2012) proposed a new knowledge based system with a set of two new rules; a) extraction of the text region and b) identification of the text-line. Their system initially rebuilds the scanned document into distinct object planes. Further, these planes were separated into homogeneous objects, distinction of text area from non-text patterns like pictures and graphs etc. Finally, separation of background illuminations, compound text images of various colors, gray levels, font sizes and styles had been performed. The proposed knowledge-based technique could give more flexibility and chance of expansion just by rules up-gradation to adjust new and different types of complex documents. However, with the increase in the number of rules, processing time of the system increases, which can further decrease the overall performance.

Murthy et al., (2013) worked upon an algorithm for the division of pre detected words extracted from natural scene images of Devanagari Script. The authors explored existing techniques and pointed out that no method was available for the segmentation of natural scene images of Devanagari script. Their method is suitable for real life applications and achieved an average performance of around 55.77%.

Roy et al., (2016) introduced a new algorithm for the classification of words of Indic handwritten scanned documents with the help of zone-wise data. The authors proposed an improvised framework of word recognition with the separation of the handwritten word images in the horizontal direction into three different zones namely upper, lower and middle. Mostly, the connected characters found in the middle zone are recognized with the help of Hidden Markov Models (HMM). To obtain more accuracy water reservoir-based method was combined with the new framework for the improvised

zone separation and character boundary detection during segmentation. They introduced a new method of sliding window oriented feature descriptors, known as Pyramid Histogram of Oriented Gradient (PHOG) for the classification of components residing in the mid zone. Experimentation was done on Devanagari and Bangla scripts for analyzing the effectiveness of the proposed technique and output showed that the proposed zone-wise recognition method performed better than the traditional recognition techniques.

2.3 Feature Extraction and Classification Techniques

Kahan et al., (1987) worked upon recognition of the Roman alphabet printed text of various fonts and sizes. For the implementation of various techniques like thinning and shape extraction, shape-clustering approach, statistical bayesian classifiers were used for better throughput. The experimentation was done on six dissimilar fonts and satisfactory results were obtained. Complex classes were also resolved using contour analysis and the suspected merged characters were broken and reclassified. However, due to the limitation of insufficient resources the developed system gives average accuracy.

Amin and Mari (1989) developed a technique to recognize the multi-fonts based printed Arabic text image documents. The method worked with simple Arabic characters and characters with one or more complementary character. Accuracy was the main problem with their method, caused due to the complexity involved in Arabic text and its interconnectivity. The authors emphasized the need to explore the vowel diacritics because even human eyes make errors during reading words in isolation.

Mori et al., (1992) presented a study of the advancements in the character classification systems along with the historical commercial advancements. The authors compared the techniques of template matching and also analyzed the character structure. The commercial OCRs were discussed in detail along with methods like expert systems and self-learning neural networks.

Sinha et al., (1993) worked upon hybrid contextual three pass algorithm for reading printed documents. The algorithm was capable to recognize different fonts of any size. The first pass generates the character hypothesis, the second generates word hypothesis, and third verifies the word hypothesis. The algorithm was tested on twenty two multi-font documents of varying quality with satisfactory results. The performance of the overall recognition system could be improvised by refinement of the word hypothesis in different text zones and by using word envelope information in dictionary partitioning.

Chaudhuri and Pal (1997b) worked upon an OCR system for reading printed Bangla and Devanagari scripts. They developed a new algorithm for both the scripts. In contrast, the set of algorithms required for feature sets, classification and knowledge base differs for both of the scripts. The system performs well on clean printed documents only. However, the system's performance decreases on low quality/degraded printed documents.

Jain et al., (2000) compared statistical approaches which were used in different stages of a character recognition system. The authors observed that there was a general unsolved problem in the recognition of complex image patterns because of arbitrary orientation, location and scale. The recognition system should be carefully designed in

terms of; a) defining pattern classes; b) pattern representation; c) feature selection and extraction; d) design of the classifier and its learning; e) selecting the training dataset and test samples; f) evaluation criteria for checking the efficiency of the system. The motive of this work was to produce clean text, robust and exclusive comparison of well-known techniques deployed in the different stages of the classification system.

Lehal (2001) designed a character recognition system for Machine Printed Gurmukhi Text. Recursive contour trace method was used for the segmentation of characters of machine printed document images of Gurmukhi script. For the segmentation of multiple lines in Gurmukhi image documents, method of average core strip height was used. During the recognition process, two different kinds of feature sets were used; a) the number of junction points, presence of loops and their pixel positions were checked and b) total number of endpoints with their position, nature of profiles of different directions were also considered. Further, the classification scheme with the combination of binary tree and the nearest neighbor classifier was used along with the development of a post-processing system for Gurmukhi script.

Bansal and Sinha (2002) proposed an algorithm which was used to separate a pair of connected characters in Devanagari script. The algorithm makes use of the shape related properties of the said script and was implemented in two passes; a) First pass segments words into individual characters with the use of statistical data like height and width of each segmented box, and b) In the second step, segmentation of the hypothesized composite characters was done. A recognition accuracy of around 85% was obtained on the segmented characters. The authors showed that the methodology used in

their work can also be applied on other scripts that has a similar structure like Bangla, Gurumukhi, Gujarati etc. However, they had not mentioned any results on these scripts and the algorithm consume more execution time due to its passes.

Guyon and Elisseeff (2003) introduced attribute and feature selection method to enhance the performance of the predictors by addressing the problem in a practical manner. Linear predictor like a linear Support Vector Machine (SVM) was also used which selects the attribute in two options: a) attribute ranking method, which uses a correlation coefficient; b) nested subset selection method that makes forward or backward selection. The work presents a better picture of the underlying process of data generation. Both of the said approaches were very diversified and supported with theoretical arguments, but there is a lack of unifying theoretical framework.

Ma and Doermann (2003) worked upon recognition of printed Hindi documents using generalized Hausdorff image comparison (GHIC) by performing character segmentation with the Shirorekha (header line) removal. Their experimental results gave recognition accuracy of approximately 88% for images with high noise level and 95% for clean images.

Pal and Chaudhuri (2004) presented a survey of the different character recognition systems on Indian language scripts. Authors detailed various development techniques used by researchers internationally. Limitations of the current systems and improvements required for the implementation of the enhanced character recognition system for the Indian Scripts were also discussed. Some of the Indian languages were not

covered by them in their work and still there is a lot of research scope in regional languages.

Kompalli et al., (2005) introduced a new method for the character classification and dictionary lookup based post processor for machine printed Devanagari script documents. The authors analyzed that most errors occur during the recognition of conjuncts because of confusions of vowels and the virama character. It was analyzed that the half consonants have various structures from full-consonants. To improve the conjunct classification, dictionary based post-processing method was used. Additionally a stage that can separate conjuncts from descender symbols can be implemented for accuracy improvement.

Frias-Martinez et al., (2006) introduced an accurate and time efficient off-line signature classification system implemented with the help of support vector machine (SVM). Authors, compared machine-learning algorithms SVM and multi-layer perceptron's (MLP) for the signature classification task. The experimental setup showed that the SVM outperforms in comparison with MLP and obtained 20% higher accuracy rate than MLP. Also, with the use of SVM lesser training time was consumed.

Lehal and Singh (2006) worked on the development of machine printed Gurmukhi OCR system. They discussed the complexities of the Gurmukhi script and the problems occurred at various stages during the implementation. They also tested the system for multi-font printed text images taken from quality books and digitally printed documents. The authors achieved good results with high accuracy. They concluded that,

still there is room for the improvement in the recognition accuracy of characters and testing is required for low quality printed texts.

Lorigo and Govindaraju (2006) presented a survey of Arabic handwriting recognition and discussed various methods which were applied on different types of image documents. Arabic handwriting recognition was the first survey to give descriptions of test data and recognition rates. They observed that the algorithm styles were changed with the increase of computational power and so as the increase of statistical techniques. Limitations like restrictive lexicons and restrictions on the appearance of text were also reported. Further, they suggested that an algorithm could be developed that are capable to handle large lexicons and used for different types of words. The n-gram model technique could also be explored for the Arabic script.

Meshesha and Jawahar, (2007) worked on self-adaptable character recognition system for scanned documents. They presented a learning system that train itself in an incremental manner and can adapt to a new set of scanned documents for accuracy improvement. The overall system contains advanced learning algorithms that gather and store information for further learning. It enables the character recognition system to accumulate the information for increasing the performance of the overall system and hence improvement in the recognition rate. The system could be further tested for different datasets on various scripts and fonts.

Meshesha and Jawahar (2008) proposed a high performance technique for matching word images of different scripts, printing variation (font, style, size) and various degradations. The authors worked upon a partial matching technique for

structural matching of the word with a number of unknown variants. The implementation revealed that by using the combined adaptive features makes effective management of the different degradations and printing issues in the printed image documents. The efficiency analysis of their work was done only on three language documents including English.

Georgios et al., (2010) worked upon a technique used for the off-line recognition of handwritten scanned documents. The granularity features are obtained from distinct subdivisions of the scanned document image for equal number of front pixels. After that a 2-stage pattern classification method is applied in which the classes having higher values in the confusion matrix are integrated up to some extent. Then, for each of the set of integrated classes, the best fit granularity features from the level which differentiate the classes are deployed. The experimentation has been performed on different handwritten scanned character datasets from CEDAR and CIL. In addition to that experiment was also conducted on handwritten numeric datasets of MNIST and CEDAR.

Pal et al., (2010) presented a novel technique for the character segmentation and classification of complex scanned machine printed documents of Bangla and Devanagari scripts. The proposed technique is independent of document skew and text curvature. The authors utilized the background and foreground data for the classification of complex Indian scripts. The popular approaches of convex hull and water reservoir principle based feature descriptors along with various angular data collected from the characters have been deployed for the classification of varying sized text of Bangla and Devanagari. The technique can be extended on other Indian scripts in which characters are connected through head line and also for the modifier recognition of Devanagari and Bangla scripts.

Siddiqi and Vincent (2010) proposed a method for classifying the writer independent text images. Their aim was to analyze small writing fragments to obtain the individuals writing style patterns. The authors have also checked the visual parameters of writing, direction and curvature by feature computation from writing patterns at various observation stages. They recommended that for extracting the frequent writing patterns, window size should be determined automatically instead of fixed value, which is based upon the writing pattern details. The authors also discussed about the existence of a partial duplicity among some feature values of their dataset.

Xu and Krauthammer (2010) introduced a character detection and extraction technique for biomedical images using the technique of iterative projection histograms. The technique was introduced to improvise the task of text retrieval and classification from biomedical images. They obtained higher accuracy by experimenting their technique on a dataset of random biomedical images which was manually labeled and then, compared the results with the existing techniques. Precision is the main limitation of their algorithm which ruins its performance level.

Singh et al., (2011) performed an evaluation and comparison of performances of curvelets and geometry based features for the character classification of offline Handwritten documents of Devanagari script using k-NN and SVM classifiers. Curvelet based features in combination with k-NN classifier showed higher effectiveness than geometry based features. The results of the recognition system could further be enhanced by using more training samples of different resolutions.

Chattopadhyay et al., (2012) presented a character recognition system for extraction text from low complex videos, which can be employed in an embedded system. The main advantages of the proposed technique were lesser processing time and consumes less memory. The authors achieved high recognition accuracy of around 84.23% than existing video character recognition systems and approximately 180 characters per frame are recognized in just 26.34 milliseconds.

Pal et al., (2012) presented a survey of nine major offline Indian Scripts and discussed various feature extraction and classification techniques. They showed that less amount of work on offline printed character recognition is available. Still there is a need to work upon handwritten character recognition and the technique of machine printed character classification cannot be replicated for handwritten characters because of the huge variation in handwriting styles.

Ramakrishnan et al., (2012) explored a number of machine learning techniques that depends upon feature vectors to get the details about the appearance of an image. These details are very importantly required for an algorithm to be efficient and accurate. In contrast the domain and specific task related feature vectors can give better performance, but needs more manual intervention, are complex and takes lot of processing time. In contrast, general-purpose feature descriptors like SIFT can be easily applied and showed good results for many tasks such as character segmentation, clustering and pattern classification. Mostly, the general-purpose feature vectors are applied on scanned document images and give poor performance during the document image analysis. The authors introduced a new self leaning feature extraction technique

synced with an image domain. The individual image components are first extracted, and after that a feature descriptor is composed by integrating these components over the multiple overlapping portions. The technique was applied on a large dataset of scanned documents and obtained higher results in comparison with the existing techniques of general-purpose feature descriptors.

Sankaran and Jawahar (2012) introduced a character recognition algorithm for Devanagari script. The character recognition results of Devanagari script are much lower than the results available for Roman scripts due to the complex character shapes, style of writing and number of conjugates. The authors provided a solution as Bidirectional Long-Short Term Memory (BLSTM), which uses a recurrent Neural Network in which it is not required to separate a word into individual characters, as most of the recognition errors occur during segmentation of words. The test results showed that the error rate at word level was reduced by 20% and at character level error rate comes down by around 9%.

Bag and Harit (2013) presented a detailed survey on advancements of character recognition system for two major Indian scripts Devanagari and Bangla. The authors analyzed that during past decades fruitful and robust research was conducted on the recognition of characters on many non-Indian scripts. A number of techniques in character recognition system for Devanagari and Bangla scripts have been analyzed along with the reported accuracy levels. Finally, the future directions of research in the character recognition system for Indian scripts are pointed out.

Grafmuller and Beyerer (2013) proposed a character classification system for rough and unpleasant industrial works by investigating various segmentation techniques.

They considered two approaches for segmentation, first one was without prior knowledge and other one was with prior knowledge. Also, the quantity of text lines, words or characters is similar for different document images. They compared various combinations of techniques of feature extraction and classification by performing experimentation of segmentation approach with prior knowledge. The system showed significant improvement, increased the segmentation process speed and gave better classification results with the prior knowledge. However, the performance of the system was not satisfactory without the prior knowledge.

Yi and Tian (2013) proposed method for the identification and obtaining text data from natural scanned or camera captured images. The algorithm worked with the combination of layout analysis, which is based on the position of candidate character and text recognition based on extracting features from character shape. They evaluated the accuracy of the presented method by experimenting on the detection of text regions, classification of the text, and character recognition. The experimental results showed that correlations improved the performance on text classification. However, the algorithm fails to work on the images having lower resolution, single character, over-exposed, multicolor character and non-regular fonts.

Elagouni et al., (2014) presented proposes two distinct neural network based character classification techniques. The first technique separates individual characters in a scanned document image before classification, whereas the later technique by-passes the segmentation stage by the integration of a multi-scale scanning method which permits combined localization and classify characters at each location and scale. Also, to

minimize the errors occurred due to confusing patterns, some linguistic knowledge is integrated. Both the techniques were applied to caption texts integrated in video frames and on text based natural scene images. Results showed that both the techniques gave higher recognition results (93% approximately) in comparison with existing approaches.

Hassan et al., (2014) introduced a novel framework for the classification of binary patters from larger datasets with the help of learning-oriented grouping of multiple feature vectors primarily for two Indian languages Bangla and Gujarati. The multiple kernel learning (MKL) was used for fast classification which successfully integrates distinct features for robust labelling. Also, a new feature representation has been used that can deal with various kinds of deformations and distortions. Their framework performs well on standard data set in comparison with the existing techniques but needs to be explored on another symbol data sets and more Indian languages.

Liu and Jiang (2014) introduced a novel technique for the character recognition of Chinese script, which works on the logic of fuzzy clustering analysis. The Chinese script contains a number of similar looking shapes and ligatures, which makes the task of character recognition time consuming and difficult to implement. To resolve these issues, a fuzzy clustering analysis technique has been introduced to make the system more efficient. The authors make use of the minimum distance algorithm in order to calculate the distance of training and testing samples, which are formulated using binarization of document images. Finally, the characters are classified by finding out the minimum distance value. For experimental work 40 scanned images are used to train the classifier and 10 scanned images are used to test the classifier. The classification results show that

the distances between the testing pattern with the correct pattern was minimum in comparison with the other pattern distances.

Aggarwal et al., (2015) presented two new methods for the classification of offline handwritten Gurmukhi text with the help of gradient features. The effectiveness of the proposed methods was tested on two different handwritten datasets of the Gurmukhi script of size 7000 & 2000 samples of scanned binary images. The first method outperforms than second in terms of accuracy and efficiency. Experimental results show recognition accuracy rates of around 97.38% for Gurmukhi characters and for Gurmukhi numerals around 99.65%. The work can be extended to test the efficiency/ accuracy of the presented method on the combined dataset of Gurmukhi numerals and characters.

Greenhalgh and Mirmehdi (2015) worked upon the technique of identification and classification of characters in traffic sign boards with the help of Maximally stable extremal regions (MSERs) and hue, saturation, and value (HSV) thresholding. The natural scene image structure of traffic sign board helps in ascertaining the search area within the identified image, in which candidate text is then explored. In this technique, initially the individual words of the recognition system are matched from frames on the basis of size. After that, the output obtained from some more recent identifications are grouped, and the histogram from the output of the character recognition system is created for each of the identified words on the basis of weight assigned by the confidence recognition system. For each individual frame, the result of histogram having highest value provides the recognized word for the said frame.

Kamble and Hegadi (2015) introduced a new technique for the character classification of handwritten Marathi language by extracting the features using rectangle histogram based gradient (R-HOG) representation. The authors evaluated the proposed technique on huge dataset collection of distinct handwritten Marathi text images. The experimental setup showed high classification accuracy with the combination of proposed feature vectors and feed-forward Artificial Neural network (FFANN).

Shivakumara et al., (2015) presented an innovative idea for the identification of text in video frames by the integration of Gradient Spatial (GSpF) and structural (GStF) features. In the proposed algorithm, they integrated the shape oriented and the spatial features on boundary, junction and intersection points. Also, the skeleton of the text was straightened for the identification of the scripts. The experimentation using proposed approach has been performed on 970 video frames of six different scripts with different variations and achieved recognition accuracy of around 83%.

Aggarwal and Singh (2016) introduced a new method for the invariant classification of Gurumukhi text with the help of Zernike moments. The authors proposed a framework for calculating the Zernike moments with the use of outer and inner circle mapping techniques. The efficiency of both the mapping techniques was evaluated on Gurmukhi binary dataset and the results show that the outer circle mapping technique gives higher results than the inner circle. A recognition accuracy improvement of around 5% – 10% was achieved with grayscale scanned images.

Li et al., (2016) worked upon a novel technique using Stroke Width Transform (SWT) for extracting the text from videos. Their technique uses a fuzzy based edge

detection algorithm in contrast with the conventional edge detection techniques. They tested the proposed technique on ICDAR 2003 and cute80 dataset. The system was able to detect/ extract text in images having impurities like complex background, blur, cursive, slant, curved, varying font shape and size. But the system fails on text images with high level of blur, false positive, unclear, very small font, transparent, very cursive.

2.4 Post-processing Techniques

Chaudhuri and Pal (1997a) worked upon OCR system for printed Bangla characters. A template-matching approach with a tree classifier was deployed to classify the complex compound characters. The character unigram statistics approach was also used for better efficiency. Further, a dictionary-based error-correction technique was employed for more efficiency and increasing the recognition rate. Finally, the extension possibilities of their work for Devanagari Script were also discussed, but work does not include any practical implementation.

Lehal and Singh (2002) presented a post-processing method on clean printed document images for enhancing the classification accuracy of the Gurmukhi script character recognition. This was the first attempt to develop a post-processor for Gurmukhi script as per their knowledge. For the design and development of method, the authors had combined the statistical data of Gurmukhi script syllables, dataset look-up and some rule based techniques on Punjabi grammar. The proposed method utilized the information which was derived from the Punjabi language text dataset. Basically, the dataset was used to get the occurrence count of various characters, at a particular location

in the word. The dataset was also used to get the information useful for distinguishing the similar looking words. This information was used to correct character recognition errors.

Gupta et al., (2011) worked upon an offline text detection and recognition system for handwritten English scanned documents with the help of heuristics and artificial intelligence. The authors incorporated three combinations of Fourier transformation feature descriptors and for the classification purpose Support vector machine (SVM) was deployed. Finally, the overall recognition results were improved using post processing technique along with lexicon.

Dutta et al., (2012) introduced a novel character recognition method for the minimization of the error rate at word level on poor degraded Indian script scanned documents. The proposed character recognition system performed well on quality scanned documents in comparison with documents with degraded text. To resolve these issues, the authors proposed solution to recognize character n-gram on scanned document images instead of using during post-processing. The advantage of exploring the presence of extra information in the character n-gram scanned document images, it becomes easier to differentiate between the confusing characters during the recognition stage. The technique efficiently deals with the degradations like broken and merged characters commonly found in scanned document images. Experimentation was performed on Malayalam and English scanned document images which show satisfactory increase in the recognition accuracy on degraded and poor quality image documents.

Gonzalez and Bergasa (2013) worked upon an algorithm which can read the characters in natural scene images. The algorithm was implemented in two phases: a)

Text was searched in the image using a collection of simple and fast-to-compute feature vectors that can efficiently distinguish between text and non-text patterns. These features depend upon the gradient and geometric properties; b) classification of prior detected text was done using a gradient feature extraction technique for the recognition of a single character. Dynamic Programming (DP) was used for the correction of grammatical errors. Work could be extended by categorizing the images on the basis of text extraction like spam filtering for images, document recognition for some of the records which could be completely or partially composed of scanned images.

Kumar and Lehal (2016) worked upon a new method of post processing for the correction of errors left by the character recognition stage of Devanagari script. The method works on the basis of confusion matrix formulated from a huge dataset of Hindi language and make possible top suggestions for the correction of non-Hindi words. The new technique has been experimented on large dataset with distinct words collected from the output of Hindi character recognition system and obtained satisfactory results. The method can be used with other character recognition systems with some customization and improvements.

Chapter 3

Image Pre-processing and Segmentation

In this chapter, segmentation stage is discussed in detail with illustrative examples. The overall performance of an Optical Character Recognition (OCR) system depends on accurate and efficient segmentation algorithms. The poor segmentation results in misclassifications of characters (Casey and Nagy, 1982; Casey et al., 1996; Garain et al., 2002; Bansal and Sinha, 2002a; Jindal et al., 2005; Jayadevan et al., 2011; Alginahi, 2013; Grafmller et al., 2013). New algorithms (named as "***Pihu***" method) have been developed for segmenting columns, lines and characters from machine printed Dogri language documents. Further, to evaluate the robustness of the new algorithms, these were tested on pre-detected words in natural scene images. The experimental results show that the proposed segmentation algorithms are time efficient and more accurate. A brief overview of the standard system design and proposed system architecture for designing an OCR system for printed Dogri language has been given.

3.1 Image Pre-processing

The basic requirement of a recognition system is the input text image which is provided as a grayscale JPG image scanned at 300 DPI. Generally, the scanned image contains anomalies like noise, skew etc., and needs to be binarized to perform various operations. To suffice the purpose need of preprocessing techniques arise. With the use of preprocessing techniques, the document image can be enhanced for the quality improvements and to obtain higher recognition results. Generally, grey scale images are

used as input to a character recognition system because of less complexity and processing time. In comparison, the processing of color images is a bit complex and takes more computation time, so the color images are initially converted into grey scale images. The final output of preprocessing stage is an improvised binary image that contains text only.

Image noise, skewness and degradations are major issues with a character recognition system. The reason of noise in an image can be due to print quality, aging, non-uniform illumination, scanner quality and resolution (Horng et al., 2013; Li et al., 2014; Srinivasa et al., 2016; Thilagavathy and Chilambuchelvan, 2016). These types of noise variations increase the errors in any character recognition system. So, it is necessary to remove the image noise before any further processing. Noise filtration techniques are deployed to remove/ minimize the high frequencies present in the image i.e., smoothing of the image, and the low frequencies, i.e., enhancement of the image corners. The image improvement methods are highlighted in the frequency and spatial domains, i.e., Fourier transforms. In some of the cases, Fourier transforms need substantial calculations, and a fewer times it is not feasible to use. The combination of a convolution matrix of size 3x3 or 5x5 with an image is simple and efficient than working with Fourier transforms and multiplication.

3.1.1 Image Filters

In a character recognition system, the image noise can be removed in grey color images using filters like maximum, minimum, range, mean and median. Out of these, the median and mean filters effectively remove the isolated pixel noise. To obtain a smooth texture image, Gaussian blur and average filters can also be deployed. In literature survey, it is explored that the mean and median filter techniques give more attention on image

foreground pixels and removes/ minimizes the background pixels. In a grayscale image the pixels which represents objects or area of interest are called foreground pixels and the remaining are known as background pixels.

Mean filter (average or low-pass frequency filter) is the simplest method for smoothing images, i.e., minimizing the intensity variations between pixels. Its major goal is to reduce the noise in the image by reducing the presence of spatial intensity derivatives. Using this method, every pixel value of the image was replaced with the computed mean value (calculated average value) of all the neighboring pixels. It facilitated in eliminating pixel values that did not have any contribution in the meaningful data i.e., those pixels that did not represent the nearby meaningful area were ignored. The mean filter was placed near a kernel that represented the structure and size of the neighborhood to be sampled during the calculation of mean intensity value. Commonly used kernel matrix were of size 3×3, 5×5, 7x7, 9x9 etc. (with positive elements) depending upon degree of smoothing. It was observed, that the larger size kernel matrix sometimes produced a blurring effect, that caused smaller objects to mix with the image background (Nixon and Aguado, 2012; Alginahi, 2010) leading to distortion of the image. Therefore, an optimum size of the kernel matrix, was further used.

Median filter was considered an effective technique for the removal of impulse noise (salt and pepper). The kernel matrix size mentioned above, therefore sorted out each of the pixel values and calculated the median value that could replace with the requisite pixel value. In case, if the neighborhood had an even number of pixels, then the average of two middle pixel values was used. The working of median filter is shown in Table 3.1.

Table 3.1 (a) Original character image (b) grayscale intensity values of image (c) 3x3 matrix (d) Sorted values and (e) Filtered center pixel median value of matrix

(a)

(b)

222	222	214	224	224	212	217	213	213	224	216	214
217	212	220	216	211	212	211	220	211	218	218	211
220	212	224	206	204	226	213	207	213	196	211	210
136	128	136	135	133	148	153	147	145	121	133	138
93	98	92	106	88	72	87	58	49	88	84	87
190	220	200	203	198	195	172	51	81	197	195	200
227	184	106	84	106	156	168	74	54	146	151	161
191	97	61	105	114	91	76	49	60	59	60	67
181	61	86	201	199	197	185	53	56	107	149	153
178	64	136	211	201	225	173	56	70	184	208	224
193	91	123	197	213	163	80	70	71	209	206	195

(c)

226	213	207
148	153	147
72	87	58

(d) 58,72,87,147,**148**,153,207,213,226

(e)

	148	

Here, the pixel intensity values of 3x3 matrix Table 3.1(c), taken from the grayscale intensity values of the original image Table 3.1(b), were initially sorted and then, median value 148 was selected to replace with the central pixel value of the matrix (Miciak, 2008).

3.1.2 Image Skewness

Skewness is another common problem that causes degradation in a document image. It can often be created during scanning/ picture clicking. The skewed text can result in wrong classification/ recognition. So, document skew detection and correction are required for the better classification. Some commonly used techniques for skew correction are Projection profiles, Hough transform and nearest neighbor clustering (Farrow, 1994; Yu and Jain, 1996; Min, 1996; O'Gorman, 1993).

Hough transform based skew correction has been used in this work. The main advantage of using Hough transform is its simplicity and accuracy. In this technique, the document images are preprocessed to highlight black blobs in presence of text. Then the highlighted black blobs are thinned to get them into single rows. Hough transformation is applied to these rows, then peaks are visible in the parametric space. On the basis of these peaks the skew in the text is calculated and corrected. The skew estimation will be poor if the text lines are scattered (Kapoor, 2004).

3.1.3 Image Thresholding

The adaptive threshold operation is performed on the preprocessed grayscale image to differentiate the foreground and background information in a document image. Thresholding is frequently used technique in the area of image segmentation (Otsu, 1979; Niblack, 1986; Sauvola et al., 2000; Sezgin et al., 2004; Yan et al., 2005; Shafait et al., 2006; Shafait et al., 2008; Toh and Isa, 2010; Nikolaou et al., 2010; Mesquita et al., 2014; Huang et al., 2014). The process of thresholding transforms a grayscale image into a

binary image i.e., pixel values of 1 and 0 (Trier et al., 1995; Badekas et al., 2005). The thresholding operation is a grey value reorganization operation f defined as:

$$f(g) = \begin{cases} 0 & if\ g < t \\ 1 & if\ g \geq t \end{cases} \quad (3.1)$$

where g is the gray value and t is the threshold value.

There are a number of thresholding techniques which have been discussed in the literature. In a character recognition system, generally, it is applied to grey scale image document. There are two categories of thresholding viz, global and local. The global thresholding methods select single threshold value of the document image which is often based on the calculation of average or mean value of the gray levels of the document image. On the other hand, the local (adaptive) thresholding divides the image document into sectors and chooses different gray level values for each sector.

In this work, Niblack's adaptive image thresholding technique has been used to differentiate background or foreground pixels with the use of statistical and texture feature measures i.e., image binarization and noise removal (noise occurred due to scanning device, document condition etc.). The technique works on the principle of local thresholding technique and provides local information of the pixels with respect to the neighborhood points (Niblack, 1986). These quality enhancements improve the accuracy of the segmentation process and helps in achieving higher recognition results. For the detection and correction of skew that mostly occurs during scanning, the header line based skew estimation technique (Pal and Chaudhuri, 2001) has been deployed. Basically, it is the angle which the lines containing text makes in the horizontal direction. After quality improvements of the image, it is forwarded to the proposed segmentation function.

3.2 Image Segmentation

In segmentation stage, the image is divided into meaningful subunits on the basis of horizontal/ vertical white pixel space. The text area is identified on the basis of the presence of black pixels in the image and then the regions containing black pixels are segmented i.e., only relevant pixels are grouped to their corresponding object segments. For segmenting an image into meaningful sub regions, the following steps are used:

a) The preprocessed image is analyzed for the presence of columns.

b) If columns exist, then those are segmented and processed as individual images.

c) Individual lines are detected and segmented.

d) Words in a line are detected and segmented.

e) Finally, characters detection and segmentation is performed.

The overall working of segmentation stage is shown in Figure 3.1.

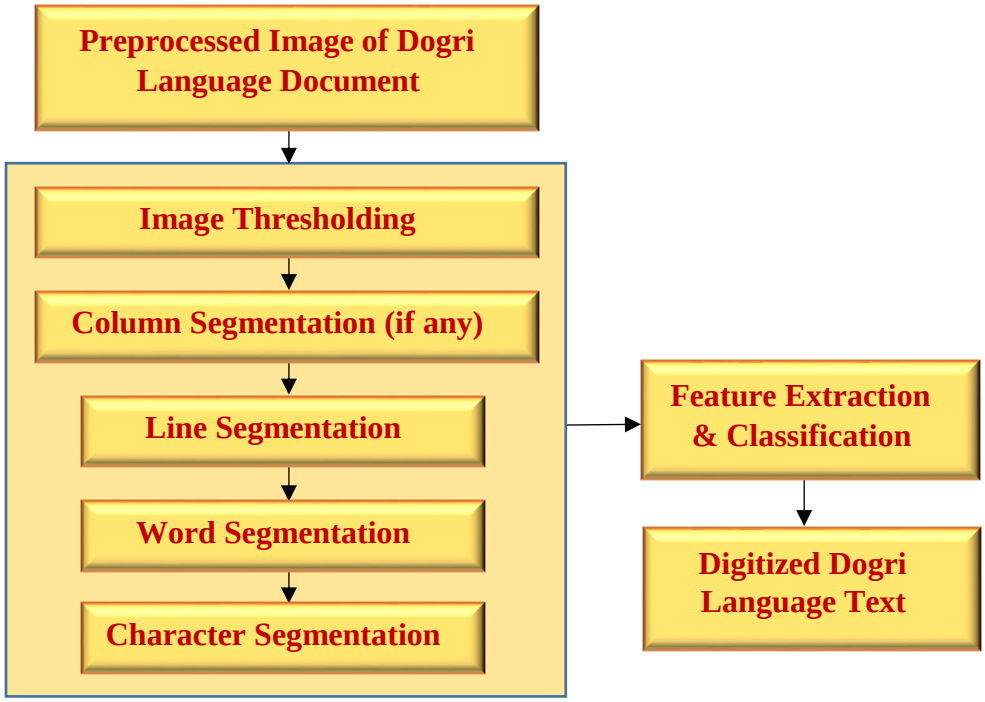


Figure 3.1 Complete Segmentation Process

A new Shirorekha (header line) based method has been proposed for the segmentation and recognition of Dogri language documents. The aim of this method is to illustrate the importance and role of the character shapes for accurate recognition. The Table 3.2 shows text images scanned from Dogri language book. Two columns text image is shown in Table 3.2(a) and single column text image is shown in Table 3.2(b).

Table 3.2 Dogri language scanned image

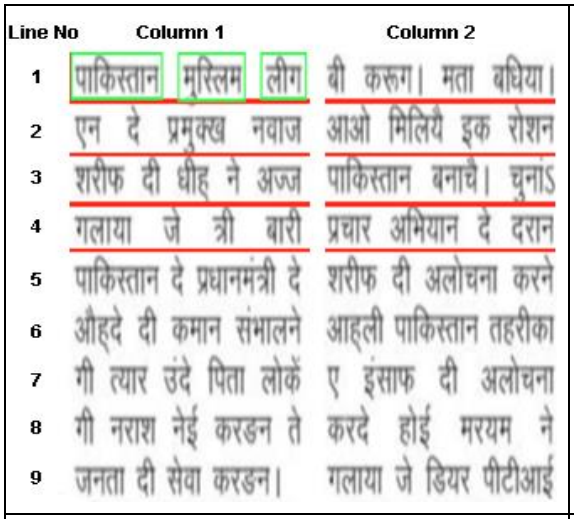
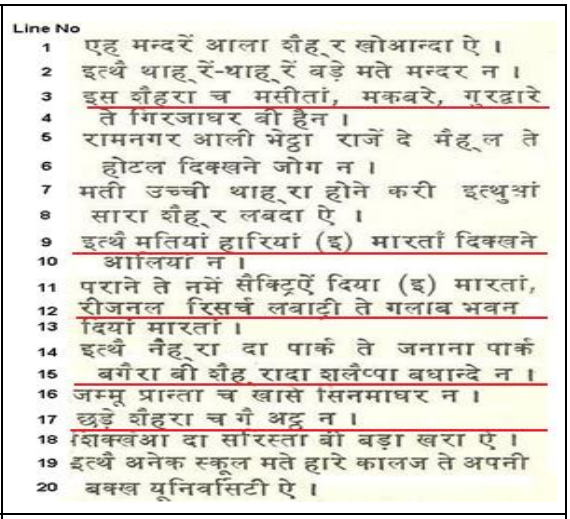
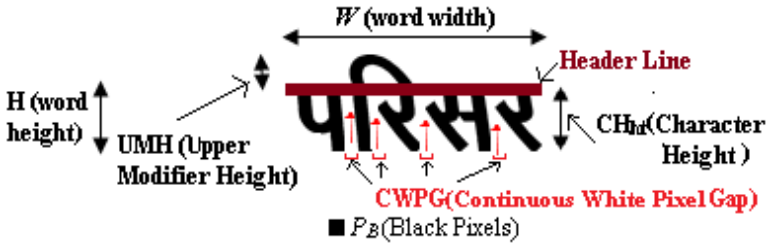
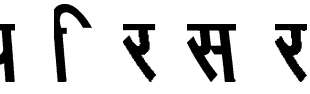
	
(a) Double column	(b) Single column

Table 3.3 (a) Original image, (b) Components of word, (c) Preprocessed image and (d) Segmented characters

(a)	
(b)	
(c)	
(d)	

A sample image of a word and its various components are shown in Table 3.3. The original input image sample is shown in Table 3.3(a) and the various components are illustrated in Table 3.3(b). Initially the input image is preprocessed using existing techniques to obtain the output as in Table 3.3(c). After cleaning the image, it is segmented into meaningful subunits as shown in Table 3.3(d).

3.2.1 Flaws of Existing Methods

In this section, flaws of the existing methods, including (Murthy et al., 2013) are elucidated. Correct segmentation of lines, words and characters is very important for accurate recognition. As discussed in the literature, existing methods remove the header line during character segmentation, which is one of the main reasons for character misclassification. On applying the existing segmentation methods, many Devanagari script / Dogri language characters are left partially/ under/ over segmented. There are a number of such poorly segmented images and some of the images are shown in Table 3.4.

The main reasons for poor segmentation are:

a) Loss of structural information during the segmentation of words into individual characters. The structural data is importantly required for unique recognition of characters.

b) Unsuitable method to detect the salient regions. During this stage, single threshold value β was chosen to segment characters.

c) The uneven blur in images, global threshold value selection, curved and closely handwritten text also contribute in the poor segmentation.

Table 3.4 Identified misclassified cases of segmented characters due to the absence of the header line

S.No.	Segmented Character		Effects/ Issues Due to header line removal
	With header line	Without header line	
1.			Will be misclassified with the English digit four
2.			Will be misclassified due to loop loss i.e., Important Structural Property
3.			Will be misclassified due to confusion between ya and tha
4.			Will be misclassified due to loop loss i.e., Important Structural Property
5.			Will be misclassified due to confusion between bha and maa
6.			Will be misclassified with the English digit three
7.			Will be misclassified with the English digit two due to loss of Important Structural Property
8.			Will be misclassified due to loop loss, i.e., Important Structural Property
9.			Will be misclassified as English character U / V
10.			Will be misclassified with as English / Roman character I / digit 1
11.			Will be misclassified as Gha
12.			Resemble with English letter o / digit 0

It was observed from the literature survey that most of the segmentation techniques on the Devanagari script (machine printed text), segment a word into smaller units with the removal of the header line. Almost, all the characters of the Dogri language (Devanagari script) contain Shirorekha (header line) on the upper portion, which makes the existing

segmentation a difficult and complex problem. In almost all of the character segmentation instances, once this header line is removed, it results in loss of important structural information as shown in Table 3.4

The structural loss has been explained using the example of the word पाप. When this word is segmented into characters using existing header line removal method, then, the output will be ५।५. During the recognition process, the first character will match with English digit four ५, the second character will match with English digit one । and third character will match with English digit four ५. Therefore, on the basis of closest match, the recognition engine output will be "four-one-four" (५।५) instead of desired output characters "paap" (प ा प). Also, few more similar type of cases are illustrated in the Table 3.4.

The Figure 3.2 clearly shows the loss of important structural information due to removal of the header line using graph comparison of gray pixel value profile of the character प and ५. With the removal of a header line, the unique shape of the character has changed and it can will be misclassified with some other pattern during a classification stage. In the light of this, the existing methodologies clearly outline the limitations and emphasize the need of certain improvements in terms of efficiency and higher recognition accuracy.

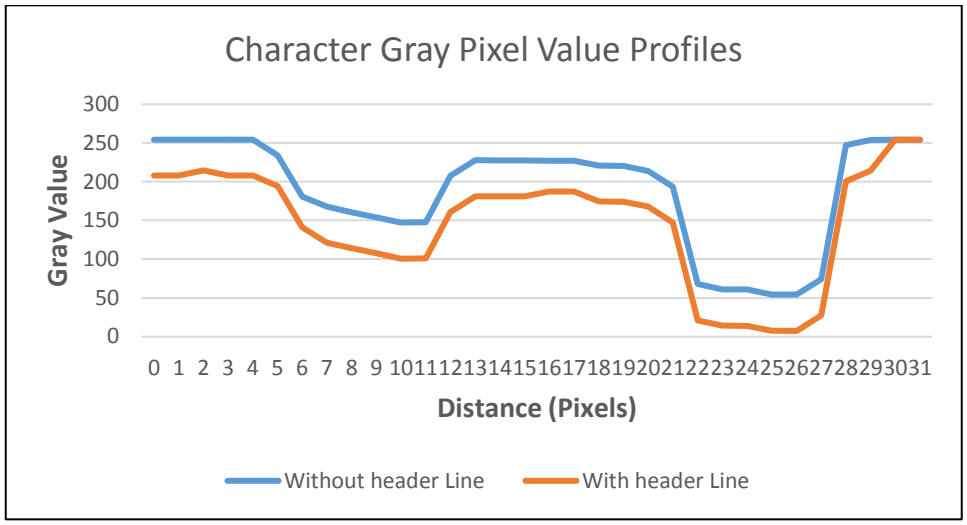


Figure 3.2 Gray pixel value variation graph for character pa (प)

3.2.2 Average Occurrence Frequency of Characters

The average occurrence frequency of Devanagari script based characters in a text page is shown in Table 3.5.

Table 3.5 Character wise average percentage occurrence frequency

Serial No.	Symbol	Average occurrence frequency
1.	प	3.0%
2.	थ	1.0%
3.	य	2.77%
4.	भ	1.5%
5.	म	4.0%
6.	ध्र	1.5%
7.	उ	2.0%
8.	श	0.5%
9.	ण	0.5%
10.	ा	10.12%
11.	घ	0.5%
12.	ठ	0.2%

Table 3.5 represents a data set derived from around 2,84,870 characters arranged from articles, story and poem books and is based upon the occurrence frequency of characters in a standard text page of A4 size (Pande, 2010).

3.3 Proposed Method- "*Pihu*"

To resolve the flaws of the existing method, discussed in the above section, a new method, named "**Pihu method**" is proposed. The proposed method "***Pihu***" is implemented using MATLAB software. The pre-processed image is provided as input and the segmentation process is accomplished using the following steps:

Step 1. The pre-processed binary image be named as IMG.

Step 2. The IMG height (H) and width (W) of the pre-processed scene image is calculated as:

$$\left.\begin{aligned} H &= \sum_{i=1}^{M} Y(i) \quad , \quad \forall\ 1 < i < M \\ W &= \sum_{j=1}^{N} X(j) \quad , \quad \forall\ 1 < j < N \end{aligned}\right\rangle \tag{3.2}$$

where $Y(i)$ and $X(i)$are the number of pixels along the y axis and the x axis respectively, M is the m^{th} pixel along the y axis, N is the n^{th} pixel along the x axis.

Step 3. The row with maximum black pixels ($P_B(HL)$) in the top 30 % area of the IMG, is then calculated.

$$P_B\,(HL) \;=\; max\{HL_1, HL_2, HL_3, \dots, HL_k\}. \tag{3.3}$$

where $HL_1 = \sum_{j=1}^{W} P_B(1,j)\,, HL_2 = \sum_{j=1}^{W} P_B(2,j)\,, \dots, HL_k = \sum_{j=1}^{W} P_B(k,j)$ are the candidate header line rows in the top 30% area of IMG containing black pixels (P_B). Out of these, row having maximum number of black pixels is selected as header line $P_B(HL)$.

Step 4. Continuous Vertical White Pixel Count ($CVWPC$) is calculated using the Eq. (3.4), from the bottom left corner up to $P_B(HL)$. The $CVWPC$ is calculated for the whole IMG as shown in red color lines in Table 3.3(b).

$$CVWPC = \sum_{i=R_{n,0}}^{P_B\,(HL)} P_W(i) \quad , \quad \forall\ i\ \neq 0\ , 1 < n \leq H \tag{3.4}$$

where $R_{n,0}$ is the n^{th} row of first column and $P_W(i)$ denotes the white pixels

Step 5. The image IMG is segmented with the following function:

$$f(char\_cut) = \begin{cases} \sum_{i=R_{n,0}}^{P_B(HL)} \sum_{j=1}^{W} IMG_{i,j}\ , \boldsymbol{True}, \text{if}(Mid(cvwpc) \geq 95\% \cup (5\%\ P_B \in 15\% LMH)) \\ \boldsymbol{True}, \quad CVWPC \geq 95\% \\ \boldsymbol{False}, \quad otherwise \end{cases} \tag{3.5}$$

where $IMG_{i,j}$ is the image, LMH is the lower modifier height, P_B are black pixels.

During the segmentation of image IMG into subunits, there may exist a compound character (CC) or characters depict as CC due to the presence of lower modifier. To check the possibility of such cases, the width of the character (CH_{wd}) to be segmented is calculated and compared with the average character width (CH_{avg_wd}).

$$\left.\begin{aligned} CH_{wd} &= f(char\_cut)_i\ , \quad 1 < i \leq L \\ CH_{avg\_wd} &= \frac{CH_{wd}(1) + CH_{wd}(2) + \cdots + CH_{wd}(L)}{L} \end{aligned}\right\rangle \tag{3.6}$$

where $f(char\_cut)$ represents segmented characters and L is the total character count.

Then, following functions are executed after verification:

Case (i): If $CH_{wd} > CH_{avg\_wd}$ and black pixels (P_B) are present in the middle area (Mid_Area) as shown in red color circle in Figure 3.3, then the sub image forms CC and is segmented using following function:

$$f(CC) = \begin{cases} \sum_{i=1}^{P_B(HL)} \sum_{j=1}^{W} IMG_{i,j} \ , & \textbf{\textit{True}}, \ \text{if}(P_B \in Mid\_Area \cup CH_{wd} > CH_{avg\_wd}) \\ & \textbf{\textit{False}}, \ otherwise \end{cases} \quad (3.7)$$

where Mid_Area is the middle area shown in Figure 3.3 and P_B are the black pixels.

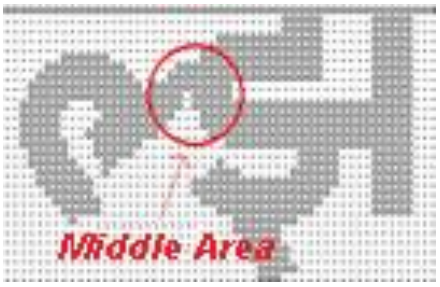


Figure 3.3 Sample binary image of a compound character.

Case (ii): If $CH_{wd} > CH_{avg\_wd}$, $Mid(cvwpc) \geq 95\%$ and remaining black pixels belongs to 5% bottom area as shown in Figure 3.4, then, the sub image is segmented using the function in Eq. (3.5).

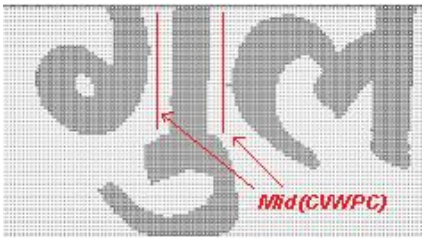


Figure 3.4 Sample binary image with lower modifier covering characters.

Step 6. The upper modifier is segmented using following function:

$$f(UM) = \begin{cases} \sum_{i=P_B(HL)}^{H} \sum_{j=1}^{W} IMG_{i,j} \ , & \textbf{\textit{True}}, \ if \ \sum_{i=P_B(HL)}^{H} P_B > 1 \\ & \textbf{\textit{False}}, \ if \ \sum_{i=P_B(HL)}^{H} P_B = 0 \end{cases} \quad (3.8)$$

where function $P_B(HL)$ is header line, (UM) is upper modifier and P_B are the black pixels.

The segmented UM is joined with the character using following function:

$$f(J) = \begin{cases} \sum_{i=1}^{H} \sum_{j=1}^{W} IMG_{i,j}, & CH_{temp} = CH_{curr} + UM, \ \text{if}(CH_{curr} < CH_{avg\_wd} \cup P_B \in Com\_Area) \\ & CH_{temp} = CH_{next} + UM, \ if(CH_{next} < CH_{avg\_wd} \cup P_B \in Com\_Area) \end{cases} \quad (3.9)$$

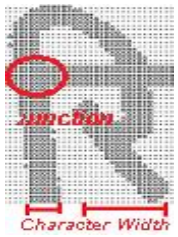
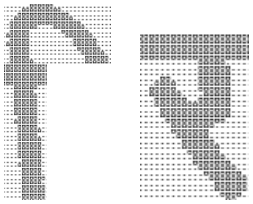
where CH_{temp} stores temporary character, CH_{curr} stores the current character, CH_{next} stores next character, Com_Area is the common/ joint pixel area and P_B are the black pixels.

The upper modifier is joined with a character, if following conditions satisfy:

a) Upper modifier having joint pixels (junction) with left character as shown by a red color circle in Table 3.6(a).

b) The width of the left/ right character should be minimum as shown in Table 3.6(b).

Table 3.6 Sample binary image (a) before segmentation (b) after segmentation

	
(a)	(b)

Similarly, other upper modifiers like ◌ॅ, ◌ॆ, ◌ॏ and ◌ॎ are segmented with current / right characters. Almost, all of the upper modifiers have a single junction point of contact with the current/ right character. If the width of the current character does not meet minimum width condition, then, upper modifier is segmented separately.

Step 7. The lower modifier height (LMH) using the average height of the character, is calculated.

$$LMH = CH_{ht} - CH_{avg\_ht} \tag{3.10}$$

Then, lower modifiers are segmented using following proposed function:

$$f(LM) = \begin{cases} \sum_{i=1}^{HL}\sum_{j=1}^{W} IMG_{i,j} , & \textbf{\textit{True}},\ if\ CH_{ht} \geq (CH_{avg\_ht} + LMH) \\ & \textbf{\textit{False}},\ if\ CH_{ht} < (CH_{avg\_ht} + LMH) \end{cases} \tag{3.11}$$

where LMH, CH_{ht} are heights of lower modifier, character and CH_{avg_ht} is the average height of a character.

The advantage of storing the modifiers separately optimizes the character class count and also helps in reducing the processing time during the character recognition process. The overall working of the proposed segmentation algorithm the "***Pihu***" Method, is illustrated in the Flow chart shown in Figure 3.5.

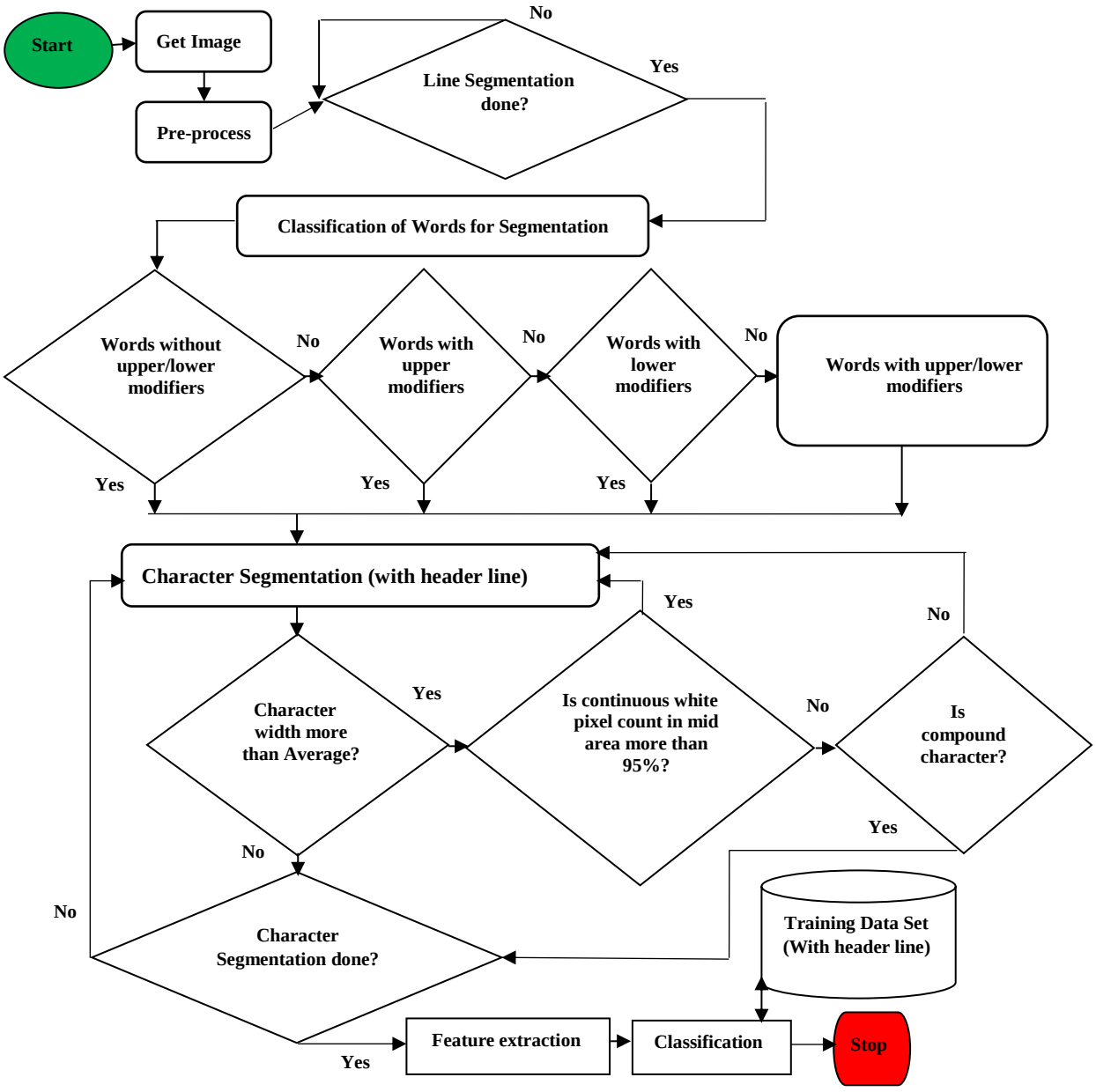
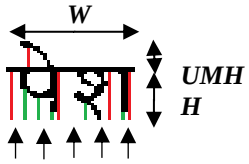
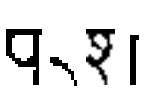
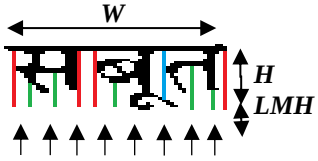
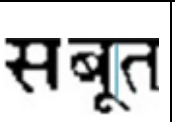
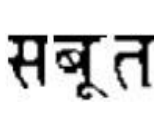
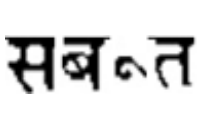
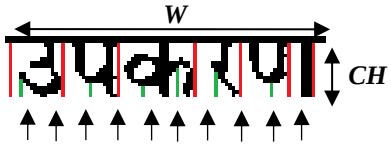
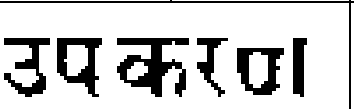


Figure 3.5 Overview of the proposed segmentation algorithm

3.4 Experimentation Work

To validate the proposed method "***Pihu***", it was applied on a number of machine printed Dogri language and Devanagari script scanned documents. The coding was done using c# (Microsoft Dot Net professional) and MATLAB. The images were stored in the form of binary text files that contain the values 0 and 1. To show the effectiveness of the proposed method "***Pihu***", segmentation steps on a few of the scanned images of the Dogri language data set is shown and explained using Table 3.7.

Table 3.7 Character segmentation (a) Input Image, (b) First pass, (c&d) Upper/Lower modifiers

Case-1				
Case-2				
Case-3				
	(a)	(b)	(c)	(d)

Accepted depth level Rejected depth level > 95% In between continuous white pixel area

Case-1: Words with Upper Modifiers

This case deals with segmentation of characters and upper modifiers ◌ॅ . The presence of black pixels above header line indicates the existence of upper modifier. In Table 3.7(a), red line (above header line) represents the white pixel gap between upper modifier and header line. Then, widths of current and next character are checked. The upper modifier is analyzed for its independent segmentation/attachment to the character having a lesser

width and common black pixels. In this case, the upper modifier is segmented separately and remaining characters are segmented (with header line) on the basis of continuous vertical white space gap up to header line. As shown in Table 3.7(c), with the addition of header line, there are high chances of accurate classification of characters उ, ए, र, प, श and ि which are otherwise misclassified.

The character ि and ी are segmented with upper modifier, if all of the following conditions are satisfied:

a) Upper modifier has common pixels (junction point) with current and left/ right character.

b) If width of current and left/ right character is minimized, then the upper modifier connects with the character having less width.

Characters like ी, ी and ी are segmented individually. These characters have a single junction point of contact with the current character and clear vertical white pixel gap is available up-to header line. In some cases the upper modifiers like ी, ी, ी and ी are segmented separately on the basis of absence of character "dandi".

Case-2: Words with Lower Modifiers

This case is more complex than the earlier one due to the presence of lower modifier ी. As shown in Table 3.7(a), the lower modifier is covering consecutive characters at the bottom area. It results in unavailability of clear vertical white pixel gap between consecutive characters from the bottom of the word, which makes separation of characters more difficult. To resolve this problem, a four step procedure of the proposed method was used and is discussed below:

Step-1: Initially average character height was computed and matched with the previously stored height of neighboring characters. If the $CH_{ht} > CH_{avg\_ht}$, then, the bottom area of the character was scanned for black pixels. These black pixels were considered as lower modifier pixels and are saved with that character as shown in Table 3.7(c) case-2.

Step-2: Afterwards, separated characters were checked for their average width and height. If $CH_{wd} > CH_{avg\_wd}$, then the character was inspected further for the existence of complex character, or, for the presence of presence of lower modifier covering consecutive characters as illustrated by blue color line in Table 3.7(b). The connected characters cannot be separated due to the presence of lower modifier ◌ॅ.

Step-3: The *cvwpc* between the characters connected due to lower modifier were computed. Then, if the $cvwpc \geq 95\%$ of CH_{ht} and the black pixels presence was in bottom area only, further segmentation was performed, the exemplified result of which is shown in Table 3.7(c). The case, in which continuous vertical white pixel gap height was less than 95% of total height or the black pixels were present in the middle area, then the characters were considered as compound characters and were left unsegmented.

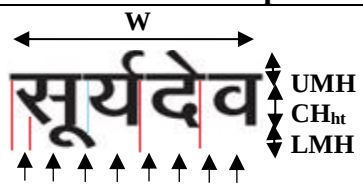
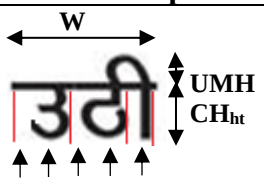
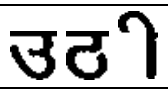
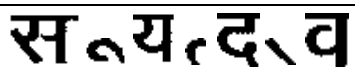
Step-4: As a final step, on the basis of average character height, lower modifier was separated as shown in Table 3.7(d). i.e., The black pixels present in the bottom area, in excess of average character height was segmented.

Case-3: Words Without Modifiers

Characters shown in Table 3.7(a) were segmented along with header line by traversing from the left bottom corner to top (up to the header line) to obtain the continuous vertical white pixel space between characters. On the basis of this white pixel space, word was

segmented into characters as shown in Table 3.7(b). During segmentation of characters all the conditions regarding the presence of upper/lower modifiers, compound character was also checked. As shown in Table 3.7(b), it was clear that the characters उ, प, क्क, र, ए and ा were correctly segmented with the proposed technique, and therefore, would result in accurate classification during the recognition phase. Two more sample images of words segmented using proposed method "***Pihu***" are shown in Table 3.8.

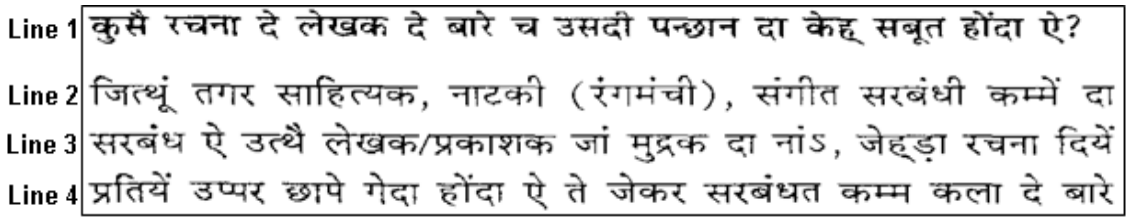
Table 3.8 Character segmentation (a) Input Image, (b) First pass, (c&d) Upper/Lower modifiers

	Example-I	**Example -II**
(a)		
(b)		
(c)		
(d)		
Accepted depth level Rejected depth level Continuous white pixel count > 95%		

3.4.1 Segmentation Outcome

Further, the effectiveness of the proposed method "***Pihu***" in segmentation process was also examined and few of the segmentation results obtained from machine printed scanned documents of Dogri language are given in Table 3.9. Because of space and time constraints, it was not possible to show all the segmentation results, but for illustration purpose, a few lines from one of the documents of the data set is shown in Table 3.9.

Table 3.9 Image segmentation (a) Input Image, (b) Segmented lines, (c) Segmented words, (d) Segmented characters of line number 1, (e) Partially segmented characters and (f) Segmented characters in 2nd pass



(a) Input Image

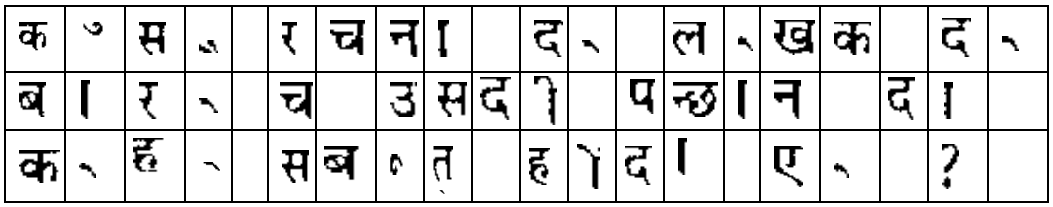
Line No	Segmented Line Images	Total Number of	
		Words	Characters
1	कुसै रचना दे लेखक दे बारे च उसदी पन्छान दा केह् सबूत होंदा ऐ?	15	46
2	जित्थूं तगर साहित्यक, नाटकी (रंगमंची), संगीत सरबंधी कम्में दा	13	49
3	सरबंध ऐ उत्थै लेखक/प्रकाशक जां मुद्रक दा नांऽ, जेह्ड़ा रचना दियें	13	50
4	प्रतियें उप्पर छापे गेदा होंदा ऐ ते जेकर सरबंधत कम्म कला दे बारे	13	45

(b) Segmented lines

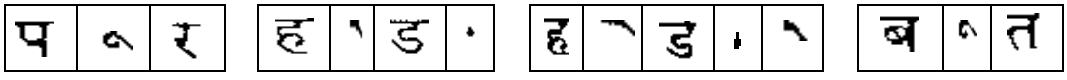
Line No	Segmented Word Images														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	कुसै	रचना	दे	लेखक	दे	बारे	च	उसदी	पन्छान	दा	केह्	सबूत	होंदा	ऐ	?
2	जित्थूं	तगर	साहित्यक	,	नाटकी	(	रंगमंची	)	,	संगीत	सरबंधी	कम्में	दा		
3	सरबंध	ऐ	उत्थै	लेखक/प्रकाशक	जां	मुद्रक	दा	नां	ऽ	,	जेह्ड़ा	रचना	दियें		
4	प्रतियें	उप्पर	छापे	गेदा	होंदा	ऐ	ते	जेकर	सरबंधत	कम्म	कला	दे	बारे		

(c) Segmented Words

(d) Partially Segmented Characters due to presence of lower modifier



(e) Segmented Characters from Line No. 1



(f) Fully segmented Characters from partially segmented characters

A paragraph of preprocessed image document of Dogri language is shown in Table 3.9(a). The proposed method "***Pihu***" was applied on the image shown in Table 3.9(a), to segment it into individual characters. In the first step, the input image is segmented into lines as shown in Table 3.9(b) and then, into words shown in Table 3.9(c). The words are then cleaved into individual characters which are shown in Table 3.9(d).

Table 3.10 Segmentation results using "***Pihu***" method on some of the Dogri language documents

Document	**Total Lines**	**Segmented**	**Total Words**	**Segmented**	**Total Characters**	**Segmented**
Dogri1.jpg	24	24	321	321	1011	1008
Dogri2.jpg	26	26	342	342	1042	1037
Dogri3.jpg	25	25	335	335	985	979
Dogri4.jpg	26	26	348	348	1049	1042
Dogri5.jpg	24	24	322	322	1018	1013
Dogri6.jpg	24	24	327	327	1028	1022
Dogri7.jpg	26	26	346	346	1057	1051
Dogri8.jpg	27	27	360	360	1128	1123
Dogri9.jpg	27	27	352	352	1134	1128
Dogri10.jpg	25	25	330	330	1014	1007
Accuracy	**Line = 100%**		**Word = 100%**		**Character = 99.46%**	

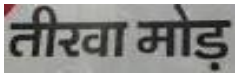
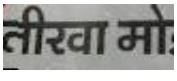
During character segmentation, some of the words are partially segmented like shown in Table 3.9(e) due to the presence of lower modifiers. These categories of partially segmented images are further segmented in second pass and the results obtained are shown in Table 3.9(f). The proposed method "***Pihu***", thus, successfully segmented all the

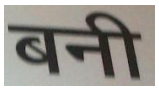
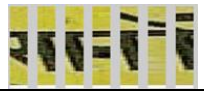
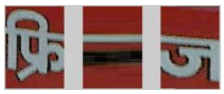
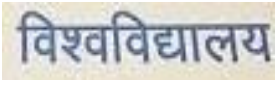
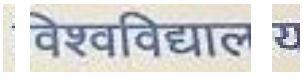
document images of Dogri language documents with different illumination levels and degradations and results obtained on some of the images are shown in Table 3.10.

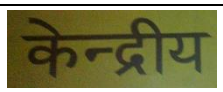
The performance and accuracy of the proposed method "***Pihu***" was verified on two different challenging natural scene image datasets of Devanagari words. The first dataset (Dataset-I) was formulated by collecting 145 natural scene images from various sources such as signboards, shop names and the internet. The second dataset (Dataset-II) contains 130 natural scene images of information sign boards, shop names, banners and posters and is the only reported available collection of natural scene images (Murthy et al., 2013).

To the best of our knowledge, only the existing method (Murthy et al., 2013) can be used for segmentation of natural scene images of Devanagari script. The flaws of existing method (Murthy et al., 2013) indicates a major flaw wherein, many scene images are left partially/ under/ over segmented. There are a large number of such poorly segmented images across the datasets mentioned above and out of those, some of the images are shown in Table 3.11.

Table 3.11 Poorly/ under/ over segmented natural scene images

Image No.	Natural Scene Image	Character segmentation output (Murthy et al., 2013)	Desired Output
1.			च ा ई न ा
2.			प र ▪ प र ा
3.			न ह ी
4.			र व़ त र ा
5.			त ी ख ा म ो ड़

6.			ब न ी
7.			स ा ल ों
8.			ि फ्र ज
9.			म ो ब ा ई ल
10.			श ह र
11.			ि ड ब्ब े
12.			प थ
13.			क ृ प ्य ा
14.			अ म ्ब ा ल ा
15.			क ै म ्प स
16.			व ि श ्व व ि द ्य ा ल य
17.			ज ै स ल म ेर
18.			स ा व ध ा न
19.			श ौ च ा ल य
20.			ल ि ए ग ा ड ि य ाँ
21.			न ई द ि ल ्ल ी

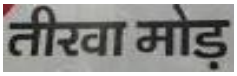
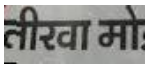
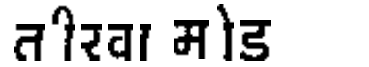
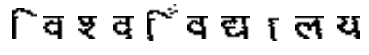
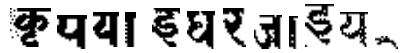
22.			क ृ प य ा इ ध र ज ा इ य े
23.			क ृ प ् य ा म न् द र प र स र
24.			ग ु ड ग ा ँ व
25.			व ि न ् ध ् य ा च ल
26.			स ् ट े श न
27.			प ु ल ि स
28.			क े न ् द ्र ी य
29.			द ल ा ल

It is evident from Table 3.11, that, the existing method (Murthy et al., 2013) completely fails to segment scene images from image no. 1 to 5. Also, from image no. 6 to 25, the said method wrongly/over segment the scene images. In addition to the above, there are a number of images that are left unsegmented (fully/partially) by the existing method (Murthy et al., 2013) and the main identified reasons for poor performance are:

a) Loss of structural information during segmentation into individual characters. The structural data is importantly required for unique recognition of characters. The existing method removes the header line of the word for segmenting into individual characters.

b) Problem in salient region detection and character segmentation methodology. During this stage, single threshold value β was chosen to segment characters.

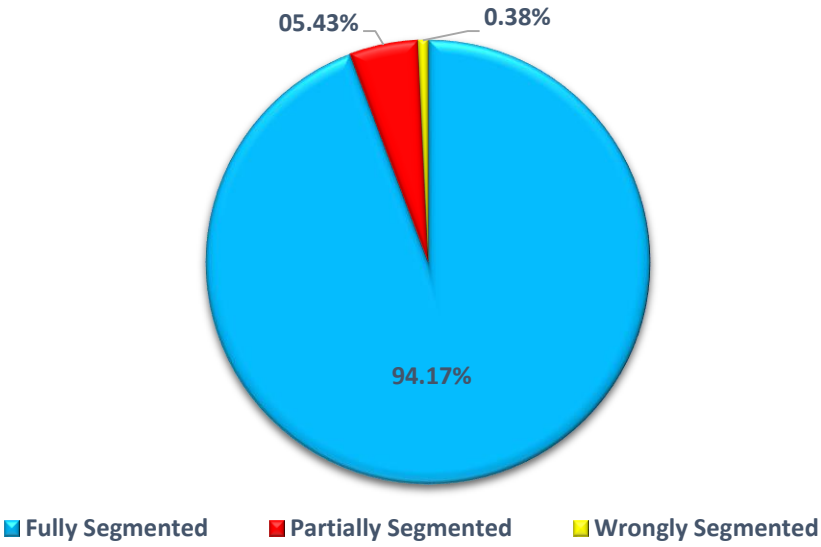
Table 3.12 Result comparison of segmented images on Dataset-I

Image No.	Natural Scene Image	Character segmentation output (Dataset-I)		Desired Output
		Existing Method (Murthy et al., 2013)	Proposed Method "*Pihu*"	
1.				र ख़ त र ा
2.				त ी ख ा म ो ड़
3.				व ि श व व ि द ्य ा ल य
4.				ज ै स ल म ेर
5.				स ा व ध ा न
6.				श ौ च ा ल य
7.				ल ि ए ग ा ड ि य ाँ
8.				न ई द ि ल ्ल ी
9.				क ृ प य ा इ ध र ज ा इ य ं
10.				क ृ प ्य ा म न ्द ि र प र ि स र

The proposed method "***Pihu***" and existing method (Murthy et al., 2013) were simultaneously applied on both the scene image datasets (Dataset-I and Dataset-II). The results obtained by applying the proposed method "***Pihu***" were compared with the results of existing method (Murthy et al., 2013). An example of the comparative analysis of results obtained are shown in Tables 3.12 and 3.14 respectively.

It is evident from the results shown in Table 3.12 that the existing method (Murthy et al., 2013) completely fails to segment images from image no. 1-2 and wrongly segmented the images from image no. 3 to 10, whereas, the proposed method "***Pihu***" successfully segmented all the scene images into individual characters which can be easily classified by the recognition system. Even though, all the image segmentation results from Dataset-I could not be included here due to large volume and the limitation of space, the statistical details of fully, partially and wrongly segmented scene images by using proposed method "***Pihu***" on Dataset-I are shown in Table 3.13.

Table 3.13 Character segmentation results of Dataset-I using proposed method "***Pihu***"



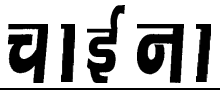
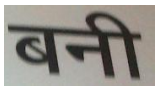
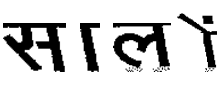
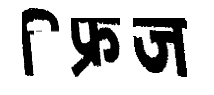
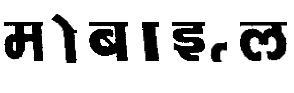
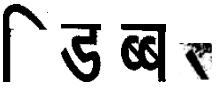
Characters	Total
Correctly Segmented	**485**
Partially Segmented	**28**
Wrongly Segmented	**02**
Overall Accuracy	**94.17%**

Based on the encouraging segmentation results obtained using the proposed method "***Pihu***", the results were compared with the available results of natural scene image

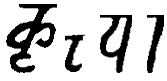
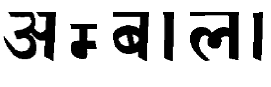
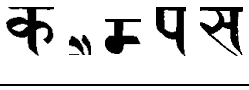
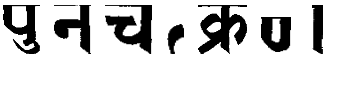
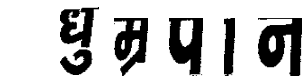
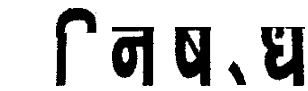
segmentation method (Murthy et al., 2013) for validation and verification. For this, Dataset-II (made available by (Murthy et al., 2013) was used and the one to one image wise result comparison was made between existing method (Murthy et al., 2013) and the proposed method "***Pihu***".

In Table 3.14, it is evident from the character segmentation results comparison that the proposed method "***Pihu***" successfully handled the cases where the existing method failed (Murthy et al., 2013). The segmentation results of the proposed method "***Pihu***" are accurate. In addition, the proposed method "***Pihu***" fully segments the modifier characters like ी, ें, ि and ी, as exemplified by the images 7-9, 11, 12 and13.

Table 3.14 Result comparison of some segmented images on Dataset-II

Image No.	Natural Scene Image	Character segmentation output (Dataset-II)	
		Existing Method (Murthy et al., 2013)	Proposed method "*Pihu*"
1.			
2.			
3.			
4.			
5.			
6.			
7.			

8.			गुडगाँव
9.			परिसर
10.			सेवक
11.			हबीबगंज
12.			पदार्थ
13.			दवाईयां
14.			गुलबर्गा
15.			व्यावसायिक
16.			पुनर्चक्रण
17.			सेंटर
18.			जैव
19.			शाखा
20.			नही
21.			सिटी
22.			पथ

23.			
24.			
25.			
26.			
27.			
28.			
29.			
30.			
31.			

During the segmentation of scene images of the mentioned Datasets using the proposed method "***Pihu***", some of the images were partially segmented due to problems like

a) Unavailability of $CVWPC$ between characters like shown in Table 3.15, (image no. 2)

b) Multi slant images like: shown in Table 3.15, (image no. 1)

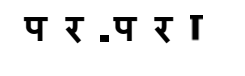
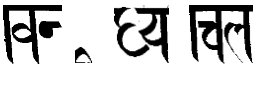
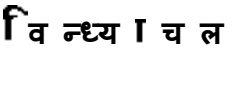
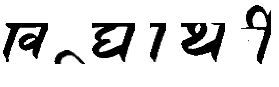
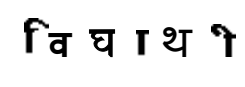
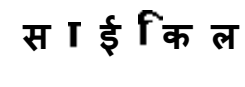
c) Skewed characters like: shown in Table 3.15, (image no. 2 and image no. 3)

d) Background unwanted noisy pixels like: shown in Table 3.15, (image no. 1)

The comparative analysis of few identified cases of partial segmentation using proposed method "***Pihu***" and existing method (Murthy et al., 2013) is shown in Table 3.15. The proposed method "***Pihu***" re-affirms its performance in comparison to existing method (Murthy et al., 2013) by further segmentation of the partially segmented images

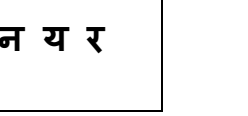
in a second pass whereas the existing method (Murthy et al., 2013) either fails to segment the image or poorly segments the images.

Table 3.15 Results comparison of partially segmented images

Image No.	Scene Image	Partially Segmented Images		Desired Output
		Existing Method (Murthy et al., 2013)	Proposed Method "*Pihu*"	
1.				
2.				
3.				
4.				

Thus, the proposed method "***Pihu***" facilitates better segmentation of handwritten skewed text, complex background and connected characters. There exists a case in which the proposed approach failed to segment the characters. The main reason for poor output was skewed and closely packed handwritten text as shown in Table 3.16.

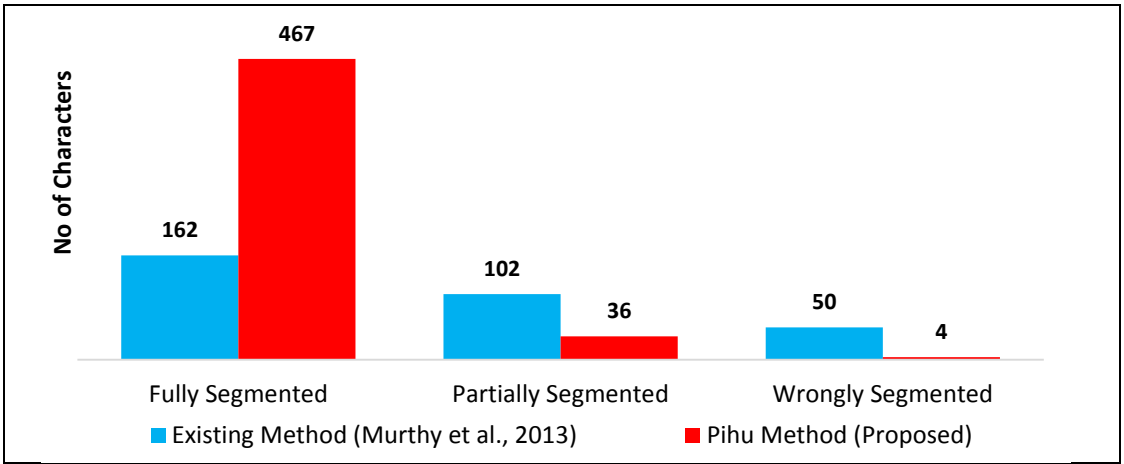
Table 3.16 Unsegmented Image

Image No.	Original Image of word	Output	Desired Output
1.			

A comparative summary of the results of proposed method "***Pihu"*** and existing method (Murthy et al., 2013) is given in Table 3.17. Presently, to the best of our knowledge, there is only one published report (Murthy et al., 2013) on segmentation of natural scene images of Devanagari words. Some additional facts from literature also support the current work,

with reference to execution time between existing (Murthy et al., 2013) and proposed method "***Pihu"***.

Table 3.17 Character segmentation performance comparison of existing method with proposed method "***Pihu***"



Characters	Existing Method (Murthy et al., 2013)	Proposed Pihu Method
Fully Segmented	**162**	**467**
Partially Segmented	**102**	**36**
Wrongly Segmented	**50**	**04**
Overall Accuracy	**55.77%**	**92.11%**

The average time taken by the proposed method "***Pihu***" to segment a natural scene image of size 1169 X 353 is around seconds, which is 56% less than reported by existing method (Murthy et al., 2013). The comparative analysis of hardware used and execution time taken for image segmentation using MATLAB software is shown in Table 3.18.

Table 3.18 Execution time comparison of existing method with the proposed Pihu method

	Existing Method (Murthy et al., 2013)	Proposed Method "*Pihu*"
Hardware used	Workstation: Intel Core 2 Duo E8400@03.00GHZ processor with 02 GB RAM	Laptop: Intel Core i7 3632QM@02.20GHZ processor with 04 GB RAM
Execution Time for segmentation	4.76 sec	2.70 sec

3.5 Conclusion

To summarize this section, a novel shape oriented segmentation algorithm has been developed and tested on machine printed documents of Dogri language including printed image documents of Devanagari script. The accurate shape based character segmentation is the major contribution of the proposed algorithm. It is explored that the structural properties of the character play an important role during segmentation phase. During development of the proposed algorithm, special attention has been given to the shape properties of the character. The said approach not only successfully resolved the identified shortcomings but also reduced the processing time.

The proposed segmentation method was applied on natural scene images for the segmentation of document images of Devanagari script. It was shown that the proposed method "***Pihu***" efficiently segments the characters from natural scene images and resolves many partially and poorly segmented cases of earlier method (Murthy et al., 2013) in addition to enhanced efficiency in segmentation time.

The performance of the proposed segmentation algorithms were validated using dataset of Dogri/ Devanagari based documents from different books, magazines, online and offline newspapers. The offline documents were scanned at 300 DPI resolution in JPG format. The overall data set contained more than 80 scanned machine printed documents with atleast 87000 characters.

Chapter 4
Feature Extraction and Classification

The chapter elucidates the most important stages of character recognition system, known as feature extraction and classification. These stages are equipped with a set of different algorithms. The feature extraction stage extracts features from image text based upon some criteria and the classification stage allot classes to the unknown patterns based upon some training set. The meaningful features of the scanned document image of Dogri language are extracted using proposed shape based segmentation and existing feature extraction techniques.

4.1 Feature Extraction

The feature extraction method should be efficient and robust enough to take care of the shape similarities, poor quality, broken and low resolution scanned text images. Specifically, in the case of segmented connected characters, the boundary area of the character may contain overlapped pixels of the characters. The main factors of structure variation are poor print quality, different font styles and size, noise, skew, etc. The selection of features becomes more challenging due to the huge variation in shapes. Therefore, it is important to select features that are capable of dealing with the structure variations and other factors for large character sets. A number of features have been explored and tested by the researchers on some datasets. However, a room to develop more generic feature selection and extraction algorithms that should be resistant to some factors like

transformations/ contrast, such as scanned image translation, scaling, stretching, rotation, skewing and mirroring (Trier, 1996; Lehal, 1999; Guyon, 2003; Singh, 2011; Ramakrishnan, 2012; Hassan, 2014; Alaei et al., 2016). The selection of number of features and feature type are the main attributes of efficiency measurement.

The feature based techniques can be grouped into two categories, known as spatial and transform domains.

a) In Spatial domain techniques (Shivakumara et al., 2015; Aharrane et al., 2015), the feature vectors are straightforwardly derived from the pixel intensity values of the character i.e., structural and statistical feature vectors can be collected directly from the character image. The geometric properties of the character can be captured using structural features and the statistical features can be obtained from statistically distributed points. The features of some structure provides the information like presence of loops, crossings, curves, straight lines, stroke end point pixels, connected branch points, etc., whereas the statistical features include mathematical calculations of the image regions like those having maximum pixels, moments, etc.

b) In the transform domain techniques (Ramakrishnan and Evgeniy, 2012), initially the text image is reconstructed into another space in order to obtain meaningful features.

In this work, hybrid features were used from both spatial (structural and statistical features) and transform (wavelet features) domains. The grouping of different feature extraction techniques is shown in Figure 4.1.

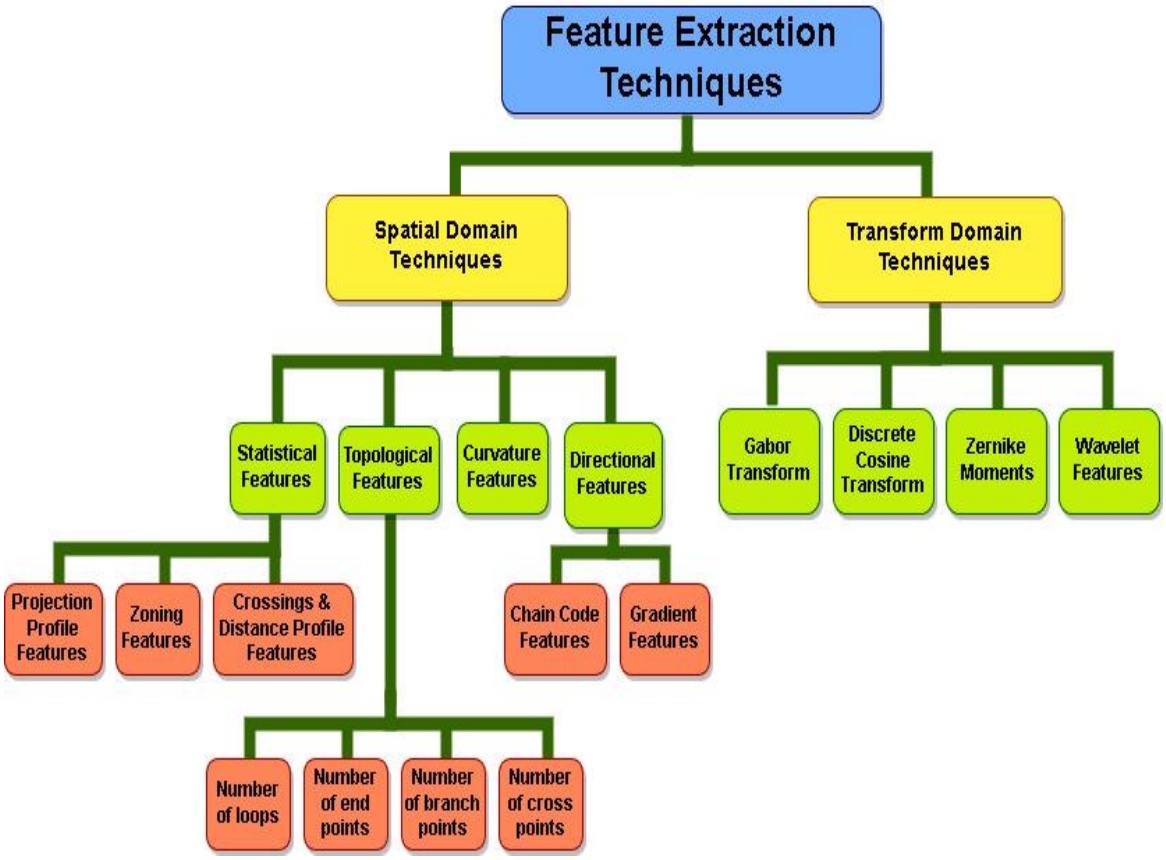


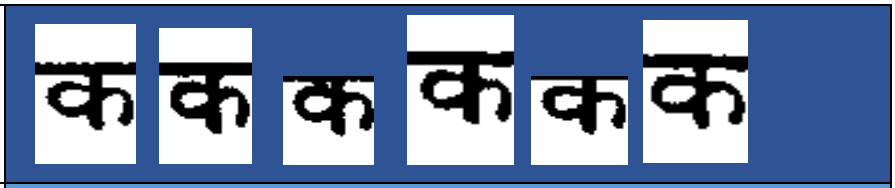
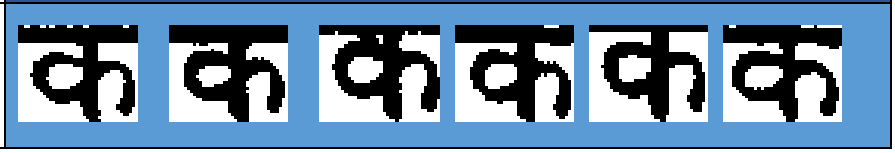
Figure 4.1 Grouping of different feature extraction techniques

These methods can be deployed on grayscale and binary character images. A number of feature extraction techniques have been discussed in the literature review (Chapter 2), many of which are loosely grouped into global and local (adaptive) feature extraction techniques. It has been already pointed out that the global feature techniques are easy to implement, but it does not consider local noise or distortions in the scanned image of the character. Whereas, the topological feature extraction methods analyze the geometry and topology of the character image. The chapter thus describes the different feature extraction techniques that were used and experimented in this research work.

4.1.1 Preparation of Images for Feature Extraction

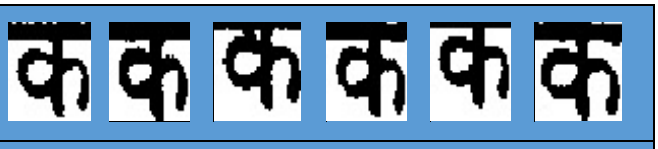
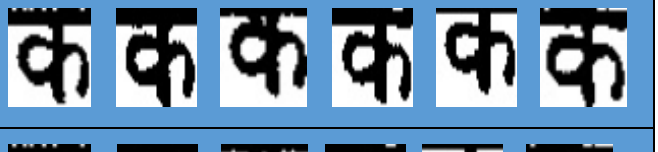
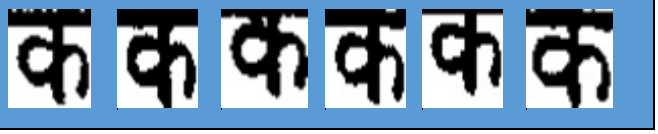
After character segmentation, some of the character images are surrounded by extra white space as shown in Table 4.1(a). This white space can unnecessarily increase the computation time and complexity, so these extra white pixels are removed by analyzing the boundary area of the character image. Table 4.1(b) shows the removal of the said white spaces and the improvised character images thus obtained.

Table 4.1 Original character image (a) with extra white space (b) after removal of extra white space

(a)	
(b)	

Secondly, the segmented character images are of different sizes because of varying size i.e., non-uniform size of character images as shown in Table 4.1(b). This may cause improper feature extraction for the same class of characters. To resolve this issue, for the meaningful and accurate feature extraction, the character images that belong to the same class should also be of the same dimensions. So, to maintain the size uniformity, the character images were scaled (resizing of character image) and the aspect ratio was also maintained. Different scaling techniques were tested on the character images shown in Table 4.1(b) and obtained scaled images as shown in Table 4.2. The implementation details along with suitable examples of scaling techniques are as follows:

Table 4.2 Character image scaled to size 32 x 32 using (a) Nearest Neighbors interpolation (b) Bilinear interpolation (c) Bicubic interpolation

(a)	
(b)	
(c)	

a) Nearest Neighbor Interpolation: This technique (Hong, 2010) is used to increase the image size by replacing all the pixels with multiple pixels that are of the same color. During image enlargement process, the actual details of the image are preserved, but generally with unwanted uneven edges such as, stairway boundary shape in character images as evident in visible in Table 4.2(a). In this approach, the pixel's P location in the enlarged text image is transformed into the original text image, and its distance with the neighboring pixels $N1, N2, N3\ and\ N4$ is computed. The pixel intensity values of P are then applied as the pixel values of its nearest neighbor, Let $(x, y), (x, y+1), (x+1, y)\ and\ (x+1, y+1)$ be the 4 neighboring pixel points of P, with values $N1(x, y), N2(x, y+1), N3(x+1, y)\ and\ N4(x+1, y+1)$ as shown in Figure 4.2. Then, the distance of $P(a, b)\ from\ \{N1(x, y), N2(x, y+1), N3(x+1, y)\ and\ N4(x+1, y+1)\}$ are computed. Finally, values of pixel P are replaced with the pixel value of neighbor i.e. value of $\{N1\ or\ N2\ or\ N3\ or\ N4\}$.

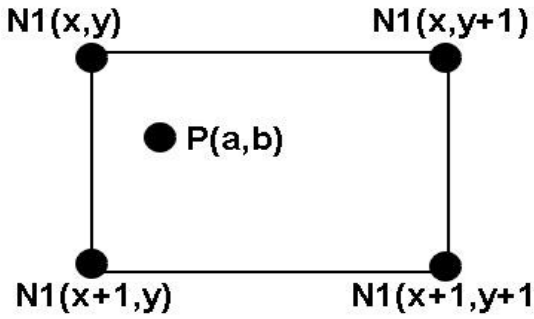


Figure 4.2 Working of Nearest Neighbor interpolation technique

b) Bilinear Interpolation: In this technique (Sen and Ke-jian, 2008), the location of a pixel P that falls in the enlarged text image is transformed to original text image and the impact of 4 neighboring pixel points say P_1, P_2, P_3 and P_4 are computed. The smallest distance to the pixel P gives higher value that shows the efficient performance. The working of the bilinear interpolation technique is illustrated in Figure 4.3.

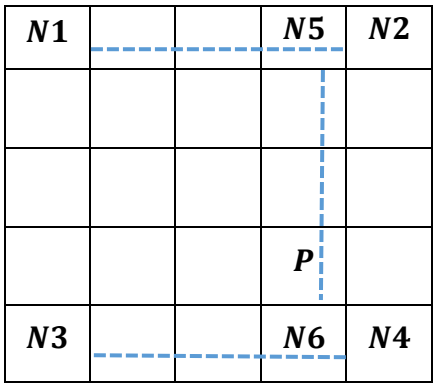


Figure 4.3 Working of bilinear interpolation technique

$N1, N2, N3, N4\ and\ P$ has the coordinates $(x, y), (x, y+1), (x+1, y), (x+1, y+1)$ and (a, b), and the bilinear interpolation technique computes the new enhanced image on these coordinates using following equations:

- The influence of $N1$ and $N2$ is computed, assigned to $N5$ using equation 4.1.

$$f\ (x, y+b) = [f\ (x, y+1) - f\ (x, y)]b\ +\ f\ (x, y) \tag{4.1}$$

- The influence of $N3$ and $N4$ is computed, assigned to $N6$ using equation 4.2.

$$f\ (x+1, y+b)\ =\ [\,f\ (x+1, y+1)\ -\ f\ (x+1, y)]b\ +\ f\ (x+1, y) \qquad (4.2)$$

- Similarly, the influence of $N5$ and $N6$ is computed, assigned to P using equation 4.3.

$$f(x+a, y+b) = (1-a)(1-b)f(x,y) - (1-a)bf(x, y+1) + a(1-b)f(x+1, y) + abf(x+1, y+1) \qquad (4.3)$$

The results obtained after applying bilinear interpolation technique are illustrated in Table 4.2(b).

c) Bicubic Interpolation: This technique is similar to bilinear interpolation, but its time complexity is more. To understand the working of bicubic interpolation technique, any unknown pixel in an enlarged image can be assigned term P, and the nearby area of this pixel is analyzed to check its influence on its sixteen adjacent neighboring pixels as shown in Figure 4.4. Then the distance of P is computed from the 16 neighboring pixels to assign the new value to P.

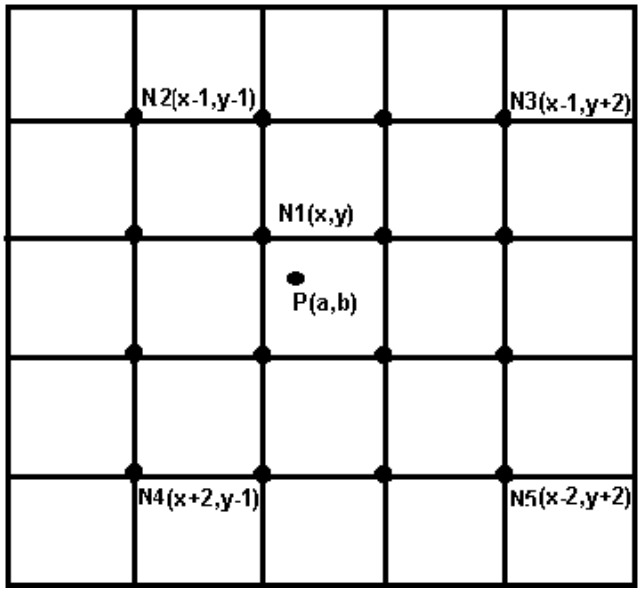


Figure 4.4 Working of bicubic interpolation technique

In comparison to bilinear interpolation, the bicubic interpolation technique makes use of advanced interpolation procedures and is influential with increased number of points. Suppose, there is need to calculate the value of a between two points x and y in the horizontal direction, then it is required to involve four pixel value of $x, y, x-1, y+1$ and obtain a smooth curve through a nonlinear computation as shown in Figure 4.5.

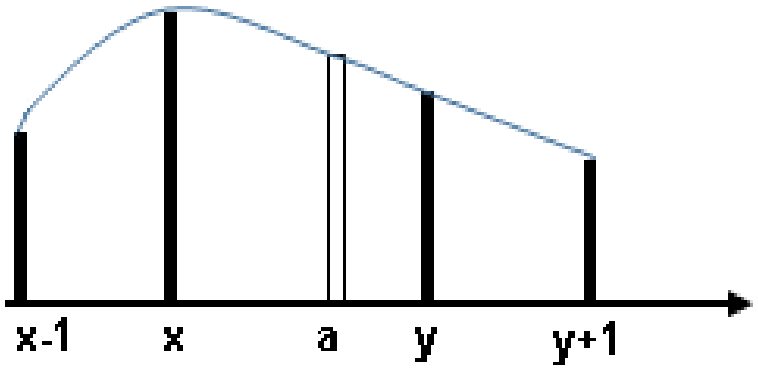


Figure 4.5 Working of nonlinear interpolation technique

The results obtained after applying the bilinear interpolation technique are shown in Table 4.2(c).

After analyzing the results of various interpolation techniques mentioned, it was observed that the bilinear and bicubic interpolation techniques work on the strategy of interpolating pixel color values and produce more accurate results than Nearest Neighbor interpolation. Further, the output obtained using bicubic interpolation technique was better than bilinear interpolation. So, the bicubic interpolation technique has been applied to scale all the character images.

4.1.2 Character Classes

After scaling the character images to uniform size, the task of extracting meaningful and unique features begins. The identical feature selection should help in achieving high

recognition results and the algorithm used for extracting the features should be capable enough to manage different instances of the similar looking characters. A general methodology is to present a character image by a set of features and then, make use of some classifier to classify the feature file into individual classes. Feature extraction is the most vital phase of any character recognition system. The performance of a classifier for recognition of pattern depends heavily on the types and number of feature values used. If huge features are selected from an image, then the overall time complexity will also increase and less number of features results in poor performance. Therefore, for the best performance of a character recognition system, an optimal amount of features needs to be extracted from an image using a statistical classifier, having small training set (Jain and Chandrasekaran, 1982; Jain, 2000; Guyon and Elisseeff, 2003; Hassan et al., 2014).

The Dogri language character set contains huge quantity of characters, modifiers and conjugates that looks similar or differ by only one pixel or stroke. There is large variation in the structure and size of characters as shown in Table 4.3. In addition to this, the situation becomes more complex when the same characters are fused with the different modifiers and conjuncts as shown in Table 4.4. The presence of different modifiers and conjuncts increases the number of character classes. In light of this, it was necessary to extract optimal number of features that were sufficient to uniquely recognize characters. All the distinct characters, modifiers and conjuncts were allotted a unique number which was known as class number. The Dogri language forms around 80 base character classes (including numerals) and 843 compound character classes. So, a total of around 923 character classes including borrowed characters were formulated.

Table 4.3 Basic character set of Dogri language with unique class numbers

Class No	Character	Class No	Character	Class No	Character
1001	क	1028	ळ	1055	ि
1002	ख	1029	व	1056	ी
1003	ग	1030	श	1057	ॅ
1004	घ	1031	ष	1058	े
1005	च	1032	स	1059	ै
1006	छ	1033	ह	1060	ै
1007	ज	1034	क्ष	1061	ॉ
1008	झ	1035	श्र	1062	ो
1009	ञ	1036	ज्ञ	1063	ो
1010	ट	1037	ॐ	1064	ौ
1011	ठ	1038	आ	1065	₹
1012	ड	1039	अ	1066	्
1013	ढ	1040	इ	1067	ः
1014	ण	1041	उ	1068	॰
1015	त	1042	ऊ	1069	ु
1016	थ	1043	ऋ	1070	ू
1017	द	1044	ऌ	1071	0
1018	ध	1045	ए	1072	1
1019	न	1046	ऍ	1073	2
1020	प	1047	ॠ	1074	3
1021	फ	1048	ा	1075	4
1022	ब	1049	ऽ	1076	5
1023	भ	1050	।	1077	6
1024	म	1051	॥	1078	7
1025	य	1052	॰	1079	8
1026	र	1053	‘	1080	9
1027	ल	1054	’		

Table 4.4 Conjugate character set of Dogri language with unique class numbers

Class No	Character	Class No	Character	Class No	Character	Class No	Character
1081	क्क	1106	क्य	1131	ख्त	1156	घ्च
1082	क्ख	1107	क्र	1132	ख्थ	1157	घ्छ
1083	क्ग	1108	क्ल	1133	ख्द	1158	घ्ज
1084	क्घ	1109	क्व	1134	ख्ध	1159	घ्झ
1085	क्ङ	1110	क्श	1135	ख्न	1160	घ्ञ
1086	क्च	1111	क्स	1136	ख्प	1161	घ्ट
1087	क्छ	1112	क्ह	1137	ख्फ	1162	घ्ठ
1088	क्ज	1113	क्ळ	1138	ख्ब	1163	घ्ड
1089	क्झ	1114	क्क्ष	1139	ख्भ	1164	घ्ढ
1090	क्ञ	1115	क्ज्ञ	1140	ख्म	1165	घ्ण
1091	क्ट	1116	ख्क	1141	ख्य	1166	घ्त
1092	क्ठ	1117	ख्ख	1142	ख्ल	1167	घ्थ
1093	क्ड	1118	ख्ग	1143	ख्व	1168	घ्द
1094	क्ढ	1119	ख्घ	1144	ख्श	1169	घ्ध
1095	क्ण	1120	ख्ङ	1145	ख्ष	1170	घ्न
1096	क्त	1121	ख्च	1146	ख्स	1171	घ्प
1097	क्थ	1122	ख्छ	1147	ख्ह	1172	घ्फ
1098	क्द	1123	ख्ज	1148	ख्ळ	1173	घ्ब
1099	क्ध	1124	ख्झ	1149	ख्क्ष	1174	घ्भ
1100	क्न	1125	ख्ञ	1150	ख्ज्ञ	1175	घ्म
1101	क्प	1126	ख्ट	1151	घ्क	1176	घ्य
1102	क्फ	1127	ख्ठ	1152	घ्ख	1177	घ्र
1103	क्ब	1128	ख्ड	1153	घ्ग	1178	घ्ल
1104	क्भ	1129	ख्ढ	1154	घ्घ	1179	घ्व
1105	क्म	1130	ख्ण	1155	घ्ङ	1180	घ्श

Class No	Character	Class No	Character	Class No	Character	Class No	Character
1181	घ्ष	1206	इ्न	1231	च्झ	1256	च्ळ
1182	घ्स	1207	इ्प	1232	च्ञ	1257	च्क्ष
1183	घ्ह	1208	इ्फ	1233	च्ट	1258	च्ज्ञ
1184	घ्ळ	1209	इ्ब	1234	च्ठ	1259	छ्ख
1185	घ्क्ष	1210	इ्भ	1235	च्ड	1260	छ्घ
1186	घ्ज्ञ	1211	इ्म	1236	च्ढ	1261	छ्ङ
1187	इ्क	1212	इ्य	1237	च्ण	1262	छ्ज
1188	इ्ख	1213	इ्र	1238	च्त	1263	छ्झ
1189	इ्ग	1214	इ्ल	1239	च्थ	1264	छ्ञ
1190	इ्घ	1215	इ्व	1240	च्द	1265	छ्ड
1191	इ्ङ	1216	इ्श	1241	च्ध	1266	छ्द
1192	इ्च	1217	इ्ष	1242	च्न	1267	छ्ध
1193	इ्छ	1218	इ्स	1243	च्प	1268	छ्र
1194	इ्ज	1219	इ्ह	1244	च्फ	1269	छ्स
1195	इ्झ	1220	इ्ळ	1245	च्ब	1270	छ्ळ
1196	इ्ञ	1221	इ्क्ष	1246	च्भ	1271	छ्क्ष
1197	इ्ट	1222	इ्ज्ञ	1247	च्म	1272	ज्क
1198	इ्ठ	1223	च्क	1248	च्य	1273	ज्ख
1199	इ्ड	1224	च्ख	1249	च्र	1274	ज्ग
1200	इ्ढ	1225	च्ग	1250	च्ल	1275	ज्घ
1201	इ्ण	1226	च्घ	1251	च्व	1276	ज्ङ
1202	इ्त	1227	च्ङ	1252	च्श	1277	ज्च
1203	इ्थ	1228	च्च	1253	च्ष	1278	ज्छ
1204	इ्द	1229	च्छ	1254	च्स	1279	ज्ज
1205	इ्ध	1230	च्ज	1255	च्ह	1280	ज्झ

Class No	Character	Class No	Character	Class No	Character	Class No	Character
1281	ज्ट	1306	ज्ञ	1331	ज्म	1356	ञ्ण
1282	ज्ठ	1307	ज्क	1332	ज्य	1357	ञ्त
1283	ज्ड	1308	ज्ख	1333	ज्ल	1358	ञ्थ
1284	ज्ढ	1309	ज्ग	1334	ज्व	1359	ञ्द
1285	ज्ण	1310	ज्घ	1335	ज्श	1360	ञ्ध
1286	ज्त	1311	ज्ङ	1336	ज्ष	1361	ञ्न
1287	ज्थ	1312	ज्च	1337	ज्स	1362	ञ्प
1288	ज्द	1313	ज्छ	1338	ज्ह	1363	ञ्फ
1289	ज्ध	1314	ज्ज	1339	ज्ळ	1364	ञ्ब
1290	ज्न	1315	ज्झ	1340	ज्क्ष	1365	ञ्भ
1291	ज्प	1316	ज्ञ	1341	ज्ज्ञ	1366	ञ्म
1292	ज्फ	1317	ज्ट	1342	ञ्क	1367	ञ्य
1293	ज्ब	1318	ज्ठ	1343	ञ्ख	1368	ञ्र
1294	ज्भ	1319	ज्ड	1344	ञ्ग	1369	ञ्ल
1295	ज्म	1320	ज्ढ	1345	ञ्घ	1370	ञ्व
1296	ज्य	1321	ज्ण	1346	ञ्ङ	1371	ञ्श
1297	ज्र	1322	ज्त	1347	ञ्च	1372	ञ्ष
1298	ज्ल	1323	ज्थ	1348	ञ्छ	1373	ञ्स
1299	ज्व	1324	ज्द	1349	ञ्ज	1374	ञ्ह
1300	ज्श	1325	ज्ध	1350	ञ्झ	1375	ञ्ळ
1301	ज्ष	1326	ज्न	1351	ञ्ञ	1376	ञ्क्ष
1302	ज्स	1327	ज्प	1352	ञ्ट	1377	ञ्ज्ञ
1303	ज्ह	1328	ज्फ	1353	ञ्ठ	1378	ट्ट
1304	ज्ळ	1329	ज्ब	1354	ञ्ड	1379	ट्ठ
1305	ज्क्ष	1330	ज्भ	1355	ञ्ढ	1380	ट्र

Class No	Character	Class No	Character	Class No	Character	Class No	Character
1381	ठ्ठ	1406	त्ड	1431	थ्ग	1456	थ्ल
1382	ठ्र	1407	त्ढ	1432	थ्घ	1457	थ्व
1383	ड्र	1408	त्ण	1433	थ्ङ	1458	थ्श
1384	ढ्ङ	1409	त्त	1434	थ्च	1459	थ्ष
1385	ढ्ज	1410	त्थ	1435	थ्छ	1460	थ्स
1386	ढ्झ	1411	त्द	1436	थ्ज	1461	थ्ह
1387	ढ्ञ	1412	त्ध	1437	थ्झ	1462	थ्ळ
1388	ढ्ड	1413	त्न	1438	थ्ञ	1463	थ्क्ष
1389	ढ्ढ	1414	त्प	1439	थ्ट	1464	थ्ज्ञ
1390	ढ्द	1415	त्फ	1440	थ्ठ	1465	द्ध
1391	ढ्र	1416	त्ब	1441	थ्ड	1466	द्भ
1392	ढ्ळ	1417	त्भ	1442	थ्ढ	1467	द्म
1393	ढ्क्ष	1418	त्म	1443	थ्ण	1468	द्र
1394	त्क	1419	त्य	1444	थ्त	1469	ध्क
1395	त्ख	1420	त्ल	1445	थ्थ	1470	ध्ख
1396	त्ग	1421	त्व	1446	थ्द	1471	ध्ग
1397	त्घ	1422	त्श	1447	थ्ध	1472	ध्घ
1398	त्ङ	1423	त्ष	1448	थ्न	1473	ध्ङ
1399	त्च	1424	त्स	1449	थ्प	1474	ध्च
1400	त्छ	1425	त्ह	1450	थ्फ	1475	ध्छ
1401	त्ज	1426	त्ळ	1451	थ्ब	1476	ध्ज
1402	त्झ	1427	त्क्ष	1452	थ्भ	1477	ध्झ
1403	त्ञ	1428	त्ज्ञ	1453	थ्म	1478	ध्ञ
1404	त्ट	1429	थ्क	1454	थ्य	1479	ध्ट
1405	त्ठ	1430	थ्ख	1455	थ्र	1480	ध्ठ

Class No	Character	Class No	Character	Class No	Character	Class No	Character
1481	ध्ड	1506	न्ख	1531	न्र	1556	प्त
1482	ध्ढ	1507	न्ग	1532	न्ल	1557	प्थ
1483	ध्ण	1508	न्घ	1533	न्व	1558	प्द
1484	ध्त	1509	न्ङ	1534	न्श	1559	प्ध
1485	ध्थ	1510	न्च	1535	न्ष	1560	प्न
1486	ध्द	1511	न्छ	1536	न्स	1561	प्प
1487	ध्ध	1512	न्ज	1537	न्ह	1562	प्फ
1488	ध्न	1513	न्झ	1538	न्ळ	1563	प्ब
1489	ध्प	1514	न्ञ	1539	न्क्ष	1564	प्भ
1490	ध्फ	1515	न्ट	1540	न्ज्ञ	1565	प्म
1491	ध्ब	1516	न्ठ	1541	प्क	1566	प्य
1492	ध्भ	1517	न्ड	1542	प्ख	1567	प्र
1493	ध्म	1518	न्ढ	1543	प्ग	1568	प्ल
1494	ध्य	1519	न्ण	1544	प्घ	1569	प्व
1495	ध्र	1520	न्त	1545	प्ङ	1570	प्श
1496	ध्ल	1521	न्थ	1546	प्च	1571	प्ष
1497	ध्व	1522	न्द	1547	प्छ	1572	प्स
1498	ध्श	1523	न्ध	1548	प्ज	1573	प्ह
1499	ध्ष	1524	न्न	1549	प्झ	1574	प्ळ
1500	ध्स	1525	न्प	1550	प्ञ	1575	प्क्ष
1501	ध्ह	1526	न्फ	1551	प्ट	1576	प्ज्ञ
1502	ध्ळ	1527	न्ब	1552	प्ठ	1577	फ्क
1503	ध्क्ष	1528	न्भ	1553	प्ड	1578	फ्ख
1504	ध्ज्ञ	1529	न्म	1554	प्ढ	1579	फ्घ
1505	न्क	1530	न्य	1555	प्ण	1580	फ्ङ

Class No	Character	Class No	Character	Class No	Character	Class No	Character
1581	फ्च	1606	ब्ङ	1631	ब्श	1656	भ्ध
1582	फ्छ	1607	ब्च	1632	ब्ष	1657	भ्न
1583	फ्ज	1608	ब्छ	1633	ब्स	1658	भ्प
1584	फ्झ	1609	ब्ज	1634	ब्ह	1659	भ्फ
1585	फ्ञ	1610	ब्झ	1635	ब्ळ	1660	भ्ब
1586	फ्ड	1611	ब्ञ	1636	ब्क्ष	1661	भ्भ
1587	फ्ढ	1612	ब्ट	1637	ब्ज्ञ	1662	भ्म
1588	फ्ण	1613	ब्ठ	1638	भ्क	1663	भ्य
1589	फ्थ	1614	ब्ड	1639	भ्ख	1664	भ्र
1590	फ्द	1615	ब्ढ	1640	भ्ग	1665	भ्ल
1591	फ्ध	1616	ब्ण	1641	भ्घ	1666	भ्व
1592	फ्फ	1617	ब्त	1642	भ्ङ	1667	भ्श
1593	फ्र	1618	ब्थ	1643	भ्च	1668	भ्ष
1594	फ्ल	1619	ब्द	1644	भ्छ	1669	भ्स
1595	फ्व	1620	ब्ध	1645	भ्ज	1670	भ्ह
1596	फ्श	1621	ब्न	1646	भ्झ	1671	भ्ळ
1597	फ्ष	1622	ब्प	1647	भ्ञ	1672	भ्क्ष
1598	फ्स	1623	ब्फ	1648	भ्ट	1673	भ्ज्ञ
1599	फ्ह	1624	ब्ब	1649	भ्ठ	1674	म्क
1600	फ्ळ	1625	ब्भ	1650	भ्ड	1675	म्ख
1601	फ्क्ष	1626	ब्म	1651	भ्ढ	1676	म्ग
1602	ब्क	1627	ब्य	1652	भ्ण	1677	म्घ
1603	ब्ख	1628	ब्र	1653	भ्त	1678	म्ङ
1604	ब्ग	1629	ब्ल	1654	भ्थ	1679	म्च
1605	ब्घ	1630	ब्व	1655	भ्द	1680	म्छ

Class No	Character	Class No	Character	Class No	Character	Class No	Character
1681	म्ज	1706	म्ह	1731	य्फ	1756	ल्ट
1682	म्झ	1707	म्ळ	1732	य्ब	1757	ल्ठ
1683	म्ञ	1708	म्क्ष	1733	य्भ	1758	ल्ड
1684	म्ट	1709	म्ज्ञ	1734	य्म	1759	ल्ढ
1685	म्ठ	1710	य्क	1735	य्य	1760	ल्ण
1686	म्ड	1711	य्ख	1736	य्र	1761	ल्त
1687	म्ढ	1712	य्ग	1737	य्ल	1762	ल्थ
1688	म्ण	1713	य्घ	1738	य्व	1763	ल्द
1689	म्त	1714	य्ङ	1739	य्श	1764	ल्ध
1690	म्थ	1715	य्च	1740	य्ष	1765	ल्न
1691	म्द	1716	य्छ	1741	य्स	1766	ल्प
1692	म्ध	1717	य्ज	1742	य्ह	1767	ल्फ
1693	म्न	1718	य्झ	1743	य्ळ	1768	ल्ब
1694	म्प	1719	य्ञ	1744	य्क्ष	1769	ल्भ
1695	म्फ	1720	य्ट	1745	य्ज्ञ	1770	ल्म
1696	म्ब	1721	य्ठ	1746	ल्क	1771	ल्य
1697	म्भ	1722	य्ड	1747	ल्ख	1772	ल्र
1698	म्म	1723	य्ढ	1748	ल्ग	1773	ल्ल
1699	म्य	1724	य्ण	1749	ल्घ	1774	ल्व
1700	म्र	1725	य्त	1750	ल्ङ	1775	ल्श
1701	म्ल	1726	य्थ	1751	ल्च	1776	ल्ष
1702	म्व	1727	य्द	1752	ल्छ	1777	ल्स
1703	म्श	1728	य्ध	1753	ल्ज	1778	ल्ह
1704	म्ष	1729	य्न	1754	ल्झ	1779	ल्ळ
1705	म्स	1730	य्प	1755	ल्ञ	1780	ल्क्ष

Class No	Character	Class No	Character	Class No	Character	Class No	Character
1781	ल्ज्ञ	1806	व्म	1831	श्ढ	1856	ष्घ
1782	व्क	1807	व्य	1832	श्ण	1857	ष्ङ
1783	व्ख	1808	व्र	1833	श्त	1858	ष्च
1784	व्ग	1809	व्ल	1834	श्थ	1859	ष्छ
1785	व्घ	1810	व्व	1835	श्द	1860	ष्ज
1786	व्ङ	1811	व्श	1836	श्ध	1861	ष्झ
1787	व्च	1812	व्ष	1837	श्न	1862	ष्ञ
1788	व्छ	1813	व्स	1838	श्प	1863	ष्ट
1789	व्ज	1814	व्ह	1839	श्फ	1864	ष्ठ
1790	व्झ	1815	व्ळ	1840	श्ब	1865	ष्ड
1791	व्ञ	1816	व्क्ष	1841	श्भ	1866	ष्ढ
1792	व्ट	1817	व्ज्ञ	1842	श्म	1867	ष्ण
1793	व्ठ	1818	श्क	1843	श्य	1868	ष्त
1794	व्ड	1819	श्ख	1844	श्ल	1869	ष्थ
1795	व्ढ	1820	श्ग	1845	श्व	1870	ष्द
1796	व्ण	1821	श्घ	1846	श्श	1871	ष्ध
1797	व्त	1822	श्ङ	1847	श्ष	1872	ष्न
1798	व्थ	1823	श्च	1848	श्स	1873	ष्प
1799	व्द	1824	श्छ	1849	श्ह	1874	ष्फ
1800	व्ध	1825	श्ज	1850	श्ळ	1875	ष्ब
1801	व्न	1826	श्झ	1851	श्क्ष	1876	ष्भ
1802	व्प	1827	श्ञ	1852	श्ज्ञ	1877	ष्म
1803	व्फ	1828	श्ट	1853	ष्क	1878	ष्य
1804	व्ब	1829	श्ठ	1854	ष्ख	1879	ष्र
1805	व्भ	1830	श्ड	1855	ष्ग	1880	ष्ल

Class No	Character	Class No	Character
1881	ष्व	1903	स्ण
1882	ष्श	1904	स्त
1883	ष्ष	1905	स्थ
1884	ष्स	1906	स्द
1885	ष्ह	1907	स्ध
1886	ष्ळ	1908	स्न
1887	ष्क्ष	1909	स्प
1888	ष्ज्ञ	1910	स्फ
1889	स्क	1911	स्ब
1890	स्ख	1912	स्भ
1891	स्ग	1913	स्म
1892	स्घ	1914	स्य
1893	स्ङ	1915	स्र
1894	स्च	1916	स्ल
1895	स्छ	1917	स्व
1896	स्ज	1918	स्श
1897	स्झ	1919	स्ष
1898	स्ञ	1920	स्स
1899	स्ट	1921	स्ह
1900	स्ठ	1922	स्ळ
1901	स्ड	1923	स्क्ष
1902	स्ढ		

After a detailed study and analysis of different feature extraction techniques, it was found that the techniques such as Discrete Cosine Transformation (DCT), Zernike moments and Gradient were suitable for extracting features from Dogri language text images. The above said techniques were used in this work and are explained in the following subsections.

4.1.3 Discrete Cosine Transformation (DCT)

DCT function separates the image into frequencies with large variance in lesser time, i.e., it encodes significant details of the image in a few coefficients which further serves as features for the image. Therefore, two-dimensional DCT can be computed using the one-dimensional DCT horizontally (row-wise) and then vertically (column-wise) across the image. DCT coefficients $D(x,y)$ of an image $I(i,j)$ are obtained using equation 4.4.

$$D(x,y) = C(x)C(y)\sum_{i=0}^{M-1}\sum_{j=0}^{N-1} P(i,j)cos\left[\frac{2i+1}{2M}x\pi\right] cos\left[\frac{2j+1}{2N}y\pi\right] \quad (4.4)$$

where $C(x) = \begin{cases} \sqrt{\frac{1}{M}}, & x = 0 \\ \sqrt{\frac{2}{M}}, & x > 0 \end{cases}$ $C(y) = \begin{cases} \sqrt{\frac{1}{N}}, & y = 0 \\ \sqrt{\frac{2}{N}}, & y > 0 \end{cases}$

$D(x,y)$ represents the DCT coefficient corresponding to the image pixel $I(i,j)$. All the input images are scaled to size $32X32$, which contains a total of *1024* (transform coefficients) frequency components. The top left side of the DCT matrix shows high frequency component i.e., maximum information area. The image $I(i,j)$ is partitioned into $8X8$ matrix. For the experimentation work *200* features have been chosen from a total of *1024* features in a zig-zag manner as shown in Figure 4.6. This value is considered optimum, as by further increase in number of features there was a negligible improvement. However, the processing time and size of feature file increased.

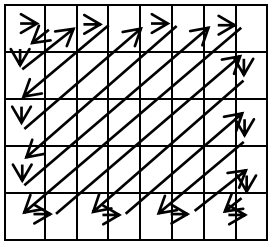


Figure 4.6 Zig-zag collection of DCT coefficients

4.1.4 Gradient Features

In this process, the input image is scaled to the size of $63X63$ and the gradient of image IMG at location (r, c) is extracted to obtain the distinctive information (Wang et al., 2005; Liu et al., 2005). The gradient capacity and direction has been calculated from the gradient vector $[G_x, G_y]^T$ as:

Gradient Magnitude is computed as $G(r,c) = \sqrt{(G_x(r,c))^2 + (G_y(r,c))^2}$ (4.5)

Gradient Direction is computed as $\theta(r,c) = \tan^{-1}\frac{G_y(r,c)}{G_x(r,c)}$ (4.6)

In an image of size say R x C, each of the pixel neighborhood is combined with Sobel templates to ascertain the G_x, G_y, x and y parameters. The mathematical representation is given by the equations (4.7) and (4.8):

$$G_x(r,c) = IMG(r-1,c+1) + 2*(r,c+1) + IMG(r+1,c+1) - IMG(r-1,c-1) - 2*IMG(r,c-1) - IMG(r+1,c-1) \quad (4.7)$$

$$G_y(r,c) = IMG(r-1,c-1) + 2*(r-1,c) + IMG(r-1,c+1) - IMG(r+1,c-1) - 2*IMG(r+1,c) - IMG(r+1,c+1) \quad (4.8)$$

where G_x represents horizontal gradient and G_y represents vertical gradient directions, (r, c) represents the bounds over the rows (R) and columns (C).

Then the gradient vector $G(x, y)$ is computed at each pixel location by means of the Sobel horizontal (x) and vertical (y) operators in Figure 4.7.

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

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

Figure 4.7 Sobel masks for gradient (a) Horizontal and (b) Vertical operators

The Figure 4.8 shows gradient magnitude and direction of the character  using Sobel masks of Figure 4.7.

(a) (b) (c)

Figure 4.8 Gradient (a) Magnitude, (b) Direction and (c) Directional gradients G_x and G_y

When the gradient vector for all the pixels are collected, then the gradient image is divided into 4 equal sized orientation planes or eight chain code direction planes as shown in Figure 4.9.

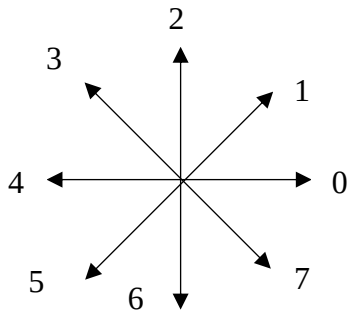


Figure 4.9 Eight directions of chain-codes

After obtaining the gradient vector of all the pixels, the gradient image is decomposed into eight chain code directions and is partitioned into 81 ($9X9$) blocks. Then, $5X5$ Gaussian filter is applied for changing the spatial resolution from $9X9$ to $5X5$ by the sub-sampling of every two horizontal and vertical blocks. This helps in generating a feature vector data of size *200*.

4.1.5 Zernike Moments

The Zernike moments represents the projections of the scanned input text image on the range spanned by the orthogonal Z-functions defined by:

$$A_{nm}(x,y) = R_{nm}(x,y)\exp\left(j\ m\ tan^{-1}\left(\frac{y}{x}\right)\right) \tag{4.9}$$

where $j = \sqrt{-1}\ n \geq 0, |m| \leq n, n - |m|\ even\ and$

$$R_{nm}(x,y) = \sum_{s=0}^{\frac{n-|m|}{2}} \frac{(-1)^s(x^2+y^2)^{\frac{n-2s}{2}}(n-s)!}{s!\left(\frac{n+|m|}{2}-s\right)!\left(\frac{n-|m|}{2}-s\right)!} \tag{4.10}$$

In other words, Zernike moments represents complex numbers using which an input text image is mapped on a collection of 2D Zernike polynomials. In this method, the magnitude is deployed as a rotation invariant feature to represent a character image. Further, the image is divided into equal sized zones and then extracted the features from each zone. Then *200* features per sample have been extracted in a zig-zag manner.

4.2 Classification

Classifier performs unique identification of the patterns on the basis of feature vectors. It ascertains the area of feature space in which an unknown character can be mapped. It uses a dataset for building the training model and then test unknown feature set of character images against the training model. In this chapter, various classifiers are explained which are used for the recognition of Dogri language characters in the present work. The

classifiers take input as a text file containing feature data (with class number) extracted from the character images in feature extraction stage.

There are a number of classification methods based on learning and are globally used for character recognition from a long time. The classification methods can be grouped into categories like statistical methods (Pal and Chaudhuri, 2004; Jayadevan et al., 2011; Lehal and Singh, 2011; Kumar et al., 2014; Lehal and Singh, 2014; Aharrane et al., 2015; Kumar et al., 2015), kernel methods (Draper et al., 1994; Frias-Martinez et al., 2006; Kale et al., 2013; Al-Boeridi, 2015; Rana and Lehal, 2015; Verma and Sharma, 2015;Verma and Sharma, 2016), artificial neural networks (Al-Boeridi et al., 2015) and methods based upon hybrid techniques (Lehal and Singh, 1999; Bhattacharya et al., 2002; Kluzner et al., 2011).

A number of classifiers have been studied and analyzed that can be used for the recognition of Dogri language documents. After detailed analysis, the present work has made use of techniques like Support Vector Machine (SVM), k-NN (k-Nearest Neighbors) and Multilayer Perceptron neural network (MPNN) for the classification of Dogri language characters. These techniques give efficient and accurate character recognition results, when fused with suitable feature extraction techniques discussed in Section 4.1.

4.2.1 k-Nearest Neighbors (k-NN): It is a non-parametric algorithm which takes input as training samples, and is commonly used for the classification and regression (Casey and Lecolinet, 1996; Kumar et al., 2011). The distance to every training sample from test sample is computed. Out of those samples, k closest training samples are cashed, where value of k is taken from (3,5,7). A label is then compared with the sample, which is most

common among cashed samples. Euclidean distance metric is used to decide common similarity between feature vectors using equation 4.11.

$$D(x,y) = \sqrt{\sum_{i=1}^{f}(x_{i-}y_i)^2} \tag{4.11}$$

where $D(x,y)$ is distance between training and testing samples, f denotes the no of features and $x,y \in R^f$

The instance oriented classifier recognizes the given input just by matching it with the training data (already classified). For any input pattern, the distance between current input and the different samples in the training data is computed, out of which, the sample having lesser distance from the input pattern is selected. An example to illustrate the working of the k-NN algorithm is shown in Figure 4.10.

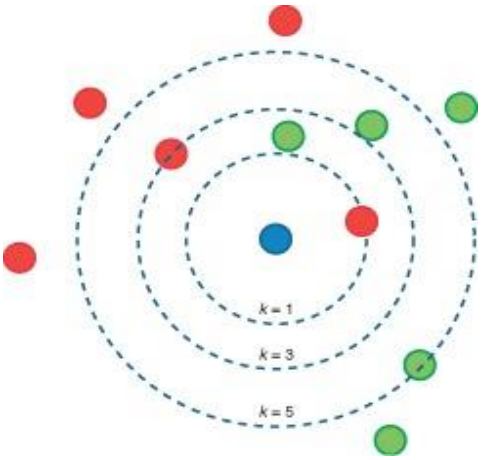


Figure 4.10 Example of k-NN classification algorithm

The combination of k-NN classifier from weka tool kit with various feature extraction techniques has been used for character recognition. The character recognition results, obtained with the testing option of cross validation (fold values $F = 3,5,7$) and percentage data split (training= 75% and testing=25%) are shown in Table 4.5.

Table 4.5 Recognition accuracy comparison of characters with and without header line, using k-NN classifier

Character Level Recognition Accuracy using k-NN (Dataset Size: 87000 Characters)									
Feature extraction technique	**Number of feature's**	**Dataset-I (With Shirorekha)**				**Dataset-II (Without Shirorekha)**			
		***F* = 3**	***F* = 5**	***F* = 7**	**75:25 % Data Split**	***F* = 3**	***F* = 5**	***F* = 7**	**75:25 % Data Split**
DCT	200	96.84	97.12	97.16	**97.36**	89.91	92.00	93.53	**94.57**
Zernike	200	95.58	95.95	96.10	**96.15**	86.44	88.12	91.41	**94.24**
Gradient	200	96.82	97.16	97.57	**98.10**	91.57	92.60	93.10	**94.42**

The graphical representation of character level results using k-NN are shown in Figure 4.11.

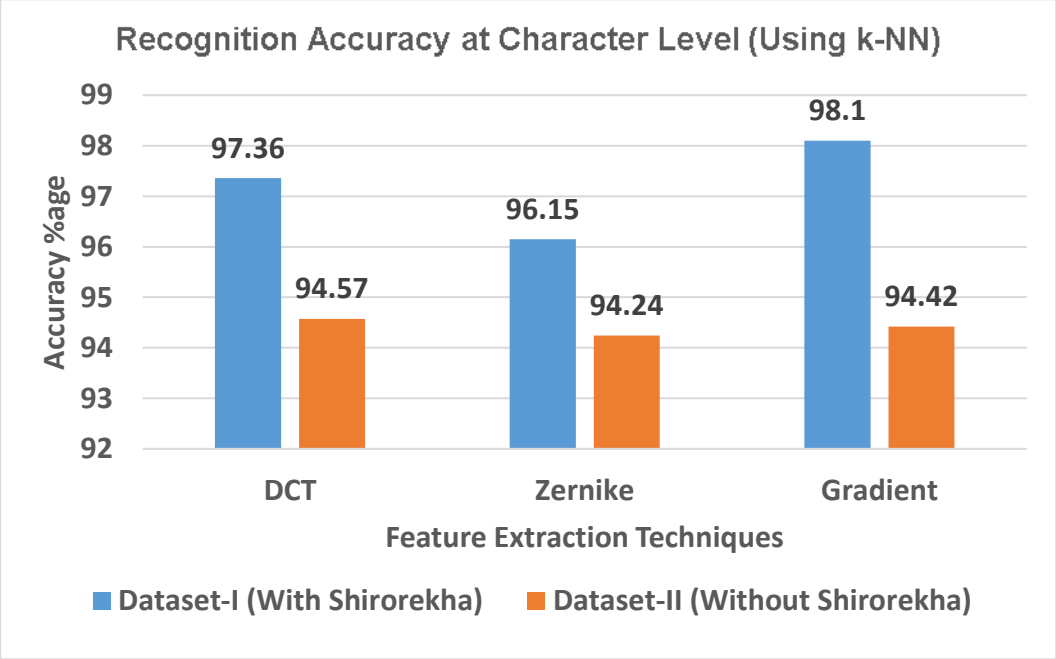


Figure 4.11 Graphical representation of results of k-NN classifier with different feature extraction methods

4.2.2 Multilayer Perceptron Neural Network (MPNN): It consists of multiple layers of nodes in a directed graph, with each layer fully connected to the next layer (Altuwaijri and Bayoumi, 1994; Gardner and Dorling, 1998; Haykin and Network, 2004; Frias-Martinez et al., 2006). Each node is a neuron (known as perceptron) with a nonlinear activation

function, except for the input nodes. In other words, a perceptron is a model that represents a single neuron which was a forerunner to larger neural networks. It utilizes a supervised learning technique trained by backpropagation algorithm. Basically, MPNN works in two unique ways, the first is the recall process in which the network input layer is provided with the training patterns and the output of which is recalled at the resulting network layer. Secondly, in the learning process, the synaptic weights are managed in the network. The working example of MPNN is shown in Figure 4.12.

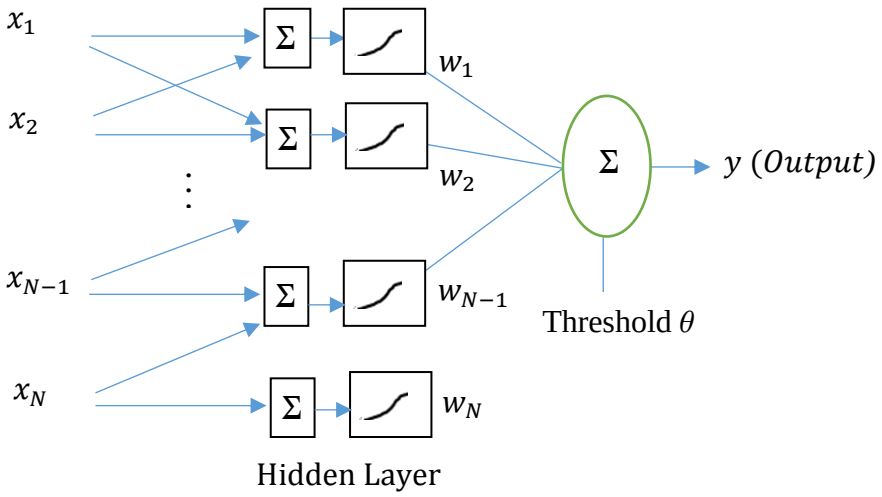


Figure 4.12 Example of Multilayer Perceptron neural network

$$u = \sum_{i=1}^{N} w_i\, x_i \tag{4.12}$$

$$y = f(u - \theta) \tag{4.13}$$

where u is the activation potential, y is the output signal,x_i are the inputs, w_i are vectors of real valued weight, θ is the threshold, N is the total number of inputs.

In this study, combination of multilayer perceptron classifier have been used from weka tool kit with different feature extraction method for the recognition of the character.

Results obtained with testing option of cross validation (fold values $F = 3,5,7$) and also, with percentage data split (training= 75% and testing= 25%) are shown in Table 4.6.

Table 4.6 Recognition accuracy comparison of characters with and without header line, using MPNN classifier

Character Level Recognition Accuracy using MPNN (Dataset Size: 87000 Characters)									
Feature extraction technique	**Number of feature's**	**Dataset-I (With Shirorekha)**				**Dataset-II (Without Shirorekha)**			
		***F* = 3**	***F* = 5**	***F* = 7**	**75:25 % Data Split**	***F* = 3**	***F* = 5**	***F* = 7**	**75:25 % Data Split**
DCT	200	95.61	96.07	96.33	**97.80**	92.01	93.86	94.11	**94.44**
Zernike	200	92.87	94.38	95.22	**96.85**	89.00	91.41	92.15	**93.96**
Gradient	200	96. 71	97.10	97.43	**98.56**	92.10	92.98	93.19	**94.17**

The graphical representation of results obtained at character level are shown in Figure 4.13

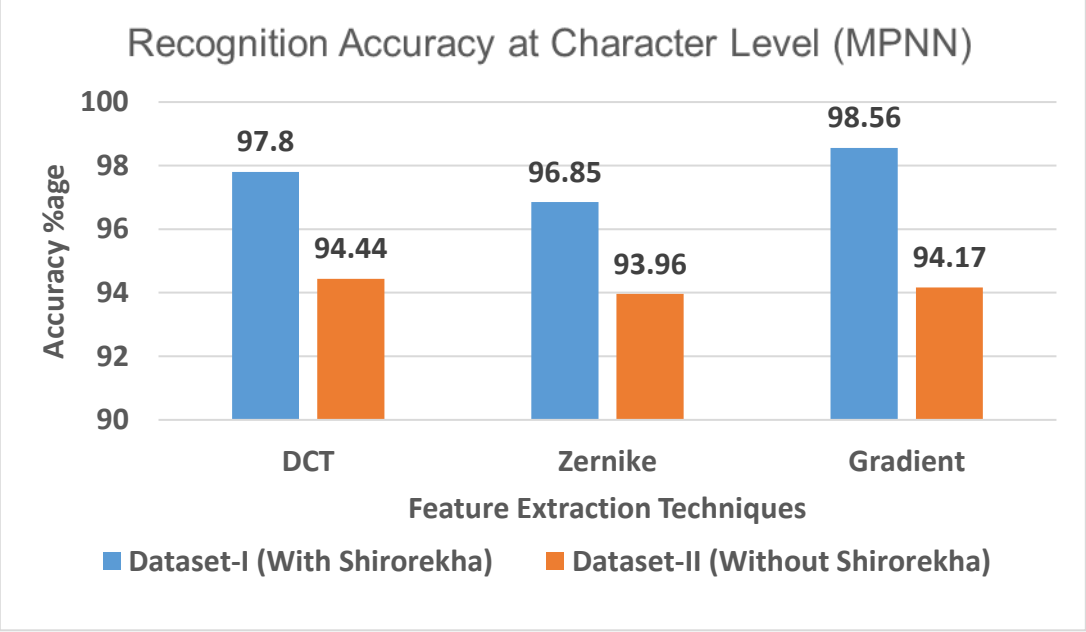


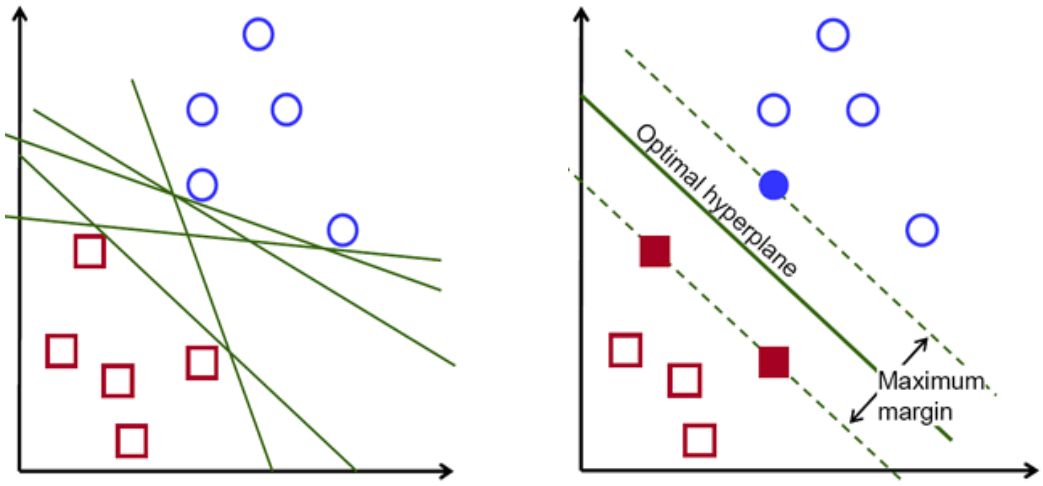
Figure 4.13 Graphical representation of results of MPNN classifier with different feature extraction methods

4.2.3 Support Vector Machine (SVM): It is a popular supervised learning technique associated with a group of learning algorithms for the classification of data (Frias-Martinez et al., 2006; Jayadevan et al., 2011). For a two class SVM, m-dimensional inputs X_i belongs to class *1* or *2* and the associated labels $Y_i = 1$ for class *1* and -1 for class 2, where $i = 1 \ldots M$. The decision function is given by equation 4.14.

$$F(x) = W^t x + b \tag{4.14}$$

where $Y_i F(x_i) \geq 1 - \xi_i \ \forall \ i = 1, \ldots, M$ and $\xi_i \geq 0$, W is an m-D vector and b is a scalar.

In other words, SVM is a discriminative classifier formally defined by a separating hyperplane. For a given labeled training data (supervised learning), the algorithm outputs an optimal hyperplane for the identification of the data. The working of SVM classifier is illustrated using Figure 4.14. As shown in the Figure 4.14(a), there are five hyper-planes (straight lines) and best hyper-plane needs to be identified to classify square boxes and circles. This task is accomplished by calculating the distances of individual hyper-planes from the objects as shown in Figure 4.14(b).



(a) (b)

Figure 4.14 Example of SVM classifier(a) input problem with many solutions and (b) solution using SVM

The combination of the SVM classifier (LibSVM) from weka tool kit with various feature extraction methods has been tested for character recognition. Testing was performed with the options cross validation (fold values $F = 3,5,7$) and percentage data split (training=75% and testing=25%). The character recognition results, obtained using feature extraction techniques and SVM classifier (Linear Kernal) are shown in Table 4.7.

Table 4.7 Recognition accuracy comparison of the characters with and without header line, using SVM(Linear)

Character Level Recognition Accuracy using Linear SVM (Dataset Size: 87000 Characters)									
Feature extraction technique	**Number of feature's**	**Dataset-I (With Shirorekha)**				**Dataset-II (Without Shirorekha)**			
		***F* = 3**	***F* = 5**	***F* = 7**	**75:25 % Data Split**	***F* = 3**	***F* = 5**	***F* = 7**	**75:25 % Data Split**
DCT	200	97.64	97.80	97.83	**98.78**	90.10	92.08	93.61	**95.62**
Zernike	200	92.11	93.42	94.57	**97.68**	83.22	85.65	86.24	**88.22**
Gradient	200	96.27	97.15	97.79	**99.10**	91.47	93.32	94.10	**95.29**

The graphical representation of results obtained at character level are shown in Figure 4.15

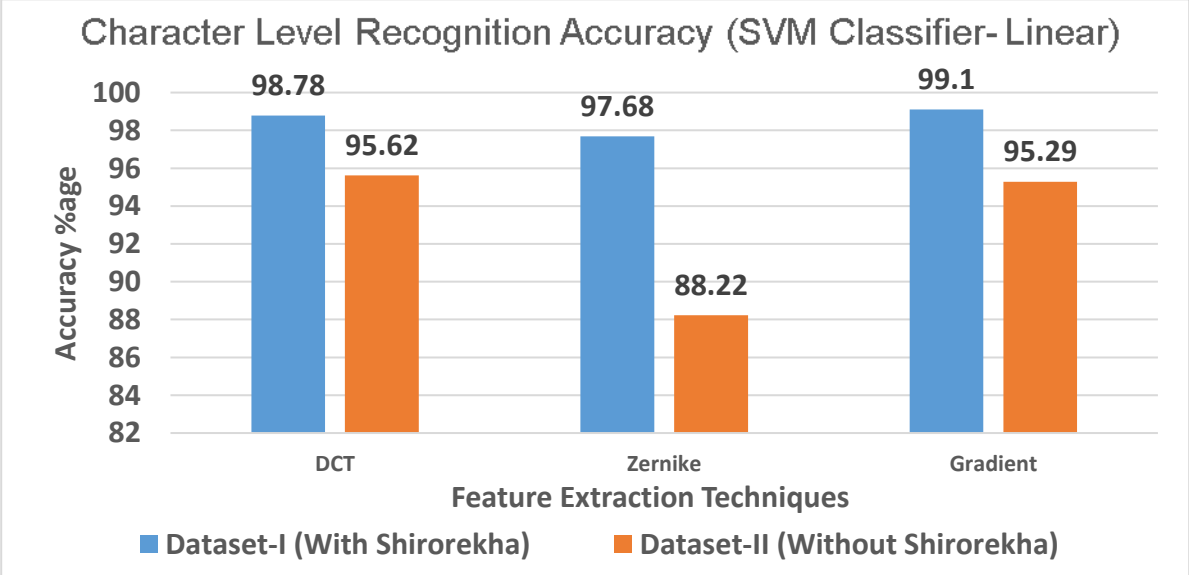


Figure 4.15 Graphical representation of results of SVM classifier(Linear) with different feature extraction methods

Table 4.8 Recognition accuracy comparison of characters with and without header line, using SVM (Polynomial)

Character Level Recognition Accuracy using Polynomial SVM (Dataset Size: 87000 Characters)									
Feature extraction technique	**Number of feature's**	**Dataset-I (With Shirorekha)**				**Dataset-II (Without Shirorekha)**			
		***F* = 3**	***F* = 5**	***F* = 7**	**75:25 % Data Split**	***F* = 3**	***F* = 5**	***F* = 7**	**75:25 % Data Split**
DCT	200	94.24	94.89	95.47	**96.12**	87.60	88.67	90.71	**93.11**
Zernike	200	90.10	93.42	93.88	**94.28**	80.52	82.55	83.00	**86.78**
Gradient	200	93.87	97.15	97.29	**97.71**	89.35	91.05	91.67	**93.84**

The graphical representation of results obtained at character level are shown in Figure 4.16

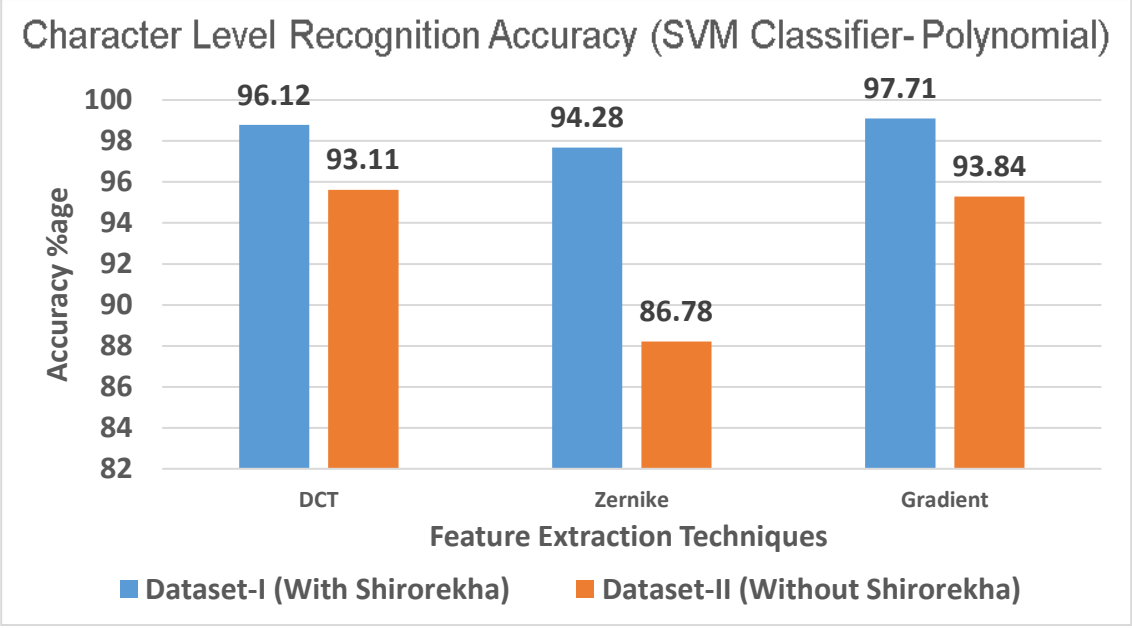
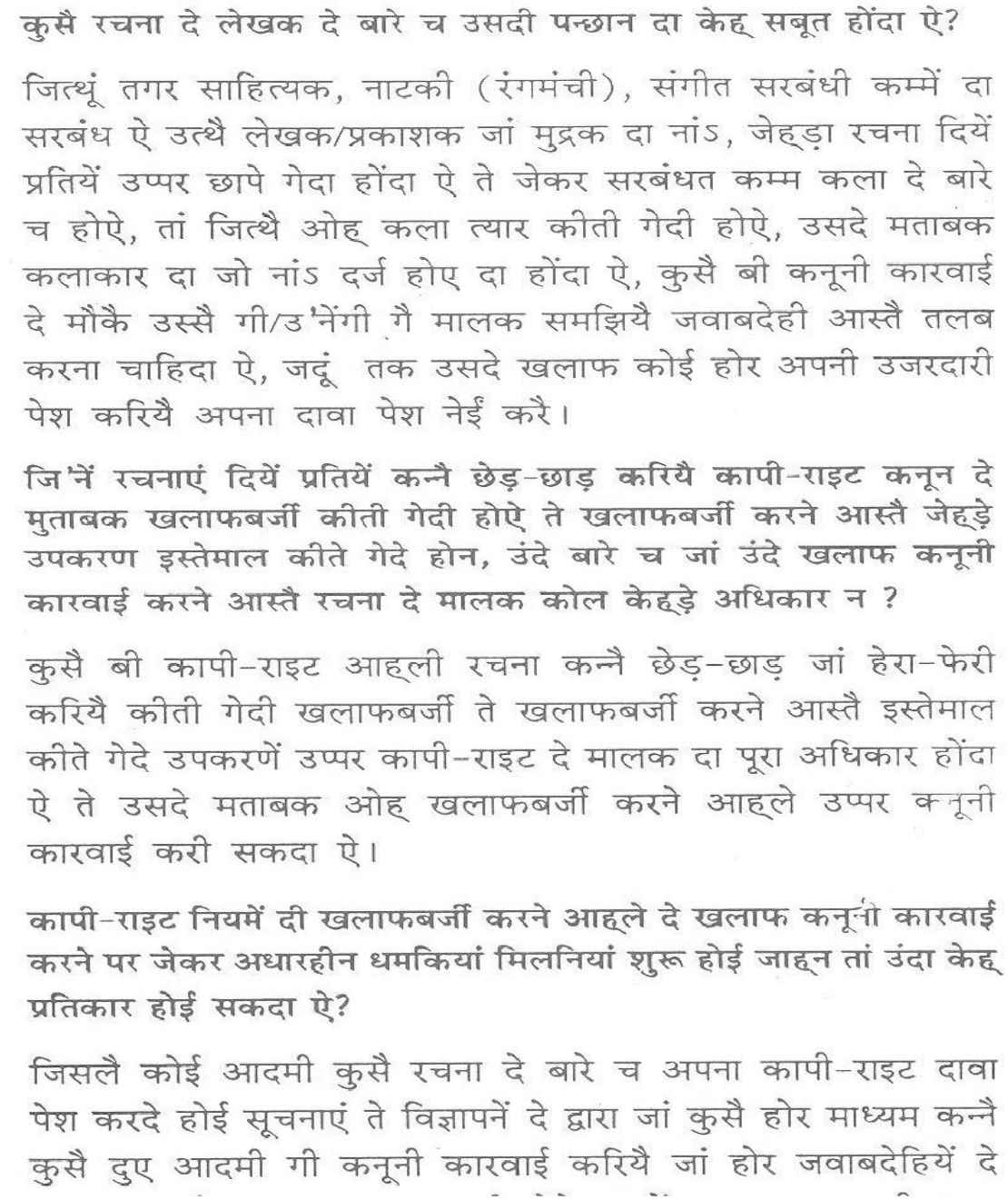


Figure 4.16 Graphical representation of results of SVM classifier (Polynomial) with different feature extraction methods

4.3 Recognition Engine Output

The sample of input images and the output of recognition engine on few Dogri language documents are shown in Figures 4.17, to 4.20.



कुसै रचना दे लेखक दे बारे च उसदी पन्छान दा केह् सबूत होंदा ऐ?

जित्थूं तगर साहित्यक, नाटकी (रंगमंची), संगीत सरबंधी कम्में दा सरबंध ऐ उत्थै लेखक/प्रकाशक जां मुद्रक दा नांऽ, जेह्ड़ा रचना दियें प्रतियें उप्पर छापे गेदा होंदा ऐ ते जेकर सरबंधत कम्म कला दे बारे च होऐ, तां जित्थै ओह् कला त्यार कीती गेदी होऐ, उसदे मताबक कलाकार दा जो नांऽ दर्ज होए दा होंदा ऐ, कुसै बी कनूनी कारवाई दे मौकै उस्सै गी/उ'नेंगी गै मालक समझियै जवाबदेही आस्तै तलब करना चाहिदा ऐ, जदूं तक उसदे खलाफ कोई होर अपनी उजरदारी पेश करियै अपना दावा पेश नेईं करै।

जि'नें रचनाएं दियें प्रतियें कन्नै छेड़-छाड़ करियै कापी-राइट कनून दे मुताबक खलाफबर्जी कीती गेदी होऐ ते खलाफबर्जी करने आस्तै जेह्ड़े उपकरण इस्तेमाल कीते गेदे होन, उंदे बारे च जां उंदे खलाफ कनूनी कारवाई करने आस्तै रचना दे मालक कोल केह्ड़े अधिकार न ?

कुसै बी कापी-राइट आह्ली रचना कन्नै छेड़-छाड़ जां हेरा-फेरी करियै कीती गेदी खलाफबर्जी ते खलाफबर्जी करने आस्तै इस्तेमाल कीते गेदे उपकरणें उप्पर कापी-राइट दे मालक दा पूरा अधिकार होंदा ऐ ते उसदे मताबक ओह् खलाफबर्जी करने आह्ले उप्पर कनूनी कारवाई करी सकदा ऐ।

कापी-राइट नियमें दी खलाफबर्जी करने आह्ले दे खलाफ कनूनी कारवाई करने पर जेकर अधारहीन धमकियां मिलनियां शुरू होई जाह्न तां उंदा केह् प्रतिकार होई सकदा ऐ?

जिसलै कोई आदमी कुसै रचना दे बारे च अपना कापी-राइट दावा पेश करदे होई सूचनाएं ते विज्ञापनें दे द्वारा जां कुसै होर माध्यम कन्नै कुसै दुए आदमी गी कनूनी कारवाई करियै जां होर जवाबदेहियें दे

Figure 4.17 Sample of Dogri language document image-1

कुसै रचना दे लेखक दे बारे च उसदी पन्छान दा केह सबूत होंदा ऐ?

जित्थू तगर साहित्यक, नाटकी (रंगमंची), संगीत सरबंधी कम्में दा सरबंध ऐ उत्थे लेखक/प्रकाशक जां मुद्रक दा नांs, जेहड़ा रचना दियें प्रतियें उप्पर छापे गेदा होंदा ऐ ते जेकर सरबंधत कम्म कता दे बारे च होऐ, तां जित्थे ओह कला त्यार कीती गेदी होऐ, उसदे मताबक कलाकार दा जो नांs दर्ज होए दा होंदा ऐ, कुसै बी कनूनी

कारवाई दे मौकै उस्सै गी/उ नेंगी गै मालक समझियै जवाबदेही आस्ते तलब करना चाहिदा ऐ, जदूं तक उसदे खलाफ कोई होर अपनी उजरदारी पेश करिये अपना दावा पेश नेई करे।

जिनें रचनाएं दियें प्रतियें कन्नै छेड़-छाड़ करियै कापी-राइट कनून दे मुताबिक खलाफबर्जी कीती गेदी होऐ ते खलाफबर्जी करने आस्तै जेहड़े उपकरण इस्तेमाल कीते गेदे होन, उंदे बारे च जां उंदे खलाफ कनूनी कारवाई करने आस्तै रचना दे मालक कोल के हड़े अधिकार न ?

कुसै बी कापी-राइट आहली रचना कन्नै छेड़-छाड़ जां हेरा-फेरी करिये कीती गेदी खलाफबजी ते खलाफबर्जी करने आस्ते इस्तेमाल कीते गेदे उपकरणें उप्पर कापी-राइट दे मालक दा पूरा अधिकार होंदा ऐ ते उसदे मताबक ओह खलाफबर्जी करने आहले उपर कानूनी कारवाई करी सकदा ऐ।

कापी-राइट नियमें दी खलाफबर्जी करने आहले दे खलाफ कनूनी कारवाई करने पर जेकर अधारहीन धमकियां मिलनियां शुरू होई जाहन तां उंदा केह प्रतिकार होई सकदा ऐ?

जिसलै कोई आदमी कुसै रचना दे बारे च अपना कापी-राइट दावा पेश करदे होई सूचनाएं ते विज्ञापनें दे दारा जां कुसै होर माध्यम कन्नै कुसै दुए आदमी गी कनूनी कारवाई करिये जां होर जवाबदेहियें दे

Figure 4.18 Recognition output of sample Dogri language document image-1

भूमिका

जि'यां के अस जानने आं जे डोगरी गद्य दा इतिहास इक सदी पैह्लें शुरू होआ हा, जदूं महाराजा रणबीर सिंह होरें प्रकाशत रूपा च इसदी नींह् रक्खने दा हीला कीता हा। उ'नें डोगरी गद्य गी आम बरतून च फ़रोग देने लेई मती सारियें संस्कृत पुस्तकें दे अनुवाद बी करोआए। सीरामपुर दे ईसाई मिशनरियें दा 1793 ई. दा प्रयास डोगरी गद्य-शैली दे विकास च इक होर मील-पत्थर दी हसीयत रखदा ऐ, जिसदे फलसरूप सैह्ल-सुबोध शैली च डोगरी बाइबल दे प्रकाशन कन्नै डोगरी गद्य दा सफर रंभ होआ। 'राजावलि' जां 'राजौली' डोगरी गद्य दे विकास दा इक होर मील-पत्थर हा, जिसदी शैली डोगरी भाशा दी कांगड़ी बानगी उप्पर अधारत ही। गुलेर दे शाही संग्रैह् चा इतिहासक म्हत्ता आह्ली इस पुस्तक दा मिलना इसदे भाशाई इतिहास च म्हत्तवपूर्ण थाह्र रखदा ऐ। बुनियादी तौरा पर एह् पुस्तक मुगल शहजादे दारा शिकोह दे कम्में दे बारे च ही, जेह्ड़ी मूल रूपा च फ़ारसी भाशा च ही। इस थमां पता लगदा ऐ जे एह् भाशा केइयें सदियें थमां इस प्रदेश दे लेखकें च लखाई आस्तै बरतोने दा माध्यम ही।

इक विद्वान-यात्री फ्रैड्रिक ड्रयू, जेह्ड़े 1862 ई. थमां 1872 ई. तगर इस रियासता च रेह्, उ'नें अपनी पुस्तक च महाराजा दे शाही दरबार, जिस च सब्भै दस्तावेज डोगरी च हे, बारै लेखा-जोखा प्रस्तुत कीते दा ऐ। ड्रयू होरें डोगरी भाशा ते इसदे व्याकरण दा ब्योरा बी त्यार कीता। कीजे मुकामी कनून-पुस्तकां डोगरी च उपलब्ध हियां, इसकरी एह् शैल चाल्ली अदालती भाशा दी भूमिका नभाई सकदी ही। शाही खानदान दे सदस्यें ते सरकारी अधिकारियें मझाटै चिट्ठी-पत्तरी च बरतोई दी डोगरी गद्य दी शैली रवायती

Figure 4.19 Sample of Dogri language document image-2

भूमिका

जियां के अस जानने आं जे डोगरी गद् दा इतिहास इक सदी पैहलें शुरू होआ हा, जर्दू महाराजा रणबीर सिंह होरें प्रकाशत रूपा च इसदी नींह रक्खने दा हीला कीता हा। उ नें डोगरी गद् गी आम बरतून च फ़रोग देने लेई मती सारियें संस्कृत पुस्तकें दे अनुवाद बी करोआए। सीरामपुर दे ईसाई मिशनरियें दा 1793 ई. दा प्रयास डोगरी गद्-शैली दे विकास च इक होर मील-पत्थर दी हसीयत रखदा ऐ, जिसदे फलसरूप सैहल-सुबोध शैली च डोगरी बाइबल दे प्रकाशन कन्न डोगरी गद् दा सफर रंभ होआ। 'राजावलि' जां 'राजौली' डोगरी गद् दे विकास दा इक होर मील-पत्थर हा, जिसदी शैली डोगरी भाशा दी कांगड़ी बानगी उप्पर अधारत ही। गुलेर दे शाही संग्रैह चा इतिहासक म्हत्ता आहली इस पुस्तक दा मिलना इसदे भाशाई इतिहास च म्हत्तवपूर्ण थाहर रखदा ऐ। बुनियादी तौरा पर एह पुस्तक मुगल शहजादे दारा शिकोह दे कम्में दे बारे च ही, जेहड़ी मूल रूपा च फ़ारसी भाशा च ही। इस थमां पता लगदा ऐ जे एह भाशा केइयें सदियें थमां इस प्रदेश दे लेखकें च लखाई आस्तै बरतोने दा माध्यम ही।

इक विदान-यात्री फ्रैंड्रिक ड्रयू, जेहड़े 1862 ई. थमां 1872 ई. लगर इस रियासता च रेह, उ नें अपनी पुस्तक च महाराजा दे शाही दरबार, जिस च सब्भे दस्तावेज डोगरी च हे, बारे लेखा-जोखा प्रस्तुत कीते दा ऐ। ड्रयू होरें डोगरी भाशा ते इसदे व्याकरण दा ब्योरा बी त्यार कीला। कीजे मुकामी कनून-पुस्तकां डोगरी च उपलब्ध हियां, इसकरी एह शैल चाल्ती अदालती भाशा दी भूमिका नभाई सकदी ही। शाही खानदान दे सदस्यें ते सरकारी अधिकारियें मझाटै। चिट्ठी-पत्तरी च बरतोई दी डोगरी गद् दी शैली रवायती

Figure 4.20 Recognition output of sample Dogri language document image-2

4.4 Confusion Matrix

During experimentation, it was analyzed that there are certain characters that are confused with other character classes. The data given in Table 4.9 represents the confusion matrix for some of the characters of Dogri language.

Table 4.9 Confusion matrix of similar looking characters

S.No.	Character	Closest Match %age with Similar Shape Characters				
1.	क	क (98.64%)	क्र (0.95%)	क़ (0.30%)	फ (0.11%)	
2.	ख	ख (99.27%)	ख़ (0.60%)	स (0.13%)		
3.	ग	ग (98.92%)	ा (0.60%)	र (0.48%)		
4.	घ	घ (98.98%)	ध (1.2%)			
5.	च	च (97.95%)	व (1.58%)	ब (0.47%)		
6.	छ	छ (98.14%)	घ (1.42%)	ध (0.44%)		
7.	ज	ज (97.40%)	ज़ (1.64%)	ज्ञ (0.86%)	ञ (0.10%)	
8.	झ	झ (99.89%)	इ (0.11%)			

9.	ञ	ञ (98.19%)	त्र (0.92%)	व (0.89%)		
10.	त्र	ञ (99.86%)	व (0.14%)			
11.	ट	ट (97.44%)	द (1.87%)	ढ (0.63%)	ढ़ (0.06%)	
12.	ठ	ठ (99.55%)	ट (00.37%)	ढ (0.8%)		
13.	ड	ड (98.82%)	ड़ (0.29%)	इ (0.19%)	ड्र (0.7%)	
14.	ढ	ढ (99.17%)	ढ़ (0.41%)	द (0.24%)	ट (0.18%)	
15.	ण	ण (99.31%)	प (0.69%)			
16.	त	त (98.58%)	त्त (0.88%)	ल (0.43%)	न (0.11%)	
17.	थ	थ (99.25%)	य (0.56%)	भ (0.19%)		
18.	द	द (97.72%)	द् (1.48%)	ट (0.53%)	ढ (0.20%)	ढ़ (0.07%)
19.	ध	ध (99.41%)	घ (0.51%)	थ (0.08%)		
20.	न	न (98.53%)	ऩ (0.89%)	त (0.37%)	भ (0.17%)	म (0.04%)

21.	प	प (98.87%)	ण (0.76%)	ष (0.24%)	फ (0.08%)	य (0.05%)
22.	फ	फ (99.37%)	क (0.49%)	क़ (0.09%)	क्र (0.05%)	
23.	ब	ब (98.29%)	व (1.10%)	च (0.61%)		
24.	भ	भ (99.16%)	भ्र (0.53%)	म (0.18%)	थ (0.13%)	
25.	म	म (99.20%)	न (0.68%)	भ (0.12%)		
26.	य	य (98%)	प (1%)	म (0.74%)	थ (0.26%)	
27.	र	र (98.71%)	ऱ (0.96%)	ग (0.33%)		
28.	ल	ल (98.17%)	न (1.53%)	त (0.30%)		
29.	व	व (98.19%)	ब (1.67%)	व्र (0.10%)	च (0.04%)	
30.	श्र	श्र (99.79%)	र (0.21%)			
31.	ष	ष (98.62%)	प (1.09%)	ण (0.29%)		
32.	स	स (98.57%)	म (0.96%)	त (0.37%)	भ (0.1%)	

33.	ह	ह (98.19%)	ह् (1.4%)	ड़ (0.41%)		
34.	क्ष	क्ष (99.68%)	श्र (0.32%)			
35.	श्र	श्र (99.70%)	क्ष (0.30%)			
36.	ज्ञ	ज्ञ (98.18%)	ज (1.52%)	ञ (0.3%)		
37.	अ	अ (99.04%)	ऊ (0.61%)	उ (0.35%)		
38.	ड़	ड़ (97.66%)	ड्र (1.77%)	ड़ (0.43%)	ड (0.14%)	
39.	उ	उ (99.66%)	ऊ (0.34%)			
40.	ऊ	ऊ (99.81%)	उ (0.19%)			
41.	ऋ	ऋ (99.48%)	त्र (0.30%)	ञ (0.22%)		
42.	ए	ए (99.81%)	प (99.81%)			
43.	ा	ा (99.10%)	ग (0.63%)	। (0.27%)		
44.	ऽ	ऽ (99.40%)	5 (0.60%)			

45.	ا	ا (99.64%)	◌ٰا (0.25%)	1 (0.11%)		
46.	◌ٔ	◌ٔ (98.65%)	‘ (1.35%)			
47.	‘	‘ (98.40%)	◌ٕ (1.10%)	’ (0.50%)		
48.	’	’ (98.31%)	◌ٖ (1.52%)	‘ (0.17%)		
49.	ﭬ◌	ﭬ◌ (100%)				
50.	◌ٓا	◌ٓا (98.49%)	◌ٓا (1.22%)	◌ٰا (0.18%)	◌ٰا (0.11%)	
51.	◌ٗ	◌ٗ (98.93%)	◌ٙ (0.80%)	◌ٚ (0.27%)		
52.	◌ٙ	◌ٙ (99.02%)	◌ٛ (0.90%)	◌ٕ (0.08%)		
53.	◌ٛ	◌ٛ (98.60%)	◌ٙ (0.95%)	◌ٖ (0.45%)		
54.	◌ٚ	◌ٚ (98.13%)	◌ٛ (1.18%)	◌ٗ (0.46%)	◌ٕ (0.23%)	
55.	◌ٓا	◌ٓا (98.61%)	◌ٰا (1.17%)	◌ٰا (0.22%)		
56.	◌ٰا	◌ٰا (99.02%)	◌ٓا (0.98%)			

57.	◌ٱ	◌ٱ (98.81%)	◌ا (1.10%)	◌آ (0.9%)		
58.	◌ٕ	◌ٕ (98.40%)	◌ٖ (1.34%)	◌ٔ (0.26%)		
59.	◌ٮ	◌ٮ (99.15%)	◌ٖ (0.85%)			
60.	◌ٖ	◌ٖ (98.59%)	◌ٕ (1.24%)	◌ٮ (0.17%)		
61.	◌ٔ	◌ٔ (99.81%)	◌ٕ (0.19%)			
62.	0	0 (100%)				
63.	1	1 (99.06%)	l (0.76%)	◌T (0.18%)		
64.	2	2 (100%)				
65.	3	3 (100%)				
66.	4	4 (100%)				
67.	5	5 (100%)	s (100%)			
68.	6	6 (100%)				

69.	**7**	**7** (100%)				
70.	**8**	**8** (100%)				
71.	**9**	**9** (100%)				

As evident from Table 4.9, the shape based technique was having very less number of confused characters in comparison with the existing techniques, outperforms and contribute in higher recognition results.

4.5 Result Comparison

The classification results were then compared with the existing results published elsewhere. Jayadevan et al. (2011) has presented a survey of the results of various feature extraction and classification techniques on Devanagari script based documents. The results of the proposed shape base method has, thus been compared with the reported techniques. As indicated in the Table 4.10, the maximum accuracy of 96.00% has been reported by Kompalli et al. (2009). Whereas, the proposed shape based technique resulted in an accuracy of 99.10%, which is 3.10% higher than existing techniques.

Additionally, experimentation has been done using SVM classifier with the polynomial option and the results obtained are shown in Table 4.8 and Figure 4.16. On comparing the results of SVM Linear and Polynomial, it was found that the results of Linear SVM exceeded the output of Polynomial SVM.

Table 4.10 Character level accuracy comparison of the proposed method with existing techniques

Author	Feature	Classifier	Shirorekha Present	Data Set (Size)	Accuracy (%age)
(Sinha and Mahabala, 1979)	Structural	Syntactic pattern analysis	No	NA	90.00
(Jayanthi et al., 1989)	Statistical	Binary tree	No	4863	95.08
(Chaudhuri and Pal, 1997c)	Statistical	Tree classifier, template matching	No	10,000	95.42
(Bansal and Sinha, 2000)	Statistical, Structural	Statistical Knowledge Sources	No	NA	87.00
(Ma and Doermann, 2003)	Structural, Statistical	Hausdorff image comparison	No	2,727	88.24
(Govindaraju et al., 2004)	Gradient	Neural networks	No	4,506	84.00
(Kompalli and Setlur, 2005)	GSC	Neural networks	No	32,413	84.77
(Dhurandhar et al., 2005)	Contours	Interpolation	No	546	93.03
(Kompalli et al., 2006)	GSC	K-nearest neighbors	No	9,297	95.00
(Natarajan et al., 2009)	Derivatives	HMM	No	21,982	91.30
(Kompalli et al., 2009)	SFSA	Stochastic finite state automaton	No	10,606	96.00
Proposed Approach	**Gradient**	**MPNN**	**Yes**	**87,000**	**98.56%**
Proposed Approach	**DCT**	**SVM**	**Yes**	**87,000**	**98.78%**
Proposed Approach	**Gradient**	**SVM**	**Yes**	**87,000**	**99.10%**

As shown in Table 4.10, the combination of the DCT with SVM and Gradient with SVM (linear kernel) performed better than other techniques using proposed shape based features. Overall, higher accuracy results are achieved on using shape based technique with different feature extraction and classification techniques. The percentage increase in character recognition accuracy is shown in Table 4.11.

Table 4.11 Percentage recognition improvement at character level using the proposed segmentation method

Feature Extraction Method	Percentage Improvement with Proposed Shape Based Algorithm		
	kNN	MPNN	SVM (Linear)
DCT	1.36%	1.80%	2.78%
Zernike	0.15%	0.85%	1.68%
Gradient	2.10%	2.56%	3.10%

As shown in Table 4.11 the highest character recognition accuracy improvement of 3.10% is obtained using Gradient features with SVM (linear kernel) classifier.

4.6 Conclusion

The features has been extracted using techniques such as DCT, Zernike and Gradient. The extracted feature files are then provided as input to the classifier. The classification of characters have been performed using different classifiers available in weka toolkit such as Multilayer perceptron neural networks (MPNN), Support Vector Machines (SVM) and *k*-Nearest Neighbors (*k*-NN).

In the present work, the combination of various feature extraction and classification techniques has been used for the recognition of characters that was a combinations of feature extraction techniques (DCT, Zernike, Gradient) and classification techniques (kNN, MPNN, SVM). For performance evaluation, the character recognition results obtained using the proposed shape-based segmentation algorithm were compared with the

results of existing techniques as shown in Table 4.10. As indicated in Table 4.10, the existing maximum accuracy of 96% on printed Devanagari documents has been reported by Kompalli and Setlur (2006). Whereas, the proposed character recognition system has achieved an impressive overall 98.56% to 99.10% recognition rate (depending upon the classifier used). The maximum recognition accuracy of 99.10% has been obtained at character level with the combination of Gradient feature descriptors and linear SVM using shape based segmentation algorithm. A notable increase of 3.10% has been obtained in the character recognition accuracy using shape based algorithms in comparison with existing techniques.

The input image sample and digitized output of the recognition system has also been illustrated in Figure 17 to Figure 20. The error analysis has been performed on the misclassified character classes and the confusion matrix has been formulated as shown Table 4.9. It has been analyzed that most of the classification errors occurred due to closely packed (merged), similar shape and broken characters.

Chapter 5

Post-processing

The post-processing method is also known as language model and is used to correct recognition errors. Its major goal is to detect and correct the misclassified characters/ words produced by the recognition system. There are two options to implement this technique. First is by manually checking the recognition output and the second is using a program that automatically detects and correct errors. The manual processing will take immense levels of human effort and time, and therefore it is important to deploy an automated technique. A common problem that is encountered with almost all the optical character recognition systems is that, after recognition of characters, some errors continue to occur due to software limitation, quality of documents or human mistakes.

To resolve these errors some techniques are used like rule-based contextual processing (Sinha, 1987; Nagata, 1998; Chaudhuri, 2002; Lehal, 2013) coding of visually same characters (Lehal et al., 2001), dictionary look-up technique (Riseman and Hanson, 1974; Wells et al., 1990; Mayes et al., 1991; Bansal and Sinha, 1999) and weighted finite-state transducer (Chowdhury et al., 2011). Out of these techniques, two methods are most popularly used for finding out the correctness of the recognized word. The first method is based upon position probability of the characters i.e., estimation of occurrence of a spelling in the recognized word. To achieve this, the statistical knowledge of the script is required (Tong and Evans, 1996). The method works using a n-gram model (character string of size n) to rule out the character string candidates that can not the accepted

(Riseman and Hanson, 1974; Suen, 1979; Yannakoudakis et al., 1990; Kompalli and Setlur, 2006; Remus and Rill, 2013). On the other hand, second method works on the basis of the language dictionary (Wells et al., 1990). In this technique, if the word is matched with some pre-existing word in the dictionary, then it is marked as correct, else further investigation is performed. In addition to the methods discussed, some spell checker and correction utilities works on word similarity strategy i.e., optimal number of suggestions for similar type of words using the dictionary partitioning scheme. In most of the post processing implementations, a combined approach is deployed for the correction of errors left during the recognition process. In Roman script post processing techniques, the word is corrected using dictionary of statistical information about the language. In contrast to that, in most of the Indian scripts, a word is separated into individual character segments. Therefore, after classification, there is need of intermediate phase that re-joins the individual characters to compose back the original word. It is also possible that during the classification process, a character can be classified into multiple classes, then in that case multiple words are formulated with the classified character classes. If a constituted word cannot be validated by using the word dictionary, then, it can be corrected using language composition rules.

A review of advancements in the post processing techniques exclusively for Indian languages explored that, most of the recognition errors in Indian script occur due to similar shape and confusing characters (Chaudhuri, 2002). For instance, the Dogri and Hindi languages share common script which is known as Devanagari script. There are a number of common words in said languages, but have different meaning. So, the existing post-

processing programs of Devanagari script cannot be deployed for detection and correction Dogri language words. Additionally, a number of other factors like confusing shapes, size, modifiers, and structure loss also influence the recognition errors which are also discussed in this chapter.

During segmentation of Dogri language documents, the line numbers are preserved along with word position, character location, upper and lower modifier index for all the characters in a text file. With the help of the data of said text file, the complete page can be regenerated in digital format using the equation 5.1. The working of the text regeneration process is illustrated using few lines from a Dogri language document and shown in Figure 5.1 and Table 5.1 to 5.3.

$$R_n = l_n + w_m + c_i \{ + \text{Modifier index LM or UM (if any)}\} \quad (5.1)$$

where R_n = Regenerated word, l_n= line number, w_m = word number, c_i = character position, LM = lower modifier and UM = upper modifier, $\{n, m \text{ and } i\} \geq 1$ are positive integers.

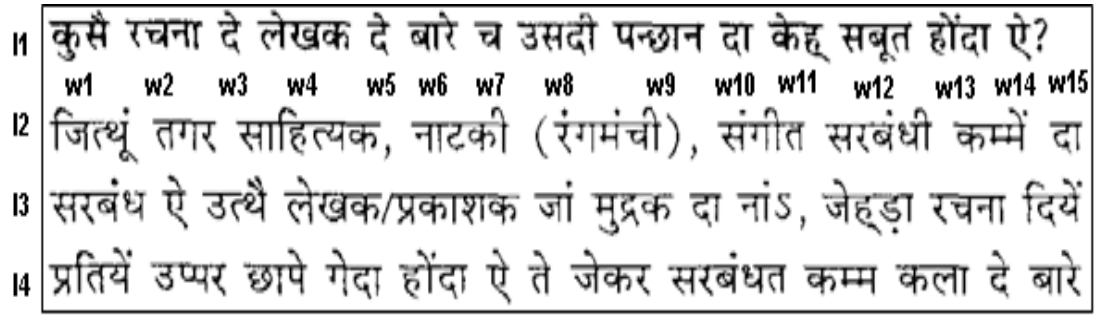


Figure 5.1 Sample of Dogri language document illustrating lines and words

The segmented lines with line numbers are shown in Table 5.1, segmented words with word position is shown in Table 5.2 and finally, the segmented characters with location index value is shown in Table 5.3.

Table 5.1 Segmented lines of Dogri language document

S. No	Segmented Line Images	Line Number
1.	कुसै रचना दे लेखक दे बारे च उसदी पन्छान दा केह् सबूत होंदा ऐ?	p1.l1.bmp
2.	जित्थूं तगर साहित्यक, नाटकी (रंगमंची), संगीत सरबंधी कम्में दा	p1.l2.bmp
3.	सरबंध ऐ उत्थै लेखक/प्रकाशक जां मुद्रक दा नांऽ, जेह्ड़ा रचना दियें	p1.l3.bmp
4.	प्रतियें उप्पर छापे गेदा होंदा ऐ ते जेकर सरबंधत कम्म कला दे बारे	p1.l4.bmp

Table 5.2 Segmented words of line no l1and word position index

Word No.	Segmented Word Images	Word Position
w1	कुसै	p1.l1.w1.bmp
w2	रचना	p1.l1.w2.bmp
w3	दे	p1.l1.w3.bmp
w4	लेखक	p1.l1.w4.bmp
w5	दे	p1.l1.w5.bmp
w6	बारे	p1.l1.w6.bmp
w7	च	p1.l1.w7.bmp
w8	उसदी	p1.l1.w8.bmp
w9	पन्छान	p1.l1.w9.bmp
w10	दा	p1.l1.w10.bmp
w11	केह्	p1.l1.w11.bmp
w12	सबूत	p1.l1.w12.bmp
w13	होंदा	p1.l1.w13.bmp
w14	ऐ	p1.l1.w14.bmp
w15	?	p1.l1.w15.bmp

Table 5.3 Segmented characters from words of line no l1and character position index

Word No.	Character and modifier location index			
w1	क	◌	स	◌
	p1.l1.w1.c1.bmp	p1.l1.w1.c1.LM1.bmp	p1.l1.w1.c2.bmp	p1.l1.w1.c2.UM1.bmp
w2	र	च	न	ा
	p1.l1.w2.c1.bmp	p1.l1.w2.c2.bmp	p1.l1.w2.c3.bmp	p1.l1.w2.c4.bmp
w3	द	◌		
	p1.l1.w3.c1.bmp	p1.l1.w3.c1.UM1.bmp		
w4	ल	◌	ख	क
	p1.l1.w4.c1.bmp	p1.l1.w4.c1.UM1.bmp	p1.l1.w4.c2.bmp	p1.l1.w4.c3.bmp
w5	द	◌		
	p1.l1.w5.c1.bmp	p1.l1.w5.c1.UM1.bmp		
w6	ब	ा	र	◌
	p1.l1.w6.c1.bmp	p1.l1.w6.c2.bmp	p1.l1.w6.c3.bmp	p1.l1.w6.c3.UM1.bmp
w7	च			
	p1.l1.w7.c1.bmp			
w8	उ	स	द	ी
	p1.l1.w8.c1.bmp	p1.l1.w8.c2.bmp	p1.l1.w8.c3.bmp	p1.l1.w8.c3.UM1.bmp
w9	प	छ	ा	न
	p1.l1.w9.c1.bmp	p1.l1.w9.c2.bmp	p1.l1.w9.c3.bmp	p1.l1.w9.c4.bmp
w10	द	ा		
	p1.l1.w10.c1.bmp	p1.l1.w10.c2.bmp		
w11	क	◌	ह	◌
	p1.l1.w11.c1.bmp	p1.l1.w11.c1.UM1.bmp	p1.l1.w11.c2.bmp	p1.l1.w11.c2.UM1.bmp
w12	स	ब	◌	ते
	p1.l1.w12.c1.bmp	p1.l1.w12.c2.bmp	p1.l1.w12.c2.LM1.bmp	p1.l1.w12.c3.bmp
w13	ह	ो	द	ा
	p1.l1.w13.c1.bmp	p1.l1.w13.c1.UM1.bmp	p1.l1.w13.c2.bmp	p1.l1.w13.c3.bmp
w14	ए	◌		
	p1.l1.w14.c1.bmp	p1.l1.w14.c1.UM1.bmp		
w15	?			
	p1.l1.w15.c1.bmp			

Finally, all the regenerated words are stored in a text file in the Unicode font format. At this point, the post-processing technique is initiated.

5.1 Similar Shapes of Dogri Language Characters

Generally, the character recognition system misclassifies some of the characters that visually looks similar as shown in Table 5.4.

Table 5.4 Similar looking shapes of Dogri language characters

Symbol	Confused Character Classes					
क	फ	क़	फ़	क्र	फ्र	व
ख	स	छ				
ॻ	श	र	ा			
घ	ध					
ड	ङ	ड़	७			
च	व					
ज	ञ	ज़	ज्ञ			
ट	ढ	द	ढ़			
ठ	ट					
ण	प					
त	ल					
थ	प	य	भ			
न	म					
प	ष	५				
ब	व					
भ	म	य				
श	श ा	र ा	ऱ ा			
उ	ऊ	3				
ा	।	1	ॻ			
ी	ॊ	ो				
ै	े	ै				

5.2 Recognition Engine Output

Some of the errors of recognition engine output are shown in Table 5.5. For illustration purpose the misclassified characters are shown in red color. These errors occur due to factors like noisy text, improper segmentation, degraded and broken text.

Table 5.5 Sample errors left out by the recognition engine due to similar looking shapes

S.No	Wrongly Recognized Output	Desired Output
1.	अटालते दा समां ।	अदालते दा समां ।
2.	अयने सयने	अपने सपने
3.	यतंग उड़ा लो ।	पतंग उड़ा लो ।
4.	सभझने लगा है ।	समझने लगा है ।
5.	फब जाना है ।	कब जाना है ।
6.	दर्जने घाडल ।	दर्जने घाऽल ।
7.	किसका ज़ोर वलता है ।	किसका ज़ोर चलता है ।
8.	पबन चला गया ।	पवन चला गया ।
9.	ऽोगरी पहाड़ी भाषा है ।	डोगरी पहाड़ी भाषा है ।
10.	तलाडच नहा लो ।	तलाऽ च नहा लो ।
11.	दाड लगा ले ।	दाऽ लगा ले ।
12.	उयलब्ध करोआने लेई सरकार ।	उपलब्ध करोआने लेई सरकार ।
13.	रवड़ा रहना ।	खड़ा रहना ।
14.	टेवनागरी लिपि ।	देवनागरी लिपि ।

The output of recognition system, has thus been, improvised to enhance the overall accuracy of OCR. To achieve high accuracy by correcting recognition errors, the partitioned dictionary of words have been formulated and deployed that are collected from Dogri and Hindi language documents. The dictionary partitioning technique helps in reducing complexity and search time (Bansal and Sinha, 2002b). The words shown in Table 5.5 are identified and corrected using the following steps:

a) The program start with the input word and then the corresponding dictionary is explored. All the probable words that have some similarity with the input word are shortlisted in a temp array.

b) Then a comparison is performed between the input word and the shortlisted words.

c) If the input word is completely matched with the word present in the dictionary, then the word is considered as correct and then next word in the sequence is checked.

d) If the matching fails, then, either closest distance match is considered for the input word or the aliases are formulated as per the input word in order to find the best match.

5.3 Word Occurrence Frequency

A partitioned dictionary of Dogri and Hindi language words has been formulated as a part of this exercise. Additionally, the occurrence frequency of frequently occurring words has been calculated and is shown in Table 5.6. The said dataset has been created from the huge collection of words (corpus of more than fifty lac words) of Dogri and Hindi language online books, documents, magazines, newspapers etc. All the words in the said text file have been arranged sequentially and allotted a unique identification number, as explained in Table 5.6.

Table 5.6 Occurrence frequency of Dogri and Hindi language words

S.No	Word	Occurrence Frequency	S.No	Word	Occurrence Frequency
1.	के	166560	2.	भी	79108
3.	में	95367	4.	कि	69260
5.	की	107077	6.	नहीं	77538
7.	से	103157	8.	ही	64904
9.	और	100224	10.	एक	64097
11.	का	101348	12.	तो	61037
13.	को	92970	14.	ने	55863
15.	है.	81620	16.	हो	54227
17.	है	85018	18.	यह	54196
19.	पर	72965	20.	इस	47323

As a demonstrative illustration, the post-processor receives the input word as **पबन**, then**,** it immediately starts exploring words in its partitioned dictionary having an exact or closest match. If the exact input word is not present, then all the nearest matching words will shortlisted as suggestion like for the word **पबन** the shortlisted words are {**पवन, पवण, पावन, पवून, पावउँ, पावनी**}. Then, on the basis of highest ranking match, the eligible word will become **पवन** and is swapped with the wrongly recognized word.

5.4 Error Correction using Dictionary Lookup Technique

The dictionary lookup technique makes comparison of the words of recognition system output with the available words in the lexicon. If a word recognized by the OCR exists in the lexicon then it is considered as the correct one, otherwise search is initiated for the most probable word (w_i) based upon the sequence of similar words $\{w_{i-(n-1)}, \dots, w_{i-1}\}$.

Table 5.7 Example of few words and corresponding candidate choices

S.No	OCR text	Correction Candidate Words
1.	कुत	कुली, कूल, कल, कली, कालू, कुल, कुला, कला
2.	अर	और, ओर, औऱ, औरो, औरों, ऑरो, आरा, अरे
3.	गह	ग्रह, ग्राह, गृह, गोह, गोह, गर्भ
4.	क्म	कर्म, कर, कार, कार्य, कर्ज, क्रम, क़र, कर्मी
5.	मन	मैना, मुन्नी, मना, मन, मैनु, मैनू, मौन, मॉन, मान
6.	दिम	दीन, दीं, दिज्ञी, दिने, दिज्ञी, दीनी, दिन
7.	माञ	मातृ, मत, मात, मैत, मात्र, मूत्र , मट, मित्र, मित्रो
8.	बाम	बाग, बैग, बग, बाघ, बाग़, बागी, बाग़ी, बाजी, बगी
9.	पक	पके, पक्का, पाक, पेका, पैक, पीके, पका
10.	संम	सांग, संघ, सान्ग, सँग, संग

For the replacement of incorrect word, initially, a confusion matrix was prepared from the word lexicon file that contains similar words and their occurrence frequencies/ ranks. The highest ranked word was selected as most suitable candidate for the replacement of incorrect word. Few examples of such words with correction candidate list of probable word replacements are shown in Table 5.7. The words shown in red color in Table 5.7 represents the correct word that needs to be replaced with the incorrectly classified OCR text.

5.5 Data Collection and Training

The dictionary based language model was deployed for the correction of errors left during the recognition stage. Initially the model was trained on the language data. For experimentation purpose, large dataset of Dogri and Hindi language words was gathered from different sources like online story books and newspapers. Then, the probable two and three pair word combinations were compiled along with their occurrence frequencies in a text file.

For example the possible combinations of two and three consecutively occurring words of following paragraph **"केंद्रीय मंत्रिमंडल अगले हफ्तै लगभग 5000 करोड़ पैंशन फार्मूले गी अनुमति देग जिस कन्नै केंद्र सरकार दे पंज मिलियन कोला बद्द मलाजमें गी फायदा मिलग।"** are shown in Table 5.8.

Table 5.8 Example of consecutive word occurances of Dogri language text

Two Pairs	Three Pairs
• केंद्रीय, मंत्रिमंडल • मंत्रिमंडल, अगले • अगले, हफ्तै • हफ्तै, लगभग • लगभग,5000 • 5000, करोड़ • करोड़, पैंशन • पैंशन, फार्मूले • फार्मूले, गी • गी, अनुमति • अनुमति, देग • देग, जिस • जिस, कन्नै • कन्नै, केंद्र • केंद्र, सरकार • सरकार, दे • दे, पंज • पंज, मिलियन • मिलियन, कोला • कोला, बद्द • बद्द, मलाजमें • मलाजमें, गी • गी, फायदा • फायदा, मिलग।	• केंद्रीय, मंत्रिमंडल, अगले • मंत्रिमंडल, अगले, हफ्तै • अगले, हफ्तै, लगभग • हफ्तै, लगभग, 5000 • लगभग, 5000, करोड़ • 5000, करोड़, पैंशन • करोड़, पैंशन, फार्मूले • पैंशन, फार्मूले, गी • फार्मूले, गी, अनुमति • गी, अनुमति, देग • अनुमति, देग, जिस • देग, जिस, कन्नै • जिस, कन्नै, केंद्र • कन्नै, केंद्र, सरकार • केंद्र, सरकार, दे • सरकार, दे, पंज • दे, पंज, मिलियन • पंज, मिलियन, कोला • मिलियन, कोला, बद्द • कोला, बद्द, मलाजमें • बद्द, मलाजमें, गी • मलाजमें, गी, फायदा • गी, फायदा, मिलग।

5.6 Results and Discussion

In this work, dictionary based technique was deployed for the detection and correction of recognition errors. The output of post-processor was manually checked at character level on five Dogri language documents i.e., the results were matched to the actual printed document. The checking was performed to evaluate the performance of the post-processor. Table 5.9 represents some of the words, corrected by applying dictionary based technique.

Table 5.9 Output of post-processing model of Dogri language text

S.No	OCR Output	Post-Processor Output	Desired Output
1.	अटालते दा समां ।	अदालते दा समां ।	अदालते दा समां ।
2.	अयने सयने	अपने सपने	अपने सपने
3.	यतंग उड़ा लो ।	पतंग उड़ा लो ।	पतंग उड़ा लो ।
4.	सभझने लगा है ।	सयझने लगा है ।	समझने लगा है ।
5.	फब जाना है ।	कब जाना है ।	कब जाना है ।
6.	दर्जने घाडल ।	दर्जने घायल ।	दर्जने घाऽल ।
7.	किसका ज़ोर वलता है	किसका ज़ोर चलता है ।	किसका ज़ोर चलता है ।
8.	पबन चला गया ।	पवन चला गया ।	पवन चला गया ।
9.	ऽोगरी पहाड़ी भाषा है	डोगरी पहाड़ी भाषा है ।	डोगरी पहाड़ी भाषा है ।
10.	तलाडच नहा लो ।	तलाऽच नहा लो ।	तलाऽ च नहा लो ।
11.	दाड लगा ले ।	दाऽ लगा ले ।	दाऽ लगा ले ।
12.	उयलब्ध करोआने लेई सरकार ।	उपलब्ध करोआने लेई सरकार ।	उपलब्ध करोआने लेई सरकार ।
13.	रवड़ा रहना ।	रबड़ा रहना ।	खड़ा रहना ।
14.	टेवनागरी लिपि ।	देवनागरी लिपि ।	देवनागरी लिपि ।

As shown in Table 5.9, the post-processor model made some corrections in the words that are wrongly classified by the recognition engine (characters at s.no 1-5,7-9,11,12,14 shown in red color). But in parallel, the post-processor partially corrected few character errors (characters at s.no 6, 10 and13 shown in sky blue color) and also in a few cases, the post-processor engine has introduced wrong characters which have otherwise been correct. Some of such cases are shown in Table 5.10. The replacements were, thus, made on the basis of word occurrence ranking i.e., the words are replaced on the basis of their occurrence frequency in a text document.

Table 5.10 Unwanted corrected output of post-processing model of Dogri language text

S.No	Recognition Output	Post-Processor Output	Desired Output
1.	किस नु मिलग	किस नु मिलेगा	किस नु मिलग
2.	तुसा दा बाग	तुसा दा बाघ	तुसा दा बाग़
3.	किस कूल दा	किस कुल दा	किस कूल दा
4.	कला च माहर है	कुला च माहर है	कला च माहर है

However, there were certain limitations still existing in the proposed approach such as a character with a dot. The characters having modifiers like '*halant*' at the bottom of character, '*bindi*' dot (.) at the top or right are mostly left unhandled by the post-processing model. The main reason behind is the low density of black pixels in the formation of '*halant*' and '*bindi*'. Mostly, it is mixed with the background noise and gets eliminated during the pre-processing stage. This limitation is expected to be solved to a certain extent as the characters which are formed using '*halant*' and '*bindi*' are very less in number.

When and wherever, these characters get identified and corrected, the approach proposed would make a signification contribution in the overall accuracy. Presently, with the help of post-processing technique an overall 0.24% increase in the accuracy has thus been obtained.

Chapter 6
Conclusion and Future Scope

In the present work testing of the proposed shape based segmentation algorithms has been done on a number of Dogri language/ Devanagari script based text documents. The experiment majorly covered around 84 base and 839 compound character classes. For explaining the algorithm details and results, statistics of ten text documents has been given in Table 3.10 of chapter 3. A number of techniques have been studied and implemented such as segmentation, feature extraction and classification. For implementation of the proposed algorithms dataset of scanned documents has been formulated. The said dataset contains grayscale text images from books, magazines, online and offline newspaper scanned at 300 DPI resolution. There were a total of around 10410 characters, including 2% to 3% compound characters. During character class formulation, it was found that the share of compound character classes is around 90.66% and that of base classes is around 9.44%. Whereas, the processing time and complexity of classification of compound characters was much more than base class characters. Therefore, this work was mainly focused on the recognition of base character classes.

During the experimentation with existing segmentation techniques, it was found that some of the base character classes had similar shapes and because of this similarity, the classifier wrongly classified some of the character classes. This issue was handled by focusing on character shapes and retention of structure during character segmentation. During the process of segmentation of characters it has been explored that some of the

characters were under, over and partially segmented. After detailed study and analysis, following reasons were identified for improper segmentation:

a) The uneven blur, noise and multiple gray levels in images, global threshold value selection, curved and closely printed text.

b) Loss of structural information during segmentation into individual characters. The structural data is importantly required for unique recognition of characters.

c) Problem in salient region detection and character segmentation methodology. During this stage, single threshold value β was chosen to segment characters.

d) Multi slant and Skewed characters.

e) Unavailability of continuous vertical white pixel gap between characters.

The above discussed anomalies contribute to the improper segmentation of characters which further results in the wrong classification of characters. In order to resolve the above issues and limitations of existing methods, a new robust method named as '***Pihu***' method; has been proposed. To determine the efficiency of the proposed method '***Pihu'***, it is applied to challenging image datasets.

Higher character recognition results have been obtained with the combination of Gradient features with SVM classifier. An accuracy rate of around 99.10% at character level was achieved. In the post processing stage N-gram technique has been used for the detection and correction of misclassified characters/ words.

6.1 Challenges Associated with Implementation

During the development of a recognition system for Dogri language, some challenging tasks have been performed and are discussed as follows:

a) As this was the first attempt of implementation of OCR for Dogri language, standard datasets were not available. The collection and scanning of Dogri language text was performed indigenously from books, newspapers, magazines etc.

b) Different shapes, style, size and non-standard font sets with huge corpus, were taken into consideration.

c) Similar looking characters increased the complexity of the study along with skewed and slanted texts.

d) Segmentation of closely packed characters and modifiers, added to the complexity.

e) Segmentation and classification of conjunct/ half consonants/ diacritic characters were also involved.

f) Recognition speed issues due to large number of classes and complexity of post processing of the classified results, were of final challenges.

6.2 Main Contributions of the Study

The major contributions of this study are briefed as follows.

a) The literature survey of all the stages of character recognition system has been done in detail. Almost, all the Indian scripts, English, Arabic and Urdu scripts were covered in the survey. The issues associated with pre-processing, segmentation, feature extraction and classification have been studied and analyzed.

b) A dataset of around two lakh Dogri language characters has been prepared from old books, magazines, newspaper and other documents. The experimental work has been carried out on two different datasets (Dataset-I and Dataset-II). The first dataset constituted characters without header lines, and the second dataset contains the same characters with

header lines. The recognition results obtained from these datasets were compared for the overall recognition accuracy of the characters, segmented using the proposed shape-based algorithm with existing techniques.

c) New segmentation algorithms have been proposed for the separation of lines, words and characters of machine printed Dogri language documents.

d) New segmentation algorithms ('***Pihu***' method) were applied to pre-detected words of Devanagari script based scene images.

e) The proposed segmentation algorithms were also tested on Hindi language documents which showed accurate output.

f) Shape oriented features have been extracted using existing feature extraction techniques (Discrete Cosine Transformation (DCT), Gradient, Gabor and Zernike Moments). The shape based features were more effective and resulted in higher character recognition accuracy.

g) For the classification of characters using extracted feature file, different classifiers such as Multilayer perceptron neural networks (MPNN), Support Vector Machines (SVM) and *k*-Nearest Neighbors (*k*-NN) were used.

h) Finally, dictionary based post processing technique was applied for the correction of errors left by the character recognition engine.

The results of the proposed shape base technique was compared with the results of earlier reported techniques. The proposed character recognition system achieved an appreciable overall recognition rate of 98.56% to 99.10% depending upon the document

image quality and classifier used. The maximum accuracy of 99.10% was obtained with the combination of Gradient feature descriptors and SVM.

6.3 Limitations

The machine printed Dogri language OCR system also have some limitations; which open avenues for further research in the area under study.

a) The character recognition system cannot perform well on images in which text is highly degraded, i.e., touching, bloody, cursive and broken characters.

b) The post-processing model needs to be trained on huge data and is expected to take a lot of processing time.

6.4 Future Scope

a) During segmentation problems like highly degraded text due to heavy noise, merged, touching and broken characters needs to be addressed.

b) The work can be extended on a multiscript character recognition system i.e., the recognition system that can identify the scripts and starts respective recognition engine.

c) The proposed method '***Pihu***' can be extended on multiscript documents in which characters are connected through header line (Shirorekha).

d) Further studies are required to develop self learning and accurate feature extraction, classification and post processing techniques.

Development of a new method to overcome the limitations outlined may be considered as a challenging future research problem. Additionally, fuzzy based feature extraction and classification techniques need to be explored in terms of accuracy improvements.

References

[1] Aggarwal, A. and Singh, C. (2016). Zernike moments-based Gurumukhi character recognition. *Applied Artificial Intelligence*. 30(5), 429-444.

[2] Aggarwal, A., Singh, K. and Singh, K. (2015). Use of gradient technique for extracting features from handwritten Gurmukhi characters and numerals. *Procedia Computer Science*. 46, 1716-1723.

[3] Aharrane, N., El Moutaouakil, K. and Satori, K. (2015). A comparison of supervised classification methods for a statistical set of features: Application: Amazigh OCR. *In Intelligent Systems and Computer Vision (ISCV), IEEE*. 1-8.

[4] Alaei, F., Alaei, A., Blumenstein, M. and Pal, U. (2016). A brief review of document image retrieval methods: Recent advances. *In IEEE International Joint Conference on Neural Networks (IJCNN), Canada*. 3500-3507.

[5] Al-Boeridi, Omar N., Ahmad SM S. and Koh S.P. (2015). A scalable hybrid decision system (HDS) for Roman word recognition using ANN SVM- study case on Malay word recognition. *Neural Computing and Applications*. 26(6), 1505-1513.

[6] Alginahi, Y. (2010). Preprocessing techniques in character recognition. *Minoru Mori (Ed.), InTech*. DOI: 10.5772/9776. 1-20.

[7] Alginahi, Y.M. (2013). A survey on Arabic character segmentation. *International Journal on Document Analysis and Recognition*. 16(2), 105-126.

[8] Alginahi, Y., Sid-Ahmed, M.A. and Ahmadi, M. (2004). Local thresholding of composite documents using multi-layer perceptron neural network. *In 47th IEEE International Mid West Symposium on Circuits and Systems, Japan*. 1209-121.

[9] Alihodzic, A. and Tuba, M. (2014). Improved bat algorithm applied to multilevel image thresholding. *The Scientific World Journal*. DOI: 10.1155/2014/176718. 1-16.

[10] Al-Khaffaf, H.S., Shafait, F., Cutter, M.P. and Breuel, T.M. (2012).On the performance of Decapod's digital font reconstruction. *In 21st International Conference on Pattern Recognition, IEEE (ICPR), Japan*. 649-652.

[11] Altuwaijri, M.M. and Bayoumi, M.A. (1994). Arabic text recognition using neural networks. *In Circuits and Systems, IEEE International Symposium (ISCAS)* 6, 415-418.

[12] Amin, A. (1998). Off-line Arabic characters recognition- The state of art. *Pattern Recognition.* 31(5), 517-530.

[13] Amin, A. (2000). Recognition of printed Arabic text based on global features and decision tree learning techniques. *Pattern Recognition.* 33(8), 1309-1323.

[14] Amin, A. and Mari, J. F. (1989). Machine recognition and correction of printed Arabic text. *IEEE Transactions on Systems, Man and Cybernetics.* 19(5), 1300-1306.

[15] Arya, S., Chhabra, I. and Lehal, G.S. (2015). Recognition of Devanagari numerals using Gabor filter. *Indian Journal of Science and Technology*. 8(27), 1-6.

[16] Badekas, E. and Papamarkos, N. (2005). Automatic evaluation of document binarization results. In *10th Iberoamerican Congress on Pattern Recognition, (CIARP), Cuba*. 1005–1014.

[17] Bag, S. and Harit, G. (2013). A survey on optical character recognition for Bangla and Devanagari scripts. *Sadhana*. 38(1), 133–168.

[18] Bansal, V. (1999). Integrating knowledge sources in Devanagari text recognition. *Ph.D. thesis. IIT Kanpur*.

[19] Bansal, V. and Sinha, R.M.K. (1999). Partitioning and searching dictionary for correction of optically read Devanagari character strings. *In 5th International Conference on Document Analysis and Recognition (ICDAR), India*. 653–656.

[20] Bansal, V. and Sinha, R.M.K. (2000). Integrating knowledge sources in Devanagari text recognition system. *IEEE Transactions on Systems, Man and Cybernetics, Part A- Systems and Humans*. 30(4), 500–505.

[21] Bansal, V. and Sinha, R.M.K. (2002a). Segmentation of touching and fused Devanagari characters. *Pattern Recognition*. 35(4), 875-893.

[22] Bansal, V. and Sinha, R.M.K. (2002b). Partitioning and searching dictionary for correction of optically read Devanagari character strings. *International Journal on Document Analysis and Recognition*. 4(4), 269–280.

[23] Bhattacharya, U., Das, T.K., Datta, A., Parui, S.K. and Chaudhuri, B.B. (2002). A hybrid scheme for hand printed numeral recognition based on a self-organizing network and MLP classifiers. *International Journal on Pattern Recognition and Artificial Intelligence*. 16(7), 845-864.

[24] Bieniecki, W., Grabowski, S. and Rozenberg, W. (2007). Image preprocessing for improving OCR accuracy. *In IEEE International Conference on Perspective Technologies and Methods in MEMS Design, MEMSTECH, Ukraine*. 75-80.

[25] Bigelow, C. (2013). Oh, oh, zero, *TUGboat*. 34(2), 168 – 181.

[26] Casey, R.G. and Lecolinet, E. (1996). A survey of methods and strategies in character segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 18(7), 690–706.

[27] Casey, R.G. and Nagy, G. (1982). Recursive segmentation and classification of composite character patterns. *In 6th International Conference on Pattern Recognition, (ICPR), Germany*. 1023-1026.

[28] Cavnar, W.B. and Trenkle, J.M. (1994). N-gram-based text categorization. *Ann Arbor MI*. 48113(2), 161-175.

[29] Chandler, D.M. (2013). Seven challenges in image quality assessment: past, present, and future research. *ISRN Signal Processing*. 2013, 905685, 1-53.

[30] Chattopadhyay, T., Jain, R. and Chaudhuri, B.B. (2012). A novel low complexity TV video OCR system. *In 21st IEEE International Conference on Pattern Recognition (ICPR), Japan*. 665-668.

[31] Chaudhuri, B.B. (2002). Towards Indian language spell-checker design. *In IEEE Language Engineering Conference (LEC'02), India*. 139-146.

[32] Chaudhuri, B.B. and Pal, U. (1997a). Skew angle detection of digitized Indian script documents. *IEEE Transactions Pattern Analysis Machine Intelligence*. 19(2), 182–186.

[33] Chaudhuri, B.B. and Pal, U. (1997b). A complete printed Bangla OCR system. *Pattern Recognition.* 31(5), 531-549.

[34] Chaudhuri, B.B. and Pal, U. (1997c). An OCR system to read two Indian language scripts- Bangla and Devanagari (Hindi). *In 4th International Conference on Document Analysis and Recognition (ICDAR), Germany*. 2, 1011-1015.

[35] Chaudhuri, B.B. and Pal, U. (1998). A complete printed Bangla OCR system. *Pattern Recognition.* 31(5), 531-549.

[36] Chaudhury, S. and Garg, R. (2008). Development of robust document analysis and recognition system for printed Indian scripts. *OCR Technical Report, Ministry of Communication and Information Technology*.

[37] Chen, Yen-Lin, H., Zeng-Wei, and Chuang, Cheng-Hung (2012). A knowledge-based system for extracting text-lines from mixed and overlapping text/graphics compound document images. *Expert Systems with Applications.* 39(1), 494-507.

[38] Chowdhury, S., Garain, U. and Chattopadhyay, T. (2011). A weighted finite-state transducer (WFST)-based language model for online Indic script handwriting recognition. *In 12th International Conference on Document Analysis and Recognition, China*. 599–602.

[39] Das, N. (2010). A script book for learning Dogri language. *Sarmal Publications, First Edition*. 1-135.

[40] Dhurandhar, A., Shankarnarayanan, K. and Jawale, R. (2005). Robust pattern recognition scheme for Devanagari script. *Computational Intelligence and*

Security (CIS), Part I, Lecture Notes in Artificial Intelligence. 3801, 1021-1026.

[41] Do, H.N., Vo, M.T., Vuong, B.Q., Pham, H.T., Nguyen, A.H. and Luong, H.Q. (2016). Automatic license plate recognition using mobile device. *In IEEE International Conference on Advanced Technologies for Communications (ATC), Vietnam.* 268-271.

[42] Dogri national language notification, last retrieved on May 2017 from *http://parliamentofindia.nic.in.*

[43] Draper, B.A., Brodley, C.E. and Utgoff, P.E. (1994). Goal-directed classification using linear machine decision trees. *IEEE Transactions on PAMI*. 16(9), 888-893.

[44] Dubey, P. and Devanand (2015). Testing and results of Hindi-Dogri machine translation system. *Indian Journal of Science and Technology*. 8(27), 22-29.

[45] Dubey, P., Pathania, S. and Devanand (2011). Comparative study of Hindi and Dogri languages with regard to machine translation. *Language In India*. 11, 298-309.

[46] Dutta, S., Sankaran, N., PramodSankar, K. and Jawahar, C.V. (2012). Robust recognition of degraded documents using character n-Grams. *In* 10th *IEEE IAPR International Workshop on Document Analysis Systems, Australia*. 130-134.

[47] Egmont-Petersen, M., de Ridder, D. and Handels, H. (2002). Image processing with neural networks- a review. *Pattern recognition*. 35(10), 2279-2301.

[48] Eikvil, L. (1993). OCR and document image analysis. *citeseer. ist. psu. edu/142042. html*.

[49] Elagouni, K., Garcia, C., Mamalet, F. and Sebillot, P. 2014. Text recognition in multimedia documents: a study of two neural-based ocr's using and avoiding character segmentation. *International Journal on Document Analysis and Recognition (IJDAR)*. 17(1), 19-31.

[50] Esakkirajan, S., Veerakumar T., Subramanyam A.N. and PremChand, C.H. (2011). Removal of high density salt and pepper noise through modified decision based unsymmetric trimmed median filter. *IEEE Signal processing letters*. 18(5), 287-290.

[51] Farrow, G., Ireton, M. and Xydeas, C. (1994). Detecting the skew angle in document images. *Signal Processing: Image Communication*. 6(2), 101–114.

[52] Fast, B.B. and Allen, D.R. (1997). OCR image preprocessing method for image enhancement of scanned documents, *U.S. Patent 5,594,815.*

[53] Frias-Martinez, E., Sanchez, A. and Velez, J. (2006). Support vector machines versus multi-layer perceptrons for efficient off-line signature recognition. *Engineering Applications of Artificial Intelligence*. 19(6), 693-704.

[54] Fu, X. and Shen, Q. (2011). Fuzzy complex numbers and their application for classifiers performance evaluation. *Pattern Recognition*. 44, 1403–1417.

[55] Gaikwad, B.P., Manza, R.R. and Manza, G.R. (2013). Video scene segmentation to separate script. In *3rd IEEE International Advance Computing Conference, (IACC), India.* 1269-1274.

[56] Garain, U. and Chaudhuri, B.B. (2002). Segmentation of touching characters in printed Devanagari and bangla scripts using fuzzy multifactorial analysis. *IEEE*

Transactions on Systems, Man, and Cybernetics, Part C- Applications and Reviews. 32(4), 449–459.

[57] Gardner, M.W. and Dorling, S.R. (1998). Artificial neural networks (the multilayer perceptron)- a review of applications in the atmospheric sciences. *Atmospheric Environment.* 32(14), 2627-2636.

[58] Georgios, V., Basilis, G. and Stavros, J.P. (2010). Handwritten character recognition through two- stage foreground sub-sampling. *Pattern Recognition*. 43(8), 2807–2816.

[59] Gonzalez, A. and Bergasa, L.M. (2013). A text reading algorithm for natural images. *Image and Vision Computing.* 31(3), 255-274.

[60] Govindan, V.K. and Shivaprasad, A.P. (1990). Character recognition-A review. *Pattern Recognition*. 23(7), 671-683.

[61] Govindaraju V., Khedekar S., Kompalli S., Farooq F., Setlur S., and Vemulapati R. (2004). Tools for enabling digital access to multilingual indic documents. *In IEEE 1st International Workshop on Document Image Analysis for Libraries, USA.* 122–133.

[62] Grafmller, M. and Beyerer, J. (2013). Performance improvement of character recognition in industrial applications using prior knowledge for more reliable segmentation. *Expert Systems with Applications*. 40(17), 6955–6963.

[63] Greenhalgh, J. and Mirmehdi, M. (2015). Recognizing text-based traffic signs. *IEEE Transactions on Intelligent Transportation Systems*. 16(3), 1360-1369.

[64] Gupta, A., Srivastava, M. and Mahanta, C. (2011). Offline handwritten character recognition using neural network. *In IEEE International on Computer Applications and Industrial Electronics (ICCAIE), Malaysia*. 102-107.

[65] Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*. 3(7-8), 1157-1182.

[66] Handley, J.C. (1998). Improving OCR accuracy through combination: a survey. *In IEEE International Conference on Systems, Man, and Cybernetics, USA.* 5, 4330 – 4333.

[67] Hassan, E., Chaudhury, S. and Gopal, M. (2014). Feature combination for binary pattern classification. *International Journal on Document Analysis and Recognition (IJDAR)*. 17(4), 375-392.

[68] Haykin, S. and Network, N. (2004). A comprehensive foundation. *Neural Networks.* 2, 41.

[69] Hong, J.Y. (2010). Digital image processing (The 2nd Edition). *Wu Han university press, China.* 114-116.

[70] Horng, S.J., Rosiyadi D., Fan, P., Wang, X. and Khan, M.K. (2014). An Adaptive Watermarking Scheme for e-government Document Images. *Multimedia Tools and Applications*. 72(3), 3085-3103.

[71] Horng, S.J., Hsu, L.Y., Li, T., Qiao, S., Gong, X., Chou, H.H. and Khan, M.K. (2013). Using sorted switching median filter to remove high-density impulse noises. *Journal of Visual Communication and Image Representation*. 24(7), 956-967.

[72] Hu, Z., Lin, J. and Wu, L. (2011). Research on OCR post-processing applications for handwritten recognition based on analysis of scientific materials. *Advances in Computer Science, Intelligent System and Environment.* 104, 131-135.

[73] Huang, C.M., Lin, Y.K. and Chang, R.W. (2014). Apply adaptive threshold operation and conditional connected-component to image text recognition. *Computer Science and Information Technology*. *2*(2), 87-94.

[74] Irum, I., Sharif, M., Raza, M. and Yasmin, M. (2014). Salt and pepper noise removal filter for 8-bit images based on local and global occurrences of grey levels as selection indicator. *Nepal Journal of Science and Technology*. 15(2), 123-132.

[75] Jacob R. (1965). Magnetic recording on pieces of mail and the like. *http://www.google.tl/patents/US3465317*

[76] Jain, A.K. and Chandrasekaran, B. (1982). Dimensionality and sample size considerations in pattern recognition practice. *In: Krishnaiah, P.R. and Kanal, L.N., Eds., Handbook of Statistics, Elsevier.* 2(39), 835–855.

[77] Jain, A.K., Duin, R.P.W. and Mao, J. (2000). Statistical pattern recognition- A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 22(1), 4-37.

[78] Jammu Prabhat. Dogri Newspaper. http:// http://www.*jammuprabhat.com*

[79] Jayadevan, R., Kolhe, S., Patil, P. and Pal, U. (2011). Offline recognition of Devanagari script- A survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C- Applications and Reviews*. 41(6), 782–796.

[80] Jayanthi, K., Suzuki, A., Kanai, H., Kawazoe, Y., Kimura, M. and Kido, K. (1989). Devanagari character recognition using structure analysis. In *IEEE 4th Region 10 International Conference (TENCON'89), India*.363-366.

[81] Jindal, K. and Kumar, R. (2016). A note on data mining based noise diagnosis and fuzzy filter design for image processing. *Computers & Electrical Engineering*. 49, 50-51.

[82] Jindal, K. and Kumar, R. (2017). A novel shape-based character segmentation method for Devanagari script. *Arabian Journal for Science and Engineering*. 42(8), 3221–3228.

[83] Jindal, M.K., Lehal, G.S. and Sharma, R.K. (2005). Segmentation problems and solutions in printed degraded Gurmukhi script. *International Journal of Signal Processing*. 2(4), 258-267.

[84] Kahan, S. Pavlidis, T. and Baird, H. S. (1987). On the recognition of printed characters of any font and size. *IEEE Transactions on PAMI*. 9(2), 274-288.

[85] Kale, K.V., Chavan, S.V., Kazi, M.M. and Rode, Y.S. (2013). Handwritten and printed Devanagari compound using multiclass SVM classifier with orthogonal moment feature. *International Journal of Computer Applications*. 71(24), 0975–8887.

[86] Kamble, P.M. and Hegadi, R.S. (2015). Handwritten Marathi character recognition using R-HOG feature. *Procedia Computer Science*. 45, 266-274.

[87] Kapoor, R., Bagai, D. and Kamal, T.S. (2004). A new algorithm for skew detection and correction. *Pattern Recognition Letters*. 25(11), 1215-1229.

[88] Kavallieratou, E., Fakotakis, N. and Kokkinakis, G. (2002). Skew angle estimation for printed and handwritten documents using the Wigner–Ville distribution. *Image and Vision Computing.* 20, 813–824.

[89] Kim, S.J., Deng, F. and Brown, M.S. (2011). Visual enhancement of old documents with hyperspectral imaging. *Pattern Recognition*. 44(7), 1461–1469.

[90] Kluzner, V., Tzadok, A., Chevion, D. and Walach, E. (2011). Hybrid approach to adaptive OCR for historical books. *In IEEE International Conference on Document Analysis and Recognition, (ICDAR), China*. 900-904.

[91] Kompalli, S., Nayak, S., Setlur, S. and Govindaraju, V. (2005). Challenges in OCR of Devanagari documents. *In 8th IEEE International Conference on Document Analysis and Recognition, South Korea*. 327-331.

[92] Kompalli, S. and Setlur, S. (2006). Design and comparison of segmentation driven and recognition driven Devanagari OCR. *In 2nd International Conference on Document Image Analysis for Libraries (DIAL'06), France*. 1-7.

[93] Kompalli, S., Setlur, S. and Govindaraju, V. (2009). Devanagari OCR using a recognition driven segmentation framework and stochastic language models. *International Journal on Document Analysis and Recognition*. 12(2), 123-138.

[94] Kour, A. and Jamwal, S.S. (2016). English-Dogri named entity recognition using statistical machine translation. *International Journal of Advanced Research in Computer Science and Software Engineering.* 6(7), 201-204.

[95] Kumar, A. and Lehal, G.S. (2016). Automatic text correction for Devanagari OCR. *Indian Journal of Science and Technology*. 9(45), 1-4.

[96] Kumar, M.K. (2008). Degraded text recognition of Gurmukhi script, *Ph. D. thesis, Thapar University, Patiala.*

[97] Kumar, M., Jindal, M.K., Sharma, R.K. (2011). k-nearest neighbor based offline handwritten Gurmukhi character recognition. *In IEEE International Conference on Image Information Processing (*ICIIP). 1-4.

[98] Kumar, S. and Rani, A. (2013). DF-LDA tree: a nonlinear multilevel classifier for pattern recognition. *Journal of Experimental & Theoretical Artificial Intelligence*. 25(2), 177-188.

[99] Kumar, S., Kumar, S., Sukavanam, N. and Raman, B. (2013). Human visual system and segment-based disparity estimation. *AEU-International Journal of Electronics and Communications*. 67(5), 372-381.

[100] Kumar, J., Ye, P. and Doermann, D. (2014). Structural similarity for document image classification and retrieval. *Pattern Recognition Letters*. 43, 119-126.

[101] Kumar, R., Sharma, R.K. and Sharma, A. (2015). Recognition of multi-stroke based online handwritten Gurmukhi aksharas. *Proceedings of the National Academy of Sciences, India Section A: Physical Sciences*, 85(1), 159-168.

[102] Kumar, R. and Singh, A. (2011). Algorithm to detect and segment Gurmukhi handwritten text into lines, words and characters. *International Journal of Engineering and Technology.* 3(4), 392.

[103] Lehal, S.G. (2001). Optical character recognition of machine printed Gurmukhi text. *Ph.D. thesis, Punjabi University, Patiala, India*.

[104] Lehal, G.S. (2013). A bilingual Gurmukhi-English OCR based on multiple script identifiers and language models. *In 4th International Workshop of Multilingual OCR, USA. 3.*

[105] Lehal, S.G. and Singh, C. (1999). Feature extraction and classification for OCR of Gurmukhi script. *Vivek*. 12(2), 2-12.

[106] Lehal, S.G. and Singh, C. (2002). A post-processor for Gurmukhi OCR. *Sadhana*, 27(1), 99-111.

[107] Lehal, S.G. and Singh, C. (2006). A complete machine printed Gurmukhi OCR system. *Vivek.* 16(3), 10-17.

[108] Lehal, G.S., Singh, C. and Lehal, R. (2001). A shape based post processor for Gurmukhi OCR. *In 6th International Conference on Document Analysis and Recognition (ICDAR), USA.* 1105-1109.

[109] Leimer, J. (1962). Design factors in the development of an optical character recognition machine. *IRE Transactions on Information Theory*. 8(2), 167–171.

[110] Li, W., Neullens, S., Breier, M., Bosling, M., Pretz, T. and Merhof, D. (2014). Text recognition for information retrieval in images of printed circuit boards. *In 40th IEEE Annual Conference of Industrial Electronics Society (IECON), USA.* 3487-3493.

[111] Li, Y., Sun, J. and Luo, H. (2014). A neuro-fuzzy network based impulse noise filtering for gray scale images. *Neurocomputing*. 127, 190-199.

[112] Likforman-Sulem, L., Darbon, J. and Smith, E.H.B. (2011). Enhancement of historical printed document images by combining total variation regularization

and non-local means filtering. *Image and Vision Computing*. 29(5), 351–363.

[113] Liu, W.Y. and Jiang, J.L. (2014). A new Chinese character recognition approach based on the fuzzy clustering analysis. *Neural Computing and Applications*. 25(2), 421-428.

[114] Liu, C.L., Koga, M. and Fujisawa, H. (2005). Gabor feature extraction for character recognition: comparison with gradient feature. *In 8th IEEE International Conference on Document Analysis and Recognition, South Korea.* 121-125.

[115] Liu, Z., Zhou, H. and Yang, N. (2010). Semi-supervised learning for text-line detection. *Pattern Recognition Letters.* 31, 1260-1273.

[116] Lorigo, L.M. and Govindaraju, V. (2006). Offline Arabic handwriting recognition- a survey. *IEEE Transactions on PAMI.* 28(5), 712-724.

[117] Louloudis, G. Gatos, B. Pratikakis, I. and Halatsis, C. (2008).Text line detection in handwritten documents. *Pattern Recognition.* 41, 3758-3772.

[118] Lu, Yi(1995). Machine printed character segmentation - an overview. *Pattern Recognition.* 28(1), 67-80.

[119] Ma, H. and Doermann, D. (2003). Adaptive Hindi ocr using generalized hausdorff image comparison. *ACM Transactions on Asian Language Information Processing.* 2(3), 193–218.

[120] Mani, N. and Srinivasan, B. (1997). Application of artificial neural network model for optical character recognition. *In IEEE International Conference on*

Systems, Man, and Cybernetics, USA. 2517-2520.

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

[122] Mayes, E., Dameran, F.J. and Mercer, R.L. (1991). Context based spelling correction. *Information Process Management*. 27, 517–522.

[123] Meshesha, M. and Jawahar, C.V. (2007). Self-Adaptable recognizer for document image collections. *In International Conference on Pattern Recognition and Machine Intelligence (PReMI), India*. 4815, 560-567.

[124] Meshesha, M. and Jawahar, C.V. (2008). Matching word images for content-based retrieval from printed document images. *International Journal of Document Analysis and Recognition (IJDAR)*. 11(1), 29–38.

[125] Mesquita, R.G., Mello, C.A. and Almeida, L.H.E.V. (2014). A new thresholding algorithm for document images based on the perception of objects by distance. *Integrated Computer-Aided Engineering*. 21(2), 133-146.

[126] Micheloni, C., Rani, A., Kumar, S. and Foresti, G.L. (2012). A balanced neural tree for pattern classification. *Neural Networks*. 27, 81-90.

[127] Miciak, M. (2008). Character recognition using Radon transformation and principal component analysis in postal applications. *In IEEE International Multiconference on Computer Science and Information Technology, Poland*. 495-500.

[128] Midday Dogri News Bulletin, Retrieved on May 2017 from *www.newsonair.com*

[129] Mill, J. and Inoue, A. (2004). Support vector classifiers and network intrusion detection. *IEEE International Conference on Fuzzy Systems, Budapest, Hungary*. 407-410.

[130] Min, Y., Cho, S.B. and Lee, Y. (1996). A data reduction method for efficient document skew estimation based on Hough transformation. *In IEEE 13th International Conference, Pattern Recognition, Austria.* 3, 732-736.

[131] Mohandes, M., Deriche, M., Ahmadi, H. and Kousa, M. (2016). An intelligent system for vehicle access control using RFID and ALPR technologies. *Arabian Journal for Science and Engineering*. 41, 3521–3530.

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

[133] Moazzam, Md.G., Rahman, T. and Uddin, M.S. (2016). Effective techniques for reduction of impulse, Gaussian and Speckle noises. *International Journal of Computer Science and Information Security*. 14(7), 45-51.

[134] Murthy, O.V.R., Roy, S., Narang, V., Hanmandlu, M. and Gupta, S. (2013). An approach to divide pre-detected Devanagari words from the scene images into characters. *Signal, Image and Video Processing*. 7(6), 1071-1082.

[135] Nagata, M. (1998). Japanese OCR Error Correction using Character Shape Similarity and Statistical Language Model. *In 17th International Conferences on Computational Linguistic* (ICCL). 922-928.

[136] Natarajan, P.S., MacRostie, E. and Decerbo, M. (2005). The bbn byblos hindi ocr system. *In Document Recognition and Retrieval, International Society for Optics and Photonics (SPIE)*. 10-16.

[137] Niblack, W. (1986). An introduction to image processing. *Prentice- Hall, Englewood Cliffs. NJ*. 115-116.

[138] Nikolaou, N., Makridis, M., Gatos, B., Stamatopoulos, N. and Papamarkos, N. (2010). Segmentation of historical machine-printed documents using adaptive run length smoothing and skeleton segmentation paths. *Image and Vision Computing*. 28(4), 590-604.

[139] Nixon, M.S. and Aguado, A.S. (2012). Feature extraction and image processing for computer vision. *Academic Press*.

[140] O'Gorman, L. (1993). The document spectrum for page layout analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 15(11), 1162-1173.

[141] Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics.* 9(1), 62–66.

[142] Paul, W.H. (1933). Handel statistical machine. U.S. Patent 1915993, http://www.google.com/patents?vid=*USPAT1915993*

[143] Pal, U. and Chaudhuri, B.B. (1997). Printed Devanagari script OCR system. *Vivek.* 10(1), 12-24.

[144] Pal, U. and Chaudhuri, B.B. (2002). Segmentation of touching characters in printed Devanagari and Bangla scripts using fuzzy multifactorial analysis. *IEEE Transactions on Systems, Man, and Cybernetics-Part C- Applications and Reviews*. 32(4), 449-459.

[145] Pal, U., Belaid, A. and Choisy, Ch. (2003). Touching numeral segmentation using water reservoir concept. *Pattern Recognition Letters.* 24(1-3), 261-272.

[146] Pal, U. and Chaudhuri, B.B. (2004). Indian script character recognition- a survey. *Pattern Recognition*. 37(9), 1887-1899.

[147] Pal, U., Jayadevan, R. and Sharma, N. (2012). Handwriting recognition in Indian regional scripts- a survey of offline techniques. *ACM Transactions on Asian Language Information Processing*. 11(4), 1-35.

[148] Pal, U., Roy, P.P., Tripathy, N. and Llados, J. (2010). Multi-oriented Bangla and Devnagari text recognition. *Pattern Recognition*. 43(12), 4124-4136.

[149] Pati, P.B. and Ramakrishnan, A.G. (2005). OCR in Indian scripts- A survey. *IETE Technical Review*, *22*(3), 217-227.

[150] Pande, H. and Dhami, H.S. (2010). Mathematical modelling of occurrence of letters and word's initials in texts of Hindi language. *SKASE Journal of Theoretical Linguistics.* 7(2), 19-38.

[151] Perez-Cortes, J.C., Amengual, J.C., Arlandis, J. and Llobet, R. (2000). Stochastic error-correcting parsing for OCR post-processing. *In 15th IEEE International Conference on Pattern Recognition, Spain*. 4, 405-408.

[152] Pant, N. and Bal, B.K. (2016). Improving Nepali OCR performance by using hybrid recognition approaches. *In 7th IEEE International Conference on Information, Intelligence, Systems and Applications (IISA), Greece.* 1-6.

[153] Ramakrishnan, K. and Evgeniy, B. (2012). Learning domain-specific feature descriptors for document images. *In 10th IEEE IAPR International Workshop on Document Analysis Systems (DAS).* 415-418.

[154] Ramteke, S.P., Shelake, R.D. and Patil, N.P. (2013). A neural network approach to printed Devanagari character recognition. *International Journal of Computer Applications.* 61(22), 0975–8887.

[155] Rana, A. and Lehal, G. S. (2015). Offline Urdu OCR using ligature based segmentation for Nastaliq script. *In Indian Journal of Science and Technology*. 8(35), 1-9

[156] Rana, A. (2017). Development of language model based optical character recognition system for Urdu script. *Ph.D. thesis, Punjabi University, Patiala, India*.

[157] Remus, R. and Rill, S. (2013). Data-driven vs. dictionary-based word n-gram feature induction for sentiment analysis. *In Language Processing and Knowledge in the Web, Lecture Notes in CS, Springer Berlin Heidelberg. 8105,* 176-183.

[158] Riseman, E.M. and Hanson, A.R. (1974). A contextual post processing system for error correction using binary N-grams. *IEEE Transactions on Computer*. c-23(5), 480-493.

[159] Roy, P.P., Bhunia, A.K., Das, A., Dey, P. and Pal, U. (2016). HMM-based Indic handwritten word recognition using zone segmentation. *Pattern Recognition*. 60, 1057-1075.

[160] Saeed, K. and Albakoor, M. (2009). Region growing based segmentation algorithm for typewritten and handwritten text recognition. *Applied Soft Computing.* 9, 608-617.

[161] Saragiotis, P. and Papamarkos, N. (2008). Local skew correction in documents. *International Journal of Pattern Recognition and Artificial Intelligence.* 22(4), 691-710.

[162] Sauvola, J. and Pietikainen, M. (2000). Adaptive document image Binarization. *Pattern Recognition*. 33(2), 225–236.

[163] Sen, W. and Ke-jian, Y. (2008). An image scaling algorithm based on bilinear interpolation with VC++. *Techniques of Automation and Applications*. 27(7), 44-45.

[164] Sezgin, M. and Sankur, B. (2004). Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging*. 13(1), 146–165.

[165] Shafait, F., Keysers, D. and Breuel, T. M. (2006). Performance comparison of six algorithms for page segmentation. *In 7th IAPR Workshop on Document Analysis Systems, New Zealand.* 368–379.

[166] Shafait, F., Keysers, D. and Breuel, T.M. (2008). Efficient implementation of local adaptive thresholding techniques using integral images. *In Electronic Imaging, International Society for Optics and Photonics*. 681510-681510.

[167] Sharma, R.K. (2011). Detection and Segmentation of Touching Handwritten Gurmukhi Script. *Ph.D. thesis. Punjabi University, Patiala*.

[168] Sharma, R.K. and Dhiman, A.S. (2010). Challenges in segmentation of text in handwritten Gurmukhi script. *In International Conference on Recent Trends in Business Administration and Information Processing (BAIP), India.* 70, 388-392.

[169] Shepard, D.H. and Cook, H. (1953). Invented a reading machine GISMO, *http://www.google.co.in/patents/US2663758*

[170] Shivakumara, P., Yuan, Z., Zhao, D., Lu, T. and Tan, C.L. (2015). New Gradient-Spatial-Structural features for video script identification. *Computer Vision and Image Understanding.* 130, 35-53.

[171] Siddiqi, I. and Vincent, N. (2010). Text independent writer recognition using redundant writing patterns with contour-based orientation and curvature features. *Pattern Recognition*. 43, 3853-3865.

[172] Singh, H., Bhama, S. and Kaur, D. (1990). Neural networks for dynamic image modeling. *Proceedings of 6th Annual Aerospace Applications of Artificial Intelligence Conference (AAAIC 90), Ohio.* 331-340.

[173] Singh, H., Chen, A., Meitzler, T. and Gerhart, G. (1995). Relative clutter metric- a new tool for image analysis. *International Conference on Automation (ICAUTO-95), India.* 12-14.

[174] Singh, J. and Lehal, G.S. (2011). Optimizing character class count for Devanagari optical character recognition. *Information Systems for Indian Languages, Communications in Computer and Information Science*. 144-149.

[175] Singh, J. and Lehal, G.S. (2014). Comparative performance analysis of Feature(S)-Classifier combination for Devanagari optical character recognition system. *International Journal of Advanced Computer Science and Applications*. 5(6), 37-42.

[176] Singh, B. Mittal, A. and Ghosh, D. (2011). An evaluation of different feature extractors and classifiers for offline handwritten Devanagari character recognition. *Journal of Pattern Recognition Research*. 6(2), 269-277.

[177] Singla, S.K. and Yadav, R.K. (2014). Optical character recognition based speech synthesis system using lab view. *Journal of Applied Research and Technology*. 12(5), 919–926.

[178] Sinha, R.M.K. (1987). Rule based contextual post-processing for Devanagari text recognition. *Pattern Recognition*. 20(5), 475-485.

[179] Sinha, R.M.K. and Mahabala, H. (1979). Machine recognition of Devanagari script. *IEEE Transactions on Systems, Man, and Cybernetics*. 9, 435-441

[180] Sinha, R.M.K., Prasada, B., Houle, G.F. and Sabourin, M. (1993). Hybrid contextual text recognition with string matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence.* 15(9), 915-925.

[181] Srinivasa, K.G., Sowmya, B.J., Kumar, D.P. and Shetty, C. (2016). Efficient image denoising for effective digitization using image processing techniques and neural networks. *International Journal of Applied Evolutionary Computation*. 7(4), 1-17.

[182] Srinivas, A.B., Agarwal, A. and Rao, R.C. (2008). An overview of OCR research in Indian scripts. *International Journal of Computer Sciences and Engineering Systems.* 2(2), 141-153.

[183] Srinivasan, K.S. and Ebenezer, D. (2007). A new fast and efficient decision-based algorithm for removal of high-density impulse noises. *IEEE Signal Processing Letters*. 14(3), 189-192.

[184] Suen, C.Y. (1979). N-gram statistics for natural language understanding and text processing. *IEEE Transaction on Pattern Analysis and Machine*

Intelligence. 1, 164–172.

[185] The constitution Act. *(2003).* Dogri national language notification, Lok Sabha Secretariat. *1189. http://parliamentofindia.nic.in*

[186] Thilagavathy, A. and Chilambuchelvan, A. (2016). An improved Fuzzy based algorithm for detecting text from images using stroke width transform. *Circuits and Systems*. 7(04), 360-370.

[187] Toh, K.K.V. and Isa, N.A.M. (2010). Noise adaptive fuzzy switching median filter for salt-and-pepper noise reduction. *IEEE Signal Processing Letters*. 17(3), 281-284.

[188] Tong, X. and Evans, D.A. (1996). A Statistical Approach to Automatic OCR Error Correction in Context. *In Proceedings of the 4th Workshop on Very Large Corpora (WVLC-4), Denmark.* 88–100.

[189] Trier, O.D., Jain, A.K. and Taxt, T. (1996). Feature extraction methods for character recognition- a survey. *Pattern Recognition*. 29(4), 641-662.

[190] Trier, O.D. and Taxt, T. (1995). Evaluation of binarization methods for document images. *IEEE Transactions On Pattern Analysis and Machine Intelligence.* 17(3), 312–315.

[191] Tu, K., Li, H. and Sun, F. (2015). A statistical learning based image denoising approach. *Frontiers of Computer Science.* 9(5), 713-719.

[192] Vashishtha, A., Agrawal, I. and Kumar, R. (2014). Devanagari handwritten numerals recognition based on invariant moments. *International Journal of*

Computer Science & Management Studies. 14(06), 2231 –5268.

[193] Verma, K. and Sharma, R.K. (2015). Performance analysis of zone based features for online handwritten Gurmukhi script recognition using support vector machine. *In 23rd International Conference on Systems Engineering, Advances in Intelligent Systems and Computing, USA*.366, 747-753.

[194] Verma, K. and Sharma, R.K. (2016). Comparison of HMM-and SVM-based stroke classifiers for Gurmukhi script. *Neural Computing and Applications*, 1-13.

[195] Wang, X., Ding, X. and Liu, C. (2005). Gabor filters-based feature extraction for character recognition. *Pattern recognition*. 38(3), 369-379.

[196] Wang, C., Zhang, F., Li, F. and Liu Q. (2010). Image spam classification based on low-level image features. *In IEEE International Conference on Communications, Circuits and Systems (ICCCAS), China.* 290-293.

[197] Wang, Y., Wu, G., Chen, G. and Chai, T. (2014). Data mining based noise diagnosis and fuzzy filter design for image processing. *Computers & Electrical Engineering.* 40, 2038–2049.

[198] Wells, C.J., Evett, L.J., Whitby, P.E. and Whitrow, R.J. (1990). Fast dictionary lookup for contextual word recognition. *Pattern Recognition*. 23, 501–508.

[199] Wikipedia, Magnetic ink character recognition (MICR code), *https://en.wikipedia.org/wiki/Magnetic_ink_character_recognition#History*

[200] Xianquan, Z., Jianzhong, Y., Tao, L. and Xuan, D. (2010). The noise

elimination of the image based on Otsu. *In IEEE International Conference on Computer Application and System Modeling (ICCASM), China.* 605-608.

[201] Xu, S. and Krauthammer, M. (2010). A new pivoting and iterative text detection algorithm for biomedical images. *Journal of Biomedical Informatics.* 43(6), 924-931.

[202] Yan, F., Zhang, H. and Kube, C.R. (2005). A multistage adaptive thresholding method. *Pattern Recognition Letters*, 26(8), 1183-1191.

[203] Yannakoudakis, E.J., Tsomokos, I. and Hutton, P.J. (1990). N-grams and their Implication to natural language understanding. *Pattern Recognition*. 23, 509–528.

[204] Yi, C. and Tian, Y. (2013). Text extraction from scene images by character appearance and structure modeling. *Computer Vision and Image Understanding*. 117(2), 182-194.

[205] Yi, C. and Tian, Y. (2014). Scene text recognition in mobile applications by character descriptor and structure configuration. *IEEE Transactions On Image Processing.* 23(7), 2972-2982.

[206] Yu, B. and Jain, A.K. (1996). A robust and fast skew detection algorithm for generic documents. *Pattern Recognition*. 29(10), 1599–1629.

[207] Zadgaonkar, A.S. and Shukla, A. (1995). Classification of Hindi Consonants According to Place of Articulation Using an Artificial Neural Network. *International Journal of Electrical Engineering Education.* 32(3), 273-275.

[208] Zipf, G.K. (2016). Human behavior and the principle of least effort- An introduction to human ecology. *Ravenio Books.*

List of Published Papers

1. **Jindal, K.**, Kumar, R. (2016). A note on data mining based noise diagnosis and fuzzy filter design for image processing. *Computers & Electrical Engineering*. 49, 50-51. **(Impact factor: 1.74)**
2. **Jindal, K.**, Kumar, R. (2017). A novel shape-based character segmentation method for Devanagari script. *Arabian Journal for Science and Engineering*. 42(8), 3221–3228. **(Impact factor: 1.09)**
3. **Jindal, K.**, Kumar, R. (2017). A new method for segmentation of pre-detected Devanagari words from the scene images- Pihu Method. *Computers & Electrical Engineering*. (**Accepted, Impact factor: 1.74**).

   https://doi.org/10.1016/j.compeleceng.2017.12.017

List of Communicated Papers

1. **Jindal, K.**, **Kumar**, R. (2017). Performance evaluation of different classifiers on shape based segmentation method for Devanagari Script. *Pattern Recognition*. (**Communicated**).
2. **Jindal, K.**, **Kumar**, R. (2017). A survey on advancements of optical character recognition for Devanagari script. *Signal Image and Video Processing*. (**Communicated**).