Machine Recognition of Printed Odiya Text

A Thesis
Submitted for the Degree of
Master of Science (Engineering)
in the Faculty of Engineering

by

Peeta Basa Pati

Department of Electrical Engineering
Indian Institute of Science
Bangalore – 560 012

DECEMBER 2002

ब्रह्मार्पणं ब्रह्म हविः
ब्रह्माग्नौ ब्रह्मणा हुतम् ।
ब्रह्मैव तेन गन्तव्यं
ब्रह्म कर्म समाधिना ॥ ४.२४ ॥

Brahman is the oblation. Brahman is the clarified butter. The oblation is poured by Brahman into the fire of Brahman. Brahman shall be realized by the one who considers everything as (a manifestation or) an act of Brahman. (4.24)

–Bhagavat Gita

Abstract

Automatic Recognition of Characters by a machine is one of the challenging problems in Artificial Intelligence. The motivation for the design of such a machine comes from the human visual system (HVS). HVS is endowed with astonishing versatility and constitutes the ultimate physical (albeit neural) realization of a pattern recognition system whose performance is not affected by geometric transformations of patterns, like characters of various styles and sizes. The prime goal of the design of such a machine is to replace the HVS in practical applications involving repetitive, monotonous tasks such as mass digitization of printed manuscripts, processing of letters and mails in postal services, job applications and banking papers.

Most research endeavors and commercial software packages focus on the Roman script. In the case of Indian scripts, the problem of automatic recognition is still a topic of considerable interest. In this thesis, an attempt to develop an *integrated* Optical Character Recognition (OCR) system for printed Odiya script is presented.

The task of automatic recognition of documents has the following major subtasks:

- **Digitization:** The process of converting the manuscript hard-copies to digital images which can be processed on a computer.
- **Preprocessing:** Preprocessing involves noise removal, skew detection and correction, binarization of the gray-valued digital image.
- **Segmentation:** This process includes separating the preprocessed image into lines, words and characters in that hierarchy.
- **Feature Extraction:** The attributes of a character, which make it distinct from other characters are called the features. The process of obtaining them from individual characters is called Feature Extraction.
- **Classification:** The extracted features are employed to make a decision on the class to which the test pattern belongs.

In this thesis, a novel binarization technique based on windows of variable width is developed and implemented. The width of the window is selected based upon the local statistics of the image. Skew in the document is detected with the help of a two level precise skew detection algorithm, employing Hough transform and statistical properties of the image. The task of segmenting individual lines from the text is accomplished employing horizontal projection vectors, while that of separating words from lines is done with the help of vertical projection vectors. The segmented words are then subjected to connected component analysis to obtain the basic characters and associated *matras*.

Identifying and extracting the right features with minimal error is one of the most important tasks in automatic recognition of documents. The ability of various types of features in discriminating Odiya characters is analyzed and the features that exhibit better discriminating capabilities are chosen for use in the recognition phase. Of the tested features, it was found that the *projection profiles* of the characters yielded better discrimination. Apart from these features, some heuristic-based features are also employed in the final classification phase.

An important requirement of pattern classifiers is their robustness to noise in the input patterns. In an attempt to design a robust classifier, various classification techniques reported in the literature are tried. These include the nearest neighbor, k-NN and modified k-NN classifiers. Apart from these *classical* pattern classification techniques, modern techniques involving Support Vector Machines (SVM's) are also employed.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

"A tree that reaches past your embrace grows from one small seed. A structure over nine stories high begins with a handful of earth. A journey of a thousand miles starts with a single step." - Confucius

---

Summary:

Optical Character Recognition (OCR) deals with the machine recognition of characters present in a document image. In the last two centuries, there have been significant efforts to develop systems which will be able to separate the region of text from a given image and recognize the contents in it. The potential of such a system to revolutionize the life of people dealing with the electronic media is immense. It is also true in the Indian context. Many researchers have put in effort to arrive at a solution and success to some extent has already been reported. Yet the design and development of a system which can tackle the problem of automatic reading of documents in a manner imitating the humans still prevails to be a far fetched dream.

1.1 Introduction

Visual perception stands as the single large obstacle on the way towards the realization of anthropomorphic systems. In order to incorporate a visually communicative link to such a system, it is also essential for it to be designed with a mechanism to decipher language scripts. A method to accomplish this goal of utmost importance is to extract individual characters from the document and to recognize them.

The human brain is a highly efficient system which performs complex associative tasks like establishing the identity of objects with their odor, recognizing objects from the sounds they make, and linking speech with visual images. In designing machines capable of performing cognitive tasks, like recognition of characters, it is appropriate to take cues from this working model. If the processing techniques employed

by the brain to execute such tasks be traced, then it would be *largely* possible to design systems capable of emulating humans.

However, the current understanding of the brain does not throw enough light on the exact mechanisms involved in perception and cognition. Though the brain areas and neural pathways involved in performing a given task can be outlined, and though the functional details of single nerve cells be explained in a precise manner, perception and cognition still stand as unsolved mysteries. Nevertheless, the innate desire of vision researchers to model the visual cognitive tasks has led to machines, which are competent enough to perform some of such tasks. Automatic recognition of characters is one among such tasks which has enthralled the human intellect and has offered a gateway to humans towards the design of script-understanding androids.

Until recently, though stones, leaves and other kinds of media have also been employed for transfer or storage of information, paper has been the prominent medium for this purpose. Even the advent of digital media has not eliminated or decreased the usage of paper as it was originally expected to. However, sincere efforts are being made to replace paper by electronic media because of the various advantages it has to offer. In such a process of mass conversion, it should be ensured, apart from the electronic preparation of current documents, that, the existing paper documents are also brought to the electronic platform.

Given the amount of information contained on paper as a result of human endeavor in various fields in the past centuries, accomplishing such a conversion with manual labor is almost impossible. In the process of automating such a process, it is essential to bring in computing machines capable of converting paper manuscripts to digital documents which can be *read and processed* by machines. The utilities provided by such systems, apart from being helpful in such a conversion process, are manifold. It is worth mentioning a few of them: postal automation, banking automation, digital library creation, and the design of reading aids for the blind in the form of text-to-speech converters.

The history of the design of Optical Character Recognition(OCR) systems dates back to as early as 1870 A.D. when Carey designed a retina scanner [1, 2]. Later, in 1890, Nipkow invented a sequential scanner which was a breakthrough, and was the fore-runner to the current, sophisticated reading machines. The design of a character recognition system by Tyurin as an aid to the visually handicapped people was a leap taken by the mankind in the field of machine recognition of printed characters. Finally, the advent of digital computers in the 1940s, provided the quantum leap towards the development of OCR technology.

The process of automatic recognition of characters is made up of the following sub-processes (Ref. Fig. 1.1):

- Scanning: The document hardcopy is converted to a digital image with the help of an optical scanner employing a process called ***digitization***.

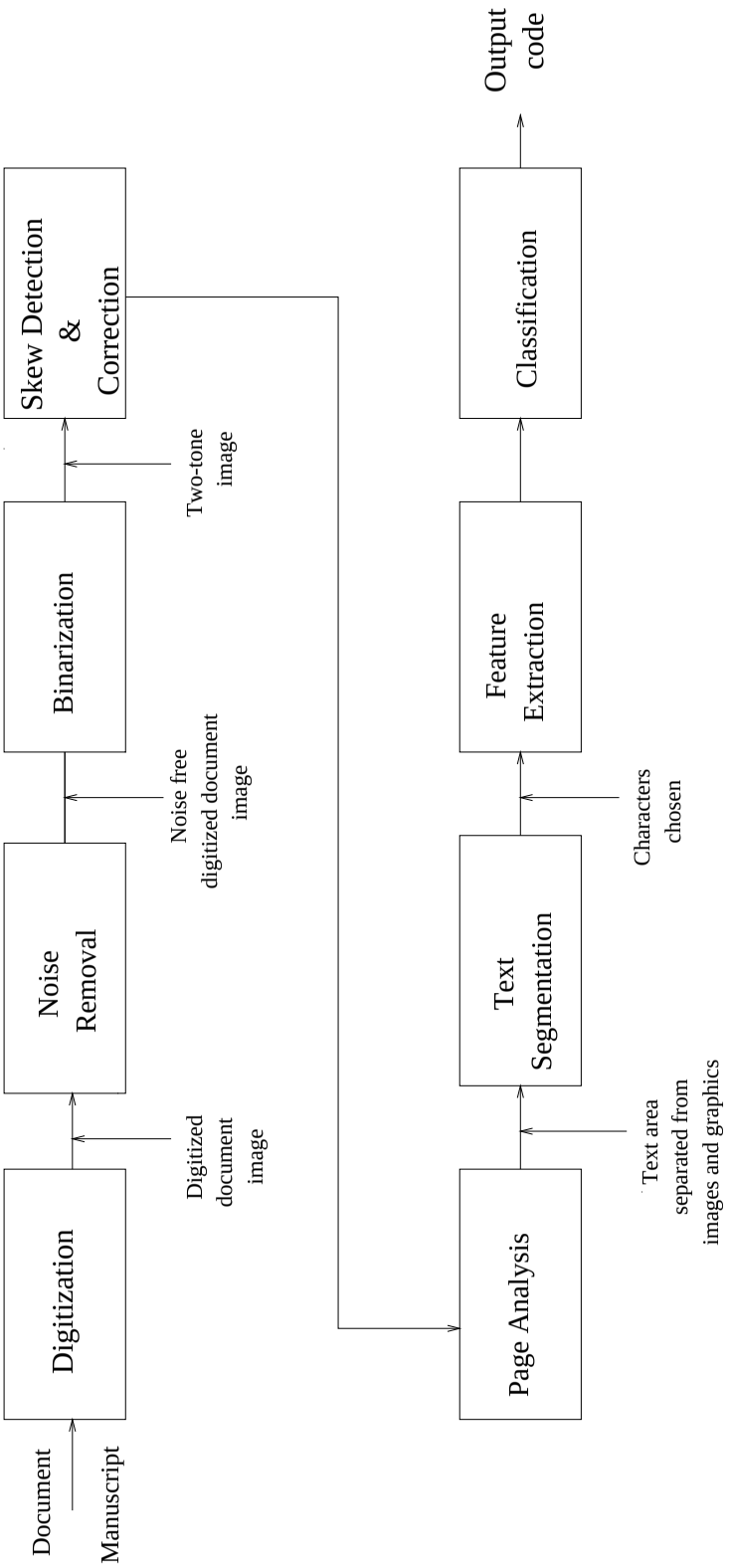


Figure 1.1: Block diagram of a typical OCR system

- Preprocessing: The digital image is passed through operations like noise filtration, skew detection and correction. These are called ***preprocessing***.
- Binarization : The conversion of the gray-valued scanned digital image to its two-tone counterpart is known as ***binarization***.
- Segmentation : Separating the objects of interest from the rest of the image is known as ***segmentation***.
- Feature Extraction : The important attributes of the character, ***features***, which differentiate it from the rest of the set, are obtained.
- Classification : On the basis of the set of features, a decision is made by a ***classifier*** as to what the given character is.

1.2 OCR work in Indian Scripts

India is a country which takes pride of being an embodiment of *unity in diversity*. The geographical and cultural vastness of the country has led to the development of more than 1500 spoken languages and 17 scripts. In a country with such rich diversity, any media of mass communication needs to transcend the language barriers. The electronic media is a prospective candidate for accomplishing this task. The conversion of all the existing documents to electronic form holds the promise of being the first step towards an *en masse* machine translation of them to other languages. This, in turn, enables the flow of thoughts and ideas across the country, thus bestowing upon it the ability to break the language barriers.

In the recent past, political bodies in India have recognized this potential of electronic technology. Prominent among the technological developments which would bring about such a change is that of processing documents automatically. A method to accomplish this is to recognize the individual characters in the given document and to later analyze them. This process of automatic recognition of the characters present in an optically-scanned document image is referred to as Optical Character Recognition (OCR). The development of OCR systems is considered as a thrust area since it has the potential to contribute to the scientific and economic advancement of the country.

Considerable efforts have been made and are still continuing towards the development of such systems. A dream of scientists working in this area is to come out with a unified OCR system, which will be able to process text of all Indian scripts. As the building blocks of such a system would be the OCR systems of individual scripts, and as such a unified system is still in its infancy, the development of OCR's in individual scripts is of significant importance.

The design of an OCR system capable of converting multi-lingual manuscripts to machine-readable code is one of the key steps in working towards the goal of machine translation. The latter opens up an

opportunity for a (nationally) unified approach in the fields of science, technology, trade and commerce. Moreover, such a system will also eliminate the need for English as the forced *lingua franca* of the country. This enables people of all walks of life to interact using their mother tongue. Further, as the documents would be processed in the official script of the state, the necessity for the knowledge of English for understanding the complex laws will be obviated.

Besides these, there are numerous applications that OCR systems have to offer which are of help in day-to-day activities of life. These include:

1. A recognizer for analyzing the digits of the number plates of vehicles in motion; this would help controlling traffic systems.

2. A reader to input text in electronic publishing.

3. Automated bill payment processing in customer-centric departments such as state electricity boards and department of Telecommunication.

4. A deciphering system that can be fit in (future) automatic vehicles for acting upon textual-signals provided on the road sides.

5. A book-keeping system, to help in examination evaluation, attendance record evaluation, marksheet reading, etc., for educational and law enforcement institutions.

Intense research and development in OCR has led to the availability of a significant number of commercial OCRs in printed Roman, Japanese, Korean, Chinese and other oriental scripts. Nevertheless, OCR's for unconstrained, hand-written character recognition are still rare. However, the availability of such commercial products for Indian scripts is still a rarity. This manifests the necessity of research and subsequent development of systems capable of processing Indian scripts. Sincere efforts towards developing such systems are being made around the country and elsewhere. Immense amounts of research work at institutions like Indian Statistical Institute, Calcutta and Center for Development of Advanced Computing, Pune have resulted in the development of OCR systems for *Devanagari* and *Bangla* scripts.

What follows is an effort to survey the endeavors of vision researchers in developing OCR's for Indian scripts in a language-by-language basis. Till date, attempts have been made to develop OCR systems for Devanagari, Bangla, Gurumukhi, Telugu, Kannada, Tamil and Odiya scripts. Of these, recognition systems for printed script in Devanagari, Bangla and Tamil have reached advanced stages of development. Efforts towards the development of OCR systems for other scripts are also on.

1.2.1 Devanagari

One of the earliest attempts towards the development of OCR in this script is by Sethi and Chatterjee [3]. In an attempt to recognize hand printed Devanagari numerals, the authors extract four primitives (*i.e.,*

the horizontal, vertical, left and right line segments) and the interconnections among these. A decision tree-based classifier is employed to recognize the numerals. Later, using a similar technique, an attempt to recognize constrained hand-printed characters has been reported by Sethi [4].

Sinha and Mahabala [5] tried to tackle the problem through the syntactic pattern recognition approach. Quantification of structural descriptions by the primitives and the interrelation between these are used for recognition in this approach. Later, the role of contextual postprocessing in Devanagari documents has been reported by Sinha [6, 7].

Chaudhuri *et al.* [1] argue that Devanagari and Bangla scripts have common graphemic features (like presence of headlines and strokes) because both possibly evolved from the same script. Further, they claim that the methodologies developed for recognition of printed Bangla characters are also applicable to printed Devanagari characters. The OCR system developed for the Bangla characters (discussed in Sec. 3.2) has been effectively adapted for use with the Devanagari script. Such an attempt has been reported to yield high recognition accuracy.

Karnik [8] effectively exploited the intersections between the headlines on the top and the droppers from the headlines for identification of individual characters in Devanagari script. After such an identification, in case of simple characters, quad-tree data structure is employed to extract relevant features. This is effectuated by continually dividing the character's area into four quadrants and counting the number of ON-pixels in each quadrant. The decision on divisions is made by appropriate threshold on the number of such ON-pixels in the given quadrant. The set of ON-pixel counts is employed as the features.

In the case of compound characters, the same feature is employed with appropriate context-based changes. On the other hand, the representation scheme for the *matras*[1] depends on the latter's spatial location and shape. In the case of matras which are placed at the bottom, the following features are also employed : (i) shape of the matra, (ii) horizontal extension length, (iii) number of horizontal lines in the matra, and (iv) length of the stroke.

Bansal and Sinha [9] have taken advantage of the structural features of Devanagari characters like (i) the presence of a straight line bar, (ii) relative positions of the straight line bars, (iii) the presence of strokes, and (iv) relative positions of the strokes. The efficacy of a knowledge-based post-processing scheme by comparing recognition accuracies with and without it has also been demonstrated by them.

The horizontal and vertical widths of the four quadrants of a character have been employed as features by Lingayath [10]. The superiority of Support Vector Machines (SVM's) over others, has been demonstrated by him after comparing the performance of various classifiers (see Sec. 3.3). The decrease in the recognition accuracy, if the characters are thinned before extracting the features, has also been

[1]Matra is the representative symbol which gets attached to a consonant when a vowel associates with the consonant to modify it

demonstrated by him. Apart from this, the possibility of achieving a high recognition rate with a Minimum Distance Classifier (MDC) if features are extracted after thinning has been put forward by him after analysis of the obtained results.

Srinivasan and Ramakrishnan [11], analyzing the applicability of Independent Component Analysis (ICA) [12, 13] to OCR in Devanagari, have shown that the independent components of characters are their strokes. They subject Devanagari hand-written characters to ICA and report a high success rate.

1.2.2 Bangla

Dutta and Chaudhuri [14] have reported an algorithm to recognize the Bangla printed as well as hand–written alphanumeric characters using curvature features. It is believed that this is the pioneering work in this script. Since the Bengali characters can adequately be represented in terms of the strokes and junctions that constitute it, an effort to exploit these from a thinned character for the purpose of recognition has been reported by them. The junction points are identified and the segments and loops of the characters are separated. These segments are then filtered to eliminate the local curvature changes. The features extracted by them from the strokes are (i) number of points of curvature maxima, (ii) number of points of curvature minima, (iii) no. of points of inflexion from -ve to +ve and from +ve to -ve curvature, and (iv) normalized positions of the maxima, minima, and points of inflexion w.r.t. stroke length. The number of strokes meeting at any junction and the latter's normalized positions in each of the strokes are also considered as components of the feature vector.

The classification work is done in two steps. The strokes and the junctions which are common over a set of characters in the alphabet set are used to group these characters at the first level with the help of a feed-forward neural network. In the second level, the relations between these features are exploited by another multi-layer network to complete the task of recognition.

Chaudhuri *et al.*[1] have contributed significantly to the development of OCR system in Bangla script. The characteristic structural features of Bangla characters are effectively used for efficient feature selection and extraction. In an effort to design a robust classification system, a tree-based decision system has been employed along with a knowledge-based error-correction mechanism.

The *shiro-rekha* (*i.e.,* the horizontal straight line that acts as the head-line for the characters) has been utilized for various preprocessing tasks such as skew detection and segmentation of lines and words. This has also been used as a cue for distinguishing Bangla and Devanagari scripts from other scripts in multi-script documents.

In order to extract the relevant features of the characters, the character is divided into three horizontal zones. This is based on the positions of the head and base-lines of the main character. The features extracted by them from the characters are: (a) width of the bounding box of the character, (b) number of border pixels per unit width of the character, (c) accumulated curvature per unit width of the character,

and (d) stroke and shape-based features. These features are made to act hierarchically in order to arrive at the final classification. Whereas the first three enable a primary grouping of the characters, the last set helps distinguishing among various classes within a given group.

Hand written character recognition in Bangla has been attempted by Bishnu *et al.* [15].

1.2.3 Kannada

Geometric moments such as zernike and pseudo-zernike moments have been extracted by Atul *et al.* [16] and a probabilistic neural network classifier has been used for recognition of printed Kannada characters. They report that the pseudo-zernike moments yield better recognition accuracy with the same classifier than the zernike moments. The histogram of the contour directions and the projection profile have also been considered as features. Classifying the features with the quadratic discriminant classifier, the recognition accuracy obtained is reported to be disappointingly low.

Ashwin [17] separates the *Aksharas* (*i.e.,* the character block templates) into its constituent components and analyzes these components as separate entities. He divides the characters into three horizontal zones and analyzes them separately (cf. Sec. 3.2). He presents an extensive study of the effect of various features on the recognition accuracy of the system. The features he has studied are the Zernike moments and some structural features.

Among Zernike moments of order up to 12, he selects 49 significant coefficients as features. Most of the Kannada characters are curved in nature and fall mostly into a single circle. Taking advantage of this property, he segments the character space into a number of radial sectors and feeds the ON-pixel counts of those regions as inputs to the classifier. Apart from this structural feature, he also has suggested the use of another, where the total number of ON-pixels across the various sectors remains constant. This task is accomplished by dividing the character space into a large number of sectors and then merging them together iteratively. Care is taken to ensure that the total number of sectors is the same as in the previous case. He claims that this modification yields an improvement of 2 to 3 percent in the recognition accuracy.

Exploring the possibility of using different classifiers like the k-NN, SVM and hierarchical SVM, he compares various feature-classifier combinations. Analyzing the results, he finally declares that the modified structural features with hierarchical classifier yields better recognition accuracy with lesser computational time.

1.2.4 Tamil

An encoded character string dictionary for recognition of Tamil characters has been proposed by Shiromoney *et al.* [18, 19]. This principle has also been utilized by Chandrasekaran *et al.* [20, 21] for recognition of multifont (printed) and constrained handwritten Tamil character recognition.

Labelled graphs have been used by Chinnuswamy and Krishnamoorthy [22] for recognition of hand-printed Tamil characters. These graphs describe the structural composition of the characters in terms of line-like primitives. A correlation-based template matching technique has been used by them, where the character under consideration is matched with the prototypes.

An online, hand-written character recognition system has been designed by Sundaresan and Sathiyakeerthi [23]. Sequence of directions and curvature are important features for online recognition of characters [24, 23]. The characters are represented by the set of x- and y-coordinates of the trajectory of the pen. Cosinusoids of the angles corresponding to this sequence form a set of features. Apart from this, the FFT and wavelet coefficients have also been used as features. A multi–layer neural network classifier acts as the decision-making unit.

A three–level hierarchical feature extraction and classification system has been attempted by Mahata [25]. Adapting the strategy of Chaudhuri *et al.* [1], the words are divided into three horizontal segments (cf. Sec. 3.2). The spatial spread of the character in the three segmental areas assigns the character into one of the four possible groups. Subsequently, the character is thinned, normalized and divided into appropriate number of rectangular sectors. This process depends on the group to which the character belongs to. Second order geometric moments are extracted from each of these rectangular blocks, which is followed by the classification with the help of a NN classifier. If this results in ambiguities, then, at the third level, DCT-based features and a NN classifier execute the final decision-making process.

1.2.5 Telugu

Around 2000 symbols have been identified in Telugu script by Rajasekaran and Deekshatulu [26]. After a careful analysis of these characters, it is found that all these characters have developed from 25 basic characters with the help of some builders and primitives. In their approach, the primitives are separated from the basic characters using a syntax-aided recognition scheme. An on-the-curve method to describe the basic characters has been tried by them. The obtained code of the basic character is compared with the dictionary of prototype codes available for all basic characters. A primitive recognition scheme is executed by a sequential template matching procedure.

Recognition of machine generated Telugu characters has been attempted by Chaudhuri *et al.* [27]. T-tuples have been used for the tracing of the skeletonized characters. The primitives, at first, are recognized by this curve tracing method, followed by the recognition of the basic character using the same algorithm.

A neural network approach has been taken by Sukhaswami *et al.* [28] for the recognition of both printed and hand–written Telugu characters. In order to achieve robustness to noise, associative memory property of Hopfield networks [29, 30] has been exploited by them. In an effort to overcome the huge memory requirement of such networks, a Multiple Neural Network Associative Memory (MNNAM) as an

alternative has also been proposed by them. The MNNAM embodies a number of small Hopfield networks connected and trained in parallel. This 2-D network consists of 12 rows and 13 columns, requiring a total of 156 neurons for the input layer. The input template is divided into a number of Scaled Windows (SW) of this size. The remainder from the division of the x-coordinate of a pixel by the window-number is used to assign a pixel (in the template) to the appropriate SW. One or all of the obtained SWs are fed to the network for recognition. The *matras* are separated from the base characters and are analyzed independently.

1.2.6 Others

There are many Indian scripts on which work has started in the recent past. Recognition of Brahmi script, an ancient Indian script, has been attempted by Shiromoney *et al.* [31]. The scheme for Tamil script (Sec. 3.4) has been adapted by them for accomplishing this task. Malayalam script has also been attempted by Chandrasekaran *et al.* [21] using their scheme for Tamil (Sec. 3.4).

Among others, the work of Lehal and Dhir [32] towards the development of an OCR system for the Gurumukhi script is significant. Similarly, work towards a Gujarati script analyzer is being carried out by Antani and Agnihotri [33]. Pati and Ramakrishnan are in the process of developing a recognition scheme for printed Odiya characters [34].

An algorithm for the identification of scripts in a multi-script document has been proposed by Chaudhuri *et al.* [35]. The characteristics of the projection profile of the lines has successfully been used for this purpose.

1.3 Odiya: Language and Script

Odiya is the native language of the state of Odisha, situated in east-central India. Odiya is listed in the 8^{th} schedule of the Indian Constitution. It is used by around 4 crore people.

Odiya, as with most Indian languages, is a phonetic language.

1.3.1 Properties of the Odiya Script

Odiya being an oriental script, a brief description of it is essential for clear understanding of the script and design of an OCR system for it. Following are some of the important properties:

1. The Odiya script, like other Indian scripts, is derived from ancient Brahmi script through various levels of transformations [1].

2. There are 12 vowels, 35 consonants and 10 numerals in the Odiya alphabet set (Ref. Fig. 1.2).

3. For almost half of the characters, there exists a vertical line towards the right end.

Figure 1.2: Odiya Alphabet

4. Some of these characters are modifications of other basic characters.

5. Vowels can occur anywhere in the word independently, unlike other Indian scripts [1].

6. Vowels combine with consonants to modify them and special symbols get added to these consonants, called as *matras*.

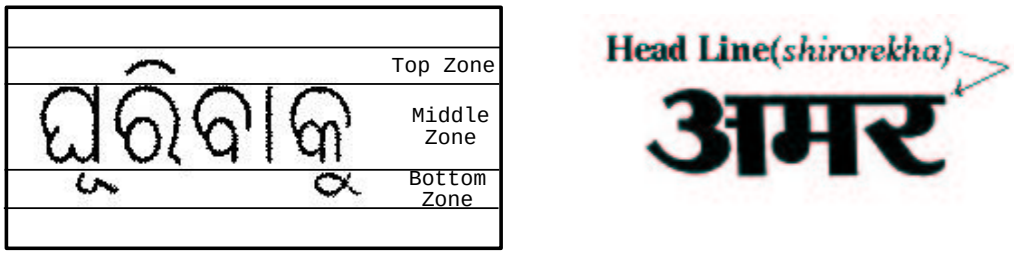
7. The matras, according to the modern practice, do not touch the consonant character while modifying the latter.

8. Combination of a consonant with other consonants is governed by a set of rules. Sometimes they modify the base characters while at other times, another symbol gets added to the base character.

9. Sometimes two consonants combine to form a completely new character known as *compound* character.

10. Each modifier symbol (vowel modifier or consonant modifier) has a specific position with respect to the base character.

11. Though three consonants can also be compounded, compounding of two consonants is more prevalent.



(a) (b)

Figure 1.3: (a) Three distinct zones of an Odiya word, (b) A Devanagari Word showing the presence of ***shirorekha*** (absent in the Odiya word).

12. Each word can be modeled as consisting of three horizontal zones (Ref Fig. 1.3). The zone in the center always has the base character; the zones above and below contain the modifiers.

13. Non-aspirated characters occur more frequently than aspirated ones.

14. Some of the consonant characters present in the alphabet set are never used independently; they are used only as modifiers.

1.4 Conclusion

In this chapter, a brief introduction to OCR and its allied processes and subprocesses has been presented. The need and motivation for development of such systems have also been mentioned. A brief review of the OCR work in Indian scripts has also been presented. The properties of the Odiya Language and the associated script have also been discussed.

Chapter 2

Digitization, Preprocessing and Segmentation

"If you steal from one author, it's plagiarism; if you steal from many, it's research." -Wilson Mizner, US playwright

---

Summary:

Digitization is the first and an essential step in the development of an OCR system. The work of preprocessing helps lay the stepping stone for isolation of individual characters which are then fed to the recognizer. Finally the segmentation isolates each and every symbol that constitutes the text portion of any document page and sets the stage ready for further action, namely, the recognition process. A systematic but brief description of these various subtasks involved and their design is provided in the sections that follow.

2.1 Introduction

The art of getting the individual blocks of patterns (*test characters*) from a digitized text document requires the implementation of quite a few number of allied operations. Once the hard copy document is digitized with the help of a scanner, the task of preprocessing starts. It includes all the processes, which remove noise, threshold the gray scale image to a two-tone one, detection of skew and its correction by inverse rotation of the image by the skew angle. Further, the processes such as separation of the text region from the rest of the items like pictures and graphics in the page, segmentation of the text region into lines, words and characters, respectively, belong to another group of subtasks known as segmentation.

Each of the subtasks involved in *pre-processing* carries importance and contributes towards the success

of the final goal. The accuracy of recognition is adversely affected if any one of these processes is undermined. So the selection of an appropriate procedure from a set of existing ones, for each of the tasks, is a challenge in itself. In this work, it is intended to pay sufficient attention to each one of these subtasks such that they perform their best and hence the outcome of the process is at the optimum.

2.2 Digitization

Digitization is the process of converting document hard-copies to digital images. It is done by scanning the hardcopy using either a flatbed scanner or hand-held scanner. For the purposes of this work, a flat bed scanner is used since it provides the advantage of stable placement, and thus avoids the distortions that arise due to hand movement in the case of hand-held scanners. A HP 3200C model flatbed scanner is used with a scanning resolution of 300 dots per inch (dpi). The hardcopies are scanned to form gray-scale images and are stored in the Tagged Image File Format (TIFF). These images are then converted to raw images for subsequent processing.

2.3 Preprocessing

The raw image obtained from the digitization process is subjected to *preprocessing*. This involves the following subtasks: (i) noise removal (ii) binarization, and (iii) skew detection and correction.

2.3.1 Noise Removal

In an ideal environment, the digital document image is an *exact replica* of the printed manuscript. However, in practice, owing to various imperfections, *noise* creeps into the digitized version of the manuscript. Typical sources of noise that can affect a digital document are:

- Inherent manuscript noise
- Non-transparent, dusty scanner bed
- Improper document placement
- Scanner-induced noise due to erroneous sampling, quantization and lighting

Noise from these sources needs to be filtered, lest it affects the recognition accuracy of the OCR system severely. This is accomplished by choosing one among the various available digital filters, such as, low-pass and median filters. Such a choice is governed by the expertise in analyzing the nature of noise and its statistical properties.

However, for purpose of the work presented in this thesis, only documents of good quality are used, where it is found that the process of binarization (see below) *nullifies* the effect of the various sources of

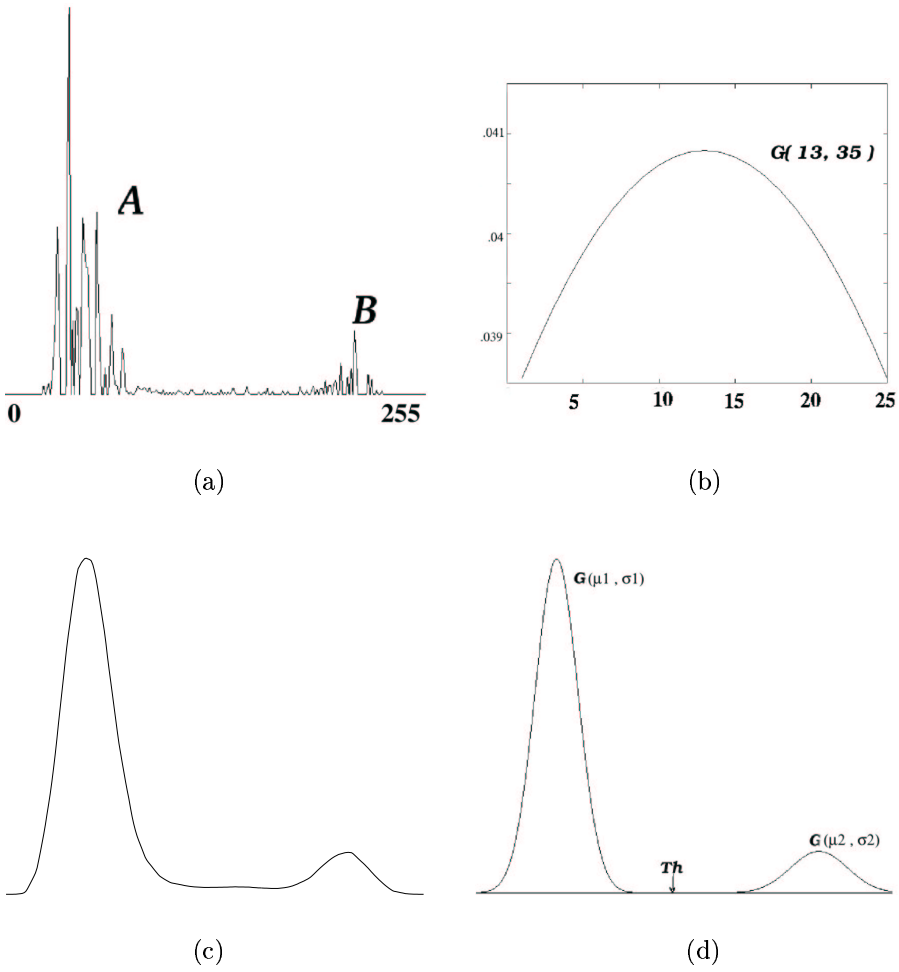


Figure 2.1: (a) Original histogram of a gray document image, (b) The Gaussian mask used to smooth (a) to obtain (c), (c) Filtered histogram of the gray document image, (d) Decomposition of the filtered histogram into its constituent Gaussian curves

noise. After experimenting with a few of the available techniques of filtration, it is found that the simple process of binarization is adequate to eliminate the noise present in the documents used.

2.3.2 Binarization

The digitized text documents contain visual information in two *clusters* of intensities. This may be observed from the gray-level histogram of any digitized text document. A typical histogram is shown in Fig. 2.1(a), where two modes **A** and **B** are clearly indicated. Such a histogram, where visual information is clustered into two *distinct* patches, is usually referred to as a **bimodal** histogram.

Separating the background of an image, characterized by a bimodal histogram, from its foreground, is achieved by thresholding the image with a suitable threshold. This produces a two-tone *binary* image. This process of converting a gray scale image to a two-tone image is referred to as ***binarization***. The

choice of this threshold value is of utmost significance in document processing.

Mathematically, if $f(x,y)$ is the input gray-scale image, $f_b(x,y)$ is the output binary image, and T_b is the threshold for binarization, then

$$f_b(x,y) = \begin{cases} p_1, & if \quad f(x,y) \leq T_b \\ p_2, & \text{otherwise,} \end{cases} \tag{2.1}$$

where p_1 and p_2 correspond to the foreground and background pixel values in the binary image.

Thresholding techniques may broadly be divided into two groups as:

- Global thresholding
- Local or adaptive thresholding

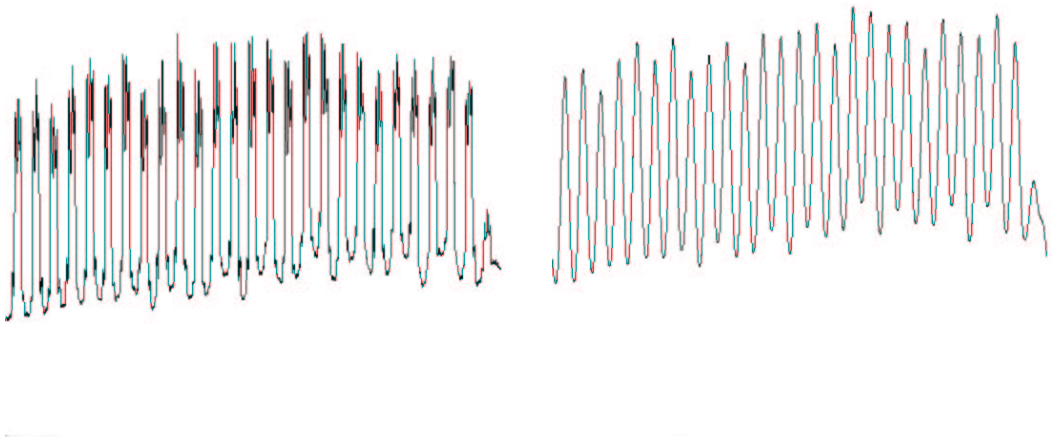
The case where T_b is constant for all x, y in the given image is called as *global* thresholding. This technique is very effective if the document quality is good. But, because of the presence of noise, a single global threshold may fail to work, thus necessitating the use of *local or adaptive* thresholds. Such a technique is useful in cases, where the nature of the noise is not known. Here, the picture is partitioned into various parts and the local properties of each sector are used for the selection of the threshold for that area. In other words, the selection of T_b depends on the local properties of segments of $f(x,y)$, rather than $f(x,y)$ as a *whole*.

For a clear text document, the histogram shows two prominent peaks [1] corresponding to black and white pixels. Different types of global and local threshold selection techniques are explained by Chaudhuri *et al.* [36] based on the histogram function of the image. Various other types of binarization techniques are also mentioned there.

In this thesis, a localized thresholding mechanism is employed, with each line of text in the document having its own threshold. The threshold is selected based on the *local* histogram of that line of text. The procedure to implement this is as follows:

1. Obtain the projection of the image onto the vertical axis (*i.e.,* the horizontal projection).
2. Convolve the projection vector with a Gaussian filter to obtain a smoothed version.
3. Find the valley points of this smoothed projection vector and declare them as the points separating subsequent lines of text in the image.
4. Consider each line to be an independent image and choose a threshold based upon the histogram of that image.

For finding the threshold value for binarization, the histogram is modeled as a sum of two Gaussians with appropriate means and variances. The mean values of the Gaussians give the two peaks of the



(a) (b)

Figure 2.2: (a) Horizontal projection vector (Original), (b) Horizontal projection vector smoothed with a truncated Gaussian mask of size=25 and σ=25.

assumed bimodal model of the histogram, while the variance values represent the respective spreads. In order to obtain the mean and variance values of the two Gaussians, an initial smoothing operation is carried out with a Gaussian filter.

The general form of a Gaussian filter is:

$$G(x) = \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \tag{2.2}$$

where, σ^2 is the variance and μ is the mean of the Gaussian or Normal distribution. In applications such as smoothing of the bimodal histogram of a typical document image, σ typically varies from 20 to 50 and μ is set to zero. In digitizing the Gaussian, a mask size of 12 to 35 can generally be employed. In the work presented in this thesis, the values of σ and the mask size are 35 and 25, respectively (Ref. Fig. 2.1(b)).

If (μ_1, σ_1) and (μ_2, σ_2) represent the mean-variance pairs of the Gaussians corresponding to the two peaks of the histogram, then the threshold T_b for that image is given as (Ref. Fig. 2.1(d))

$$T_b = \frac{\mu_1\sigma_2 + \mu_2\sigma_1}{\sigma_1 + \sigma_2} \tag{2.3}$$

2.3.3 Skew Detection and Correction

Due to improper placement of the manuscript on the scanner bed, some tilt is induced in the digital image. This angle of tilt is called as ***skew angle***. It is essential to detect and correct this angle of skew. The absence of skew correction leads to erroneous text segmentation and hence recognition accuracy falls

drastically. This process of *normalizing* the orientation of the document involves two steps: (i) precise detection of the skew angle — skew detection, and (ii) the rectification of the misalignment — skew correction.

Skew Detection

Detection of the skew angle of a document is one of the well studied subprocesses of document analysis. Various skew detection algorithms are described in [37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48]. Chaudhuri *et al.* [45] propose an algorithm which works well for most of the north Indian script documents. This is due to the presence of a straight horizontal line (see Fig. 1.3(b)) at the top of each word, popularly known as the ***shirorekha*** in the Indian context. But, since this line is absent in the Odiya script (see Fig. 1.2 and Fig. 1.3(a)), the need for a precise skew detection algorithm has led to the search for an alternative. In this thesis, the skew detection algorithm proposed by Kaushik *et al.* [25, 49] is employed and yields excellent performance. A brief description of this algorithm is as follows:

The algorithm proposed by Kaushik *et al.* invokes a hierarchical paradigm for finding the skew angle. At the first level of this process, a coarse estimation of the skew with an accuracy of $\pm\ 0.25^o$ is achieved. The steps involved in coarse estimation of the skew are:

- Run-length smoothing — It is the process that attempts to fill up the space within characters and between characters in a word. This connects all the characters of a particular word.
- Interim-line image formation — Interim line is the line mediating the background gap between two subsequent lines of text in a document image. In order to find this, find the mid-points of all pairs of consecutive words along the vertical direction (obtained after run-length smoothing) and declare them to be the constituent of the *interim line*. Form an image with these lines and call it as the **interim line image**.
- Apply Hough Transform [50, 51] on this interim line image to find a coarse estimate of the angle of skew.

The next level of the skew detection algorithm estimates the skew angle to a finer precision. The algorithm runs as follows:

- Segment the document into individual lines employing the rough estimate of the skew angle.
- Create individual images for each segmented line.
- Superimpose all these line images by aligning their centers to obtain a single image. This image is called as the *Scatter Image*.
- The principal axis of the scatter image is obtained using **principal component analysis**.

- Declare the angle that this principal axis subtends with the horizontal axis as the precise skew angle.

Theoretically, the scatter image $\mathcal{F}(x, y)$ is a representation of the distribution of ON-pixels in a given image. It formalizes this distribution as a fraction of the total area of the image under consideration, representing the estimate of the probability of getting an ON-pixel at a given spatial co-ordinate position in a line image. The probability distribution is assumed to be jointly Gaussian in both horizontal (x) and vertical (y) directions. The co-variance matrix corresponding to the ON-pixel co-ordinates of the scatter image $\mathcal{C}$ is then represented as:

$$\mathcal{C}\ \omega = \lambda\ \omega$$

where ω is an eigen vector of $\mathcal{C}$ and λ is the eigen value corresponding to ω. λ represents the variance of the assumed Gaussian distribution in the direction represented by ω.

$\mathcal{C}$ is diagonalized and the two eigen values λ_1, λ_2 and their corresponding eigenvectors ω_1, ω_2 are found. The eigen vector of $\mathcal{C}$ which is associated with the larger of the eigen values represents the direction of alignment of the text lines in the image. In other words, if $(\alpha, \beta)^T$ is an eigen vector in the eigen space, then the precise skew angle is given as:

$$\phi\ =\ \tan^{-1}\left(\frac{\beta}{\alpha}\right)\ =\ \tan^{-1}\left(\frac{\lambda_1\ -\ \mathcal{C}_{11}}{\mathcal{C}_{12}}\right)$$

where $\mathcal{C}_{11}$ and $\mathcal{C}_{12}$ are elements of $\mathcal{C}$ in usual matrix notation.

Skew Correction

Once the precise skew angle, ϕ, is found out, the document needs to be rotated in order to correct the misalignment. This is done as follows:

1. Up-sample the image by a factor of two or three. Pass it through a low pass filter for smoothing.
2. Rotate the image by ϕ in the appropriate direction. In order to avoid unfilled points (due to the inability of discrete representation to represent fractions), a bilinear interpolation scheme is employed.
3. Down-sample the rotated image back to the order of its original dimensions.

In the scheme originally proposed by Kaushik *et al.*, the binarized image is used for rotation. However, in the work presented in this thesis, the original gray level image is used for the purpose of rotation and found to yield a better performance (See Fig. 4.4(b)).

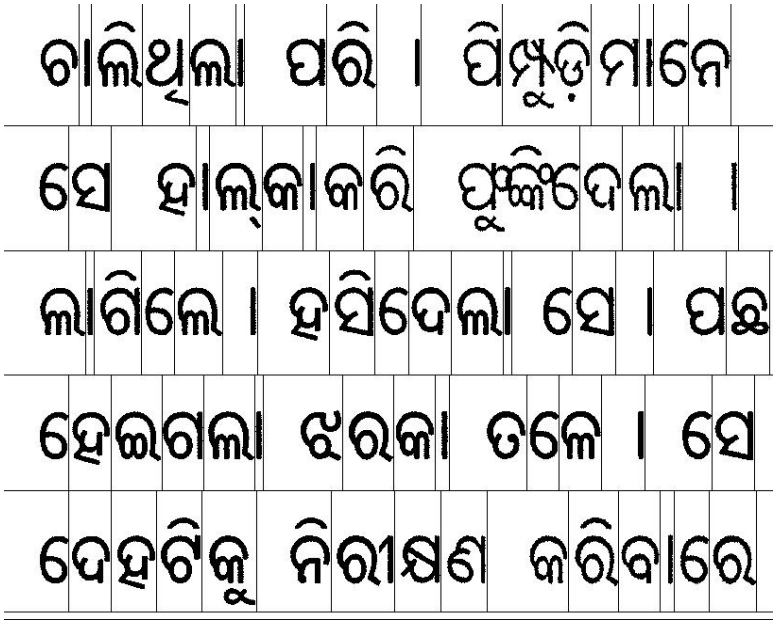


Figure 2.3: A segmented document

2.4 Segmentation

The process of subdividing the image into its constituent parts to obtain the objects of interest is called ***segmentation*** [50]. Automatic segmentation is one of the most difficult tasks in image analysis. The recognition accuracy of any OCR system is seriously affected by the effectiveness of this operation.

The process of segmentation can be subdivided into the following major tasks:

- Separation of areas of text from those of images and graphics
- Separation of individual lines of text
- Separation of individual words of a line
- Separation of characters of a word

Taking into account the influence of the quality of segmentation on the system performance, utmost care is exercised to ensure proper segmentation at each of the above mentioned steps. The ultimate goal of this process is to ensure proper separation of every character from other characters and the background.

2.4.1 Text-area Segmentation

The presence of pictures and graphics, together with text is common in various documents. It is essential to take this into account to ensure robustness and proper functioning of an OCR system. Quite a few studies have already been reported dealing with the problem of isolation of text areas in a document

image. The general belief is that such a task is independent of the script of the document as the gray-level variations in these areas are *completely* different from those in text areas.

A standard way of doing this is by employing features that are used to define and segment textures [52, 53, 54]. The assumption here is that the texture features corresponding to text areas are distinctly different from those of pictures and graphs. The normal texture features employed are (a) spectral features involving Fourier transform, (b) structural features – periodicity, directionality, Gabor features, etc., and (c) statistical features – moments and co-occurance matrices.

However, in this work, as *text-only* documents are considered for analysis, the process of separating text-areas from a given document is not addressed.

2.4.2 Line Segmentation

The act of identifying individual lines of text in the image is called as line segmentation. This is accomplished with the help of the horizontal projection. The underlying assumption here is that there is *sufficient* gap between successive lines, which sets corresponding elements in the horizontal projection vector to values close to zero. This region, where there is a *dip* in the horizontal projection vector, is selected and the mid point of this region is declared as the point which separates the lines. In cases where sufficient gap does not exist between lines, segmenting the lines becomes difficult. Here, the projection vector may be smoothed before extracting the valley points. In this work, as the valleys were observable without difficulty, smoothing the projection vector was not required.

2.4.3 Word Segmentation

Once the segmentation of the lines is over, each segmented line is considered as an individual image in order to proceed with word segmentation. To do this, the projection of a line's image onto the horizontal axis (*i.e., vertical projection vector*) is employed. A detailed study of the above vector reveals that the space gap between characters is in the range of 0.25 to 0.35 times the average width of characters and between the words is in the range of 0.8 to 1.1 times the average width of the characters in a document page.

A rough estimate of the width of the characters is obtained. Then using the information of the average intra- and inter-word spacing, the words in the line are segmented. Actually, a separate level of segmentation is not done for characters. Here a count of the number of words present in any line is done and character level segmentation is started.

2.4.4 Character Segmentation

Separating the words into individual characters (*i.e.,* basic characters, modified characters, matras, numerals or special symbols) is the final step in the segmentation process. Based upon the information

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

Figure 2.4: Imperfect segmentation of characters: (a) matra/stroke linked to one character extending to the spatial region of another, (b) due to erroneous selection of threshold for binarization, (c) error in (b) corrected by selection of appropriate threshold, and (d) inherent physical connection among characters.

on the gap between character blocks, obtained from a study of the properties of various documents, the vertical projection vector is segmented to yield individual character blocks (see Fig. 2.3).

It is not rare to find more than one foreground object in an individual character block. The reasons attributed to this are:

(I) the absence of linear separability of spatial regions due to inherent characteristics of the associated characters and matras

(II) the projection of a matra joining the projected regions of two basic characters (See Fig. 2.4(a))

(III) the strokes of one character extending to come under the spatial location of another character and hence combining the projections (See Fig. 2.4(a))

(IV) the existence of physical connectivity between two characters due to inappropriate threshold selection during binarization (See Fig. 2.4(b))

(V) the inherent existence of physical connection between characters in the manuscript itself (See Fig. 2.4(d)).

The difficulties caused by the first three situations can be handled by simple connected component analysis. For the fourth and fifth categories, the threshold level is changed to see if the characters are getting disconnected. If a change in threshold results in a disconnection, then connected component analysis is employed to segment the character block. As an instance of the effect of the binarization threshold on segmentation, see Fig. 2.4(b) and Fig. 2.4(c). Whereas Fig. 2.4(b) exhibits an inappropriate threshold, Fig. 2.4(c) shows the *correct* segmentation due to an appropriate choice. Otherwise, the error is declared to be due to improper preparation of the manuscript. In this work, the last category of errors is not considered.

Segmentation of individual character blocks completes preprocessing and sets the stage for the recognition task.

2.5 Discussion

After experimenting with various resolutions of scanning, a resolution of 300 dpi (which is twice the normal scanning resolution) has been chosen. It has been found that this resolution adds advantages at various levels of preprocessing and segmentation, in spite of its disadvantages with respect to memory requirements and computational complexity. Further, bigger bounding boxes for individual characters also help in robust extraction of features.

Within the framework of this thesis, emphasis has not been laid on the task of noise removal to the extent it should have been. This is because of the assumption on the quality of the manuscripts used in this work. However, a more rigorous study can lead to accurate design of filters which can take care of noise in document images.

Apart from this, the assumption on the absence of pictures and graphics in the image of the document restricts the applicability of the system, rendering it application-specific. However, standard texture analysis algorithms can be adapted for accomplishing this task. As this task is largely independent of the script of the text, the adaptation will be easy for documents of any script.

Though one school of thought suggests the separation of the main character from the modifier symbol, even in cases where they are connected, within the framework of this thesis, the modified characters are treated as separate symbols. This, no doubt, increases the database size but reduces the complexity of the process of segmentation.

The case where two characters are connected physically has not been taken into consideration. This is a potential area for future research where an extensive study of the properties of the characters and the script aids in the development of new algorithms. Use of knowledge can also help in such cases. Explicitly, *a priori* information about the occurrence of specific structures (such as loops and strokes) at particular positions of a word will lead to the design of methods for separation of such joint characters.

2.6 Conclusion

In this chapter, the steps involved in preprocessing a gray-scaled text document image, so as to make it *feedable* to a character recognizer, have been presented. This includes the processes of digitization, binarization, skew detection and correction and segmentation into individual characters. An improved binarization technique has been proposed and employed for obtaining a two-tone counterpart of the gray-scale image. Existing techniques, carefully chosen on the basis of their relevance to Odiya documents, have been adapted for skew detection. Segmentation of the characters from the text areas has been achieved using projection vectors and connected component analysis.

Chapter 3

Feature Extraction and Classification

"Human memory is a marvellous but fallacious instrument.... The memories which lie within us are not carved in stone; not only do they tend to become erased as the years go by, but often they change, or even increase by incorporating extraneous features." -Primo Levi, Italian-Jewish writer

---

Summary:

Features are the representative measures of a class of patterns. Once a pattern is represented by its set of chosen features, the process of decision making about its possible class value is known as classification. Both the processes of feature extraction and classification are important and together constitute the most important task known as recognition. A brief description of various types of available techniques in both groups is presented in this chapter. Also, a new approach to decorrelate the characters, before extracting features, is proposed, which improves the accuracy of recognition.

3.1 Introduction

Patterns belonging to a given class are differentiated from one another employing some characteristic *features* of them. For example, the presence of beard/mustache, color of eyes/hair, shape of nose are generally employed as distinguishing features for representation of human faces. Similarly, other pertinent distinguishing features are employed for machine recognition of patterns like objects, speech and text.

Ideally, features employed and extracted for the purpose of pattern recognition should have the following properties:

- A one-to-one correspondence between the pattern and the features extracted. That is, **uniqueness** should be guaranteed.

- Mapping from the pattern space to the feature space should be **continuous** and **robust**. In other words, there should be no discontinuity in the mapping.
- The pattern should be reconstructible from the features alone. That is, the features should form a **complete** representation of the pattern.
- The features should be **extractable** with computationally efficient procedures.

Apart from these, the struggle that a pattern recognition researcher puts towards emulating the *apparently simple* perceptual and cognitive skills of humans, motivates towards features that are made use of by the natural visual system. However, it is extremely difficult to imitate the biological pattern recognizer. The prominent reason for this is that the functional architecture of the human visual system is still a mystery.

Nevertheless, in practice, it should be noted that finding features satisfying all the above conditions is a difficult task. A complete, robust, continuous and unique representation scheme is still a dream of the pattern recognition research community. As far as the question on the use of features similar to the biological system goes, it should be noted here that, it is not *essential* to have a computational system to work exactly as its biological counterpart to achieve the goal. For example, though the binocular eye and the LASER range-finder perform the same job of depth estimation, the working mechanisms behind them are *completely* different. Hence, the features made use of by machines need not be precisely those used by humans (also see [55, 56]).

The process of **feature selection**, which involves the choice of the right features for a given problem, has to be done on a case-to-case basis governed by the structural and statistical properties of input patterns. A general rule for feature selection is to use features which exhibit maximal discrimination capability. Identifying and extracting the right features with minimal error is an important task in automatic recognition of documents. A broad taxonomy of features relevant to character recognition is as follows [18]:

- **Correlation-based features:**

  Correlation-based features involve distance measures which are computed based on a point-by-point analysis of the input characters. Obviously, this cannot make a robust feature owing to the possible presence of noise and distortion in the characters. Further, such features are not *even* invariant to affine transformations.

- **Transform-based features:**

  Transform-based features employ an orthogonal *transform* in the axes of representation in order to emphasize certain attributes of the input characters. Apart from being relatively robust to noise and distortion, such features also help in reducing the dimensionality of the feature vector. Some of

the transforms employed for this purpose are the Fourier transform, the Karhunen-Loeve transform, the wavelet transform and the Harr transform.

- **Statistical features:**

  The features derived from the statistical distribution of points and/or characters include zoning, moments, cumulants and characteristic loci. They provide high speed and low computational complexity and are invariant to changes in font face.

- **Geometrical features:**

  Geometrical (or topological) features are extracted from the shape of the input characters. Some typical features are the strokes, lines and the relative positions of strokes (either individually or in group). These features are highly tolerant to most types of distortions.

Once the representative features of a given pattern are extracted, they are used to make a decision on the class to which the pattern belongs. An ideal decision making device, the **classifier**, should have the following properties:

1. It should be robust to noise and distortions within a given class of patterns.
2. It should be sensitive to inter-class variations in the feature vectors.
3. The false-acceptance ratio[1] and false-rejection ratio[2] should be less.
4. It should be able to make a good *generalization*. In other words, the classifier should be able to extrapolate the knowledge base such that patterns which have not been already *seen* are also correctly classified.
5. The computational complexity of the recognizer should be minimal.

Though the literature points towards a number of classifiers capable of making a decision with minimal discrimination, finding an optimal classifier for a problem has to be done on a case-to-case basis. A taxonomy of the classifiers available in the literature may be given as follows [57, 58]:

- **Classifiers based on a distance measure**

  The simplest and the most generally adopted approach to classifier design is based on a measure of similarity. The patterns which are similar in some form or other are grouped into same classes. The recognition process then employs the distance measure to decide the class to which a given pattern

[1] The ratio between the number of patterns of other classes classified into a given class and the number of correct classifications is called as false-acceptance ratio

[2] The ratio between the number of patterns of a given class classified as belonging to other classes and the number of correct classifications is called as false-rejection ratio

belongs. The design of such a classifier involves the selection of (i) an appropriate metric, and (ii) the prototypes corresponding to each class, such that all sub-class variations are well represented. The most common metric employed for classification purposes is the Euclidean distance, though other choices, like the Manhattan distance also exist.

The simplest of the classifiers belonging to this group is the 1-NN rule classifier, commonly known as the Nearest Neighbor (NN) classifier. Though various improvements over this existing method has resulted in development of advanced classifiers (like k-NN, condensed NN, edited NN classifiers), the NN classifier is still used popularly with a high success rate. The advantages of using the NN classifier are manifold, and include (i) ease of use, (ii) minimal computational requirement, and (iii) non-requirement of any *a priori* information. Whereas the k-NN classifier helps improving the robustness of the basic NN classifier, condensed NN and edited NN effectively reduce the number of prototypes in the sample space.

- **Probabilistic classifiers**

  This group of classifiers takes a probabilistic route for making a decision on the class to which the given pattern belongs. In this case, *a priori* knowledge of (a) probability of class-occurrence, $p(\omega_i)$, and (b) the class conditional densities $p(\theta/\omega_i)$ corresponding to a pattern θ and classes ω_i are assumed. While the former is the probability of the occurrence of a pattern belonging to class ω_i, the latter indicates the probability that the class ω_i is that of the pattern θ.

  In most cases, the conditional probabilities are assumed to take some standard statistical distribution. Once the form of the distribution is assumed, the parameters governing its exact nature are to be determined. This is accomplished employing standard techniques, like, Bayesian learning rule, maximum likelihood estimation and Parzen window approach.

- **Classifiers based on geometric approaches**

  The third group of classifiers are based on the estimation of decision boundaries in the feature space, making use of appropriate error-minimizing criteria. Though mentions of many approaches belonging to this group have been made in the literature, the multilayer perceptron (MLP) stands apart in being a widely used method. A relatively recent addition to this group is the Support Vector Machine (SVM). SVM's are understood to be a set of optimal classifiers, where the decision boundary optimally separates the individual class spaces.

  The advantages that these types of classifiers offer include (i) automatic extraction of parameters, (ii) inherent non-linearity offering approximation to a wide range of decision boundaries, and (iii) trainability. However, the major disadvantage of such techniques is that an addition of a new prototype requires the repetition of the time-consuming training process.

- **Decision tree classifiers**

  Instead of using a single-step decision making process, robustness of the classification procedure can be improved by a hierarchical tree structured classifier [59, 58]. Each node in the decision tree corresponds to a feature or a set of features. Based upon the value of the features that a given pattern possesses, it is *allotted* to one of the various subgroups. Patterns in this subgroup are then discriminated employing another feature (or set of features), which corresponds to another node in the decision tree. The classifier employs a hierarchical, iterative selection of features which optimizes both the robustness and the computational complexity. A general rule for the selection of the features at a node involves optimizing the discriminating capability of that feature with respect to the patterns which converge at that node.

3.2 Feature Extraction

In this thesis, representative sets of the afore mentioned classes of features are examined for their effectiveness in representing Odiya characters. The features which are considered in this work are:

- Correlation-based features: image pixel value matching and a distance measure based on the *travelling wave* concept [60]. A new, efficient computational paradigm is also proposed as a substitute for this feature.
- Statistical features: geometric moments and coefficients of Legendre polynomials of various orders.
- Structural features: horizontal and vertical projection profiles, features based on rectangular and radial tiling, presence and absence of straight lines and closed loops.
- Transform-based features: DCT coefficients.

In what follows, a brief description of each of the above features along with a method to extract them is presented.

3.2.1 Image Pixel Values

Here, the image pixel values themselves are considered as features and used for comparing the test patterns with the templates[3]. This procedure of arriving at a distance measure employing a pixel-by-pixel comparison of test image and template is called as *Template matching* and is one of the most primitive methods for pattern analysis. In this case, the number of matching pixels (pixels which are ON/OFF in **both** template and test patterns) is taken as the count of the closeness of the test character with the template. The maximization of this count helps in recognizing the character.

[3]The prototypes against which a test pattern is compared are called as *Templates*

3.2.2 Symmetric Autowave Distance

In [60], Biktashev *et al.* explored the possibility of using nonlinear media as a highly parallel computation tool for image classification and recognition. They exploited the analogy between binary images and point sets and suggested a new nonlinear version of the Hausdorff metric — which they call the symmetric "autowave" distance (SAD) — for comparing two images. Using this distance measure in a OCR system, they demonstrated that the system performs more efficiently. For the use of similar ideas in pattern recognition, see [61, 62].

In extracting this distance, each binarized pattern is considered to form a wave. Considering the test pattern to be one of the waves and the referee pattern (template) to be the other, the definition of and the algorithm for extracting the SAD, as proposed by Bikatshev *et al.* [60] are as follows:

Let the set of ON-pixels in the test pattern $t(i,j)$ be represented as $\mathcal{T} = \{(i,j) : t(i,j) \text{ is ON}\}$ and that in the reference pattern $r(i,j)$ be represented as $\mathcal{R} = \{(i,j) : r(i,j) \text{ is ON}\}$. Then, the SAD $w_s(\mathcal{T}, \mathcal{R})$ is given as

$$w_s(\mathcal{T}, \mathcal{R}) = max(w_a(\mathcal{T}, \mathcal{R}), w_a(\mathcal{R}, \mathcal{T})) \tag{3.1}$$

where $w_a(\mathcal{R}, \mathcal{T})$ is the asymmetric autowave distance computed as follows:

(a) Find the points of intersection and union between the two patterns. Mathematically, the intersection and union sets are given as:

$$\begin{aligned} \mathcal{U}_{tr} &= \mathcal{T} \cup \mathcal{R} \\ \mathcal{I}_{tr} &= \mathcal{T} \cap \mathcal{R} \end{aligned} \tag{3.2}$$

(b) Instantiate a traveling wave at each point in $\mathcal{I}_{tr}$ and allow it to travel in all directions (8-neighborhood is assumed), so that it covers the whole of the difference set $\mathcal{D}_{tr} = \mathcal{U}_{tr}/\mathcal{I}_{tr}$. Explicitly, for each point $(m,n) \in \mathcal{I}_{tr}$, make

$$\mathcal{I}_{tr} = \mathcal{I}_{tr} \cup \mathcal{Z} \tag{3.3}$$

where $\mathcal{Z} = \{(i,j) : (i,j) \in \mathcal{D}_{tr} \cap \mathcal{N}^8_{mn}\}$ where $\mathcal{N}^8_{mn}$ represents the set of points in the 8-neighborhood of (m,n).

(c) The number of iterations needed by the above process to cover the entire $\mathcal{D}_{tr}$ gives a measure of the "autowave" distance between the pattern and the template. In case the process is not able to cover the

entire $\mathcal{D}_{tr}$ (due to unconnected spurious points), the point from which further wave propagation is not possible is considered as the point of convergence.

However, in the case of application to OCR, the symmetric and asymmetric autowave distances are the same because $w_a(\mathcal{T},\ \mathcal{R}) = w_a(\mathcal{R},\ \mathcal{T})$. $w_s(\mathcal{T},\ \mathcal{R})$ has the following properties:

(1) $w_s(\mathcal{T},\ \mathcal{R}) \geq 0$;

(2) $w_s(\mathcal{T},\ \mathcal{R}) = 0$ iff $\mathcal{T} = \mathcal{R}$;

The major disadvantage of this approach is its computational complexity. As the wave has to propagate from each point in $\mathcal{I}_{tr}$ covering the entire $\mathcal{D}_{tr}$, the amount of memory and time required is enormous. In an attempt to reduce the computational requirements, a modification to the above algorithm is proposed.

To this end, an union image $U(i,j)$ is defined as follows:

$$U(i,j) = \begin{cases} 2 & : \quad (i,j) \in \mathcal{I}_{tr} \\ 1 & : \quad (i,j) \in \mathcal{D}_{tr} \\ 0 & : \quad \text{otherwise} \end{cases} \tag{3.4}$$

Computationally, $U(i,j)$ may be obtained by adding the reference and test patterns. Now, considering the union set $\mathcal{U}_{tr}$ as a surface S over the $x-y$ plane, a volume V enclosed by S along with appropriate ordinates (see Fig. 3.1) is given as:

$$V = \sum_{i=1}^{R} \sum_{j=1}^{C} U(i,j) \tag{3.5}$$

where R and C are the number of rows and columns in the image, respectively.

Considering orthographic projection and an infinite light source projecting rays perpendicular to the $x-y$ plane, an area A is defined as a projection of S (see Fig. 3.1). A gives the total number of non-zero pixels present in $U(i,j)$. The ratio of the enclosed volume V to the projected area A gives the average length of the ordinates dropping from the surface under consideration. In the sequel, this ratio is referred to as the Mean Height of the Union Image (MHUI). The extraction of MHUI is computationally efficient as compared to that of the SAD (a quantitative analysis is presented in Sec. 4.3.2). The reference pattern which gives the maximal value (of this feature) corresponds to the test pattern. Under ideal conditions, the test and reference patterns match exactly, and the value of MHUI is the highest possible, because:

(a) Volume V will be maximum as all the points in $U(i,j)$ will be set to their maximum values

(b) Projected area A decreases as the dimension of $\mathcal{D}_{tr}$ is zero

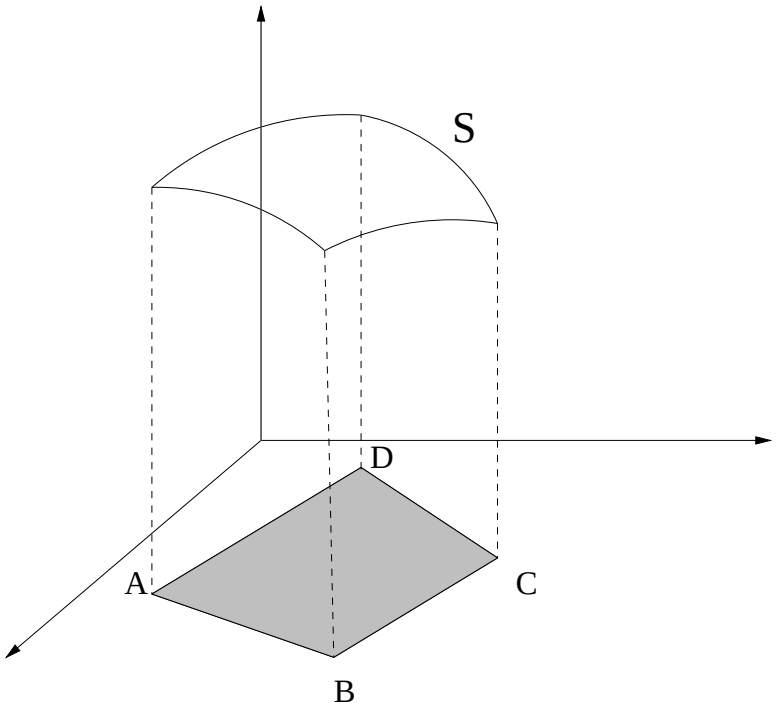


Figure 3.1: Graphic representation of the 3-D volume enclosed by surface S and the projected area indicated by ABCD

3.2.3 Projection Profiles

Each character is normalized in size to a fixed value based on its aspect ratio[4], which, in turn, is dependent on whether it is a *matra* or a normal character. Rows and columns with no ON-pixels and are adjacent to the boundaries are removed before such a normalization. This is to ensure that the character touches all the boundaries of the size-normalized matrix. The projection profiles of an image along various directions are obtained by adding pixels along that direction to form a vector. It is normal practice to consider projection vectors along the horizontal, vertical and the two diagonal directions.

In general, the projection profiles are affected by various parameters like the nature of the character (normal or bold font) and the selection of threshold. However, it is found that they work well if (i) the characters under consideration are normalized to a particular size and (ii) subjected to a subsequent normalization w.r.t. the total number of ON-pixels in the pattern. In this work, a given character is divided into 4 quadrants and from each quadrant the vertical, horizontal and the two diagonal projection profiles are together taken to form a feature vector.

3.2.4 Geometric Moments

The nth order moment M_n for a given function $f(x)$ is given as:

[4]Aspect ratio is defined as the ratio between the width and the height of the bounding box of a character

$$M_n = \mathbf{E}(x^n) = \int x^n f(x)\, dx \tag{3.6}$$

The above definition extends to two dimensions as:

$$M_{l,k} = \int \int x^l y^k f(x,y)\, dx\, dy \tag{3.7}$$

The definitions in Eqns. 3.6 and 3.7 apply to continuous functions. In the case of digital images, which may be considered as two-dimensional discrete functions, the equation for the (l,k) term of n^{th} order geometric moments is given as:

$$m_{l,k} = N_f \left(\sum_{i=0}^{R} \sum_{j=0}^{C} i^l j^k I(i,j) \right) \tag{3.8}$$

where $l, k \in \{0, 1, 2, \ldots, n\}$ and $l + k = n$; R, C are the number of rows and columns of the given image and N_f is a normalization factor given as:

$$N_f = \left\{ \sum_{i}^{R} \sum_{j}^{C} (i^n + j^n) \right\}^{-1} \tag{3.9}$$

where n is the order of the moment.

3.2.5 Coefficients of Legendre Polynomials

The Legendre moments of a two-dimensional function $f(x,y)$ are defined as:

$$\Lambda(n,m) = \frac{(2n+1)(2m+1)}{4} \int_{-1}^{1} \int_{-1}^{1} P_n(x) P_m(y) f(x,y)\, dx\, dy \tag{3.10}$$

where the kernel function $P_n(x)$ represents the Legendre polynomial of order n along the x-axis. The values of x and y range from -1 to 1.

Discretizing the above continuous function for its applicability to digital images,

$$\lambda(n,m) = \frac{(2n+1)(2m+1)}{4} \sum_{i}^{R} \sum_{j}^{C} P_n(x_i) P_m(y_j) I(i,j) \tag{3.11}$$

where $I(i,j)$ represents the digital image, x_i and y_j are the values of x and y corresponding to the pixel with co-ordinates i and j. This notation is used in order to differentiate the ranges of (i,j) and (x_i, y_j); whereas $1 \leq i \leq R$ and $1 \leq j \leq C$, $-1 \leq x_i, y_j \leq 1$. So, a corresponding mapping has to be done for a conversion from the (i,j) co-ordinates to (x_i, y_j) co-ordinates.

The most important properties of the Legendre polynomial $P_n(x)$, which are used in defining the basis polynomials and computing the coefficients are:

(a) The recursive relation:

$$\begin{aligned} P_0(x) &= 1 \\ P_1(x) &= x \\ P_n(x) &= \frac{1}{n}\left((2n-1)\,x\,P_{n-1}(x) - (n-1)\,P_{n-2}(x)\right), \quad \forall\, n \geq 2 \end{aligned} \tag{3.12}$$

(b) The integral relation:

$$\int P_n(x)\,dx = \frac{P_{n+1}(x) - P_{n-1}(x)}{2n+1} \tag{3.13}$$

(c) The orientation of the Legendre moment can be adjusted by appropriate choice of m and n.

Legendre moments, as defined above, are orthogonal moments and hence are suitable as features for character recognition. This is because of the inter-feature independence they offer. Since Legendre polynomials are valid over a range of $-1 \leq x \leq 1$, the character matrix coordinates are mapped to it. The pseudo values of i and j, x_i and y_j respectively, are used for this purpose. Though the functional form of the moments can be varied by changing m and n, in the presented work, only the case where $m = n$ is only considered.

3.2.6 Rectangular- and Sector-based methods

Instead of computing the above features (the projection profiles, Legendre moments and the Geometric moments) on the entire image in a global perspective, a more localized approach can also be employed. This may be accomplished in two ways:

1. Dividing the image into rectangular blocks.

2. Dividing the image into sectors and tracks (see Fig. 3.2).

Apart from extracting the afore mentioned features from the individual blocks, the ON-pixel count in that region is also considered as a feature. In order to accomplish this, in the former case, the entire character is divided into blocks of 6×6 and a count is obtained. In the latter case, the character pattern is divided into a number of sectors. The geometric centre of the pattern is found and the ON-pixel farthest from the centre is detected. The length of the line connecting the center and this pixel is divided in such a way that each segment gives rise to an annular region of same area as that of the central circle. Then each ring is divided into 12 sectors, each sweeping a 30^o angle for obtaining a count of ON-pixels.

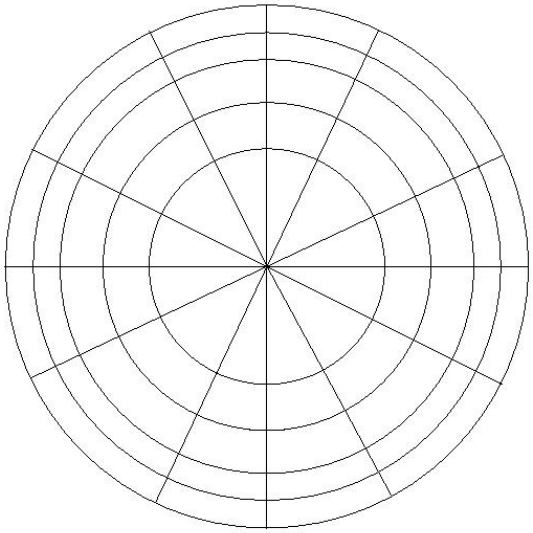


Figure 3.2: Frame showing radial division

3.2.7 Structural Features

Characters in the Odiya script may be grouped based on the presence or the absence of (a) a vertical straight line, and (b) a loop. It is observed that both the above features are such that they divide the set of Odiya characters into two almost equal groups. Taking this as a cue, characters may be grouped as follows:

- Characters which have both vertical line and loop
- Characters which have a vertical line but no loop
- Characters which have a loop but no vertical line
- Characters which have neither a loop nor a vertical line

Such a grouping helps in designing a hierarchical classification system.

3.2.8 The Discrete Cosine Transform

The Discrete Cosine Transform (DCT) is a transformation technique like the Discrete Fourier Transform, involving spectral decorrelation of the input data. For an image $I(i,j)$, the DCT coefficient matrix $B(k,l)$ is given as follows:

$$B(k,l) = \sum_{i=0}^{R-1} \sum_{j=0}^{C-1} I(i,j) \cos\left(\frac{\pi k(2i+1)}{2\,R}\right) \cos\left(\frac{\pi l(2j+1)}{2\,C}\right) \tag{3.14}$$

where R and C are the number of rows and columns of the image matrix; k and l are the frequency indices along the i and j directions respectively.

3.3 Classification

Once the appropriate set of features are extracted from the test pattern, they are matched with the same set of features extracted from the reference patterns. The final decision on the class to which the test pattern belongs is made by finding the reference pattern having maximal resemblance to it.

The design of such a classifier has a profound influence on the performance of the OCR system. Though the design has to be made on a case-to-case basis, the literature tends to agree on the notion that a hierarchical tree-based classifier is better suited to pattern classification. In such classifiers, the decision-taking process is distributed among various levels, where each level is, generally, dependent on different features. The final decision is a weighted combination of these individual outcomes. Since the relative weights and the hierarchy significantly affect the performance, they are to be chosen with utmost care.

What follows is a brief account of the various classifiers employed.

3.3.1 Nearest Neighbor (NN) Classifier

This classifier makes use of a simple feature matching technique to declare the class of the test pattern. The algorithmic listing of the procedure is as follows:

1. Compute the Euclidean distances between the test (feature) vector and each of the reference vectors.
2. Find the reference vector which yields the minimum distance.
3. Declare the class of the test pattern as that of the reference vector found in Step 2.

3.3.2 k-Nearest Neigbour (k-NN) Classifier

The k-NN classifier is a generalization of the NN classifier. NN classifier relies on the nearest neighbor for its final decision. The possibility of this yielding an erroneous decision is high because the obtained single neighbor may be an outlier of some other class. In order to avoid this, and to improve the robustness of the approach, the k-NN classifier works with k patterns in the neighborhood of the test pattern in the feature space. The algorithmic listing of the decision making process is as follows:

1. Compute the Euclidean distances between the test (feature) vector and each of the reference vectors.
2. Sort the reference vectors based on the distances and choose the least-distant k patterns.

3. Find the representation of various classes in this $k-$neighborhood space. This is found by counting the number of patterns belonging to a given class, within the $k-$neighborhood space.

4. Declare the test pattern to be belonging to that class which has maximal representation.

3.3.3 Modified k-Nearest Neighbor (mk-NN) Classifier

The mk-NN classifier is a modified version of the k-NN classifier. In the k-NN classifier, though a search in the k-neighborhood drives the classification, the distance of a template from the test character does not play any role. The k-NN classifier acts on a *population encoding* kind of strategy where *only* the representation from each of the classes plays a crucial role, with equal weights assigned to each of the representations. In an effort to bring about a weighted representation, the mk-NN classifier associates a distance-based weight with each prototype member in the k-neighborhood. This weighted representation schedule then drives the classification process. The weight is calculated as follows:

Let $\mathcal{S} = \{s_1, s_2, \ldots, s_k\}$ be the set of reference patterns in the $k-$neighborhood of the test pattern (in the feature space), sorted in the increasing order of their distances from the test pattern. Let $\mathcal{X} = \{x_1, x_2, \ldots, x_k\}$ be their respective distances from the test pattern, where x_1 is the minimum and x_k is the maximum distance. Let $\mathcal{W} = \{w_1, w_2, \ldots, w_k\}$ be a weight set with w_i as the weight assigned to pattern s_i (based on its distance from the test feature vector), given as:

$$w_i = \frac{x_k - x_i}{x_k - x_1} \tag{3.15}$$

The algorithm is given as:

1. Compute the Euclidean distances, x_i, between the test (feature) vector and the reference vectors.

2. Sort the reference vectors based on the distances and choose the least-distant k patterns.

3. Calculate weight w_i associated with each reference pattern s_i in the $k-$neighborhood.

4. Find the class which contributes maximum weight in the neighborhood. Declare the test pattern to be belonging to this class.

3.3.4 Support Vector Machines

Amongst the discriminant approaches for classification, one of the most recent is the Support Vector Machine (SVM) [63], where the optimal hyper-plane decides the separation between individual classes of patterns. The creation of a unique model to represent a class, derived by training the model with prototypes of each class, aids in maximization of the correct classification rate.

The concept of optimal hyper-plane was initially proposed by Vapnik [64]. The optimality is in the following sense: the average distance between the hyper-plane and the closest training points (on both

sides) is maximal. Whereas data with linear separability may be analyzed with a hyper-plane, linearly non-separable data is analyzed with *Kernel functions* (defined below) of other types such as polynomials (of higher order), Gaussian and tan-sigmoid. It may be noted here that a linear case yields a hyper-plane.

A SVM classifier is typically defined with the *Kernel* and the vectors which act as *supports* to the decision surface. Once the functional form of the kernel is chosen, the problem of obtaining the support vectors is solved within an optimization framework. The output of a SVM is a linear combination of the training examples projected on to a high dimensional feature space through the use of *kernel* functions.

SVM Torch, a software developed by a group at IDIAP, Switzerland, [65] is used in this work.

3.4 Decorrelation with Superquadrics

A close analysis of Odiya script reveals the existence of high redundancy among the various characters. Removal of this redundancy will help in arriving at features which offer better discriminating capabilities. This is because the removal of the correlation among the characters (decorrelating the characters) will emphasize those regions in them which are unique to them. In this section, a *new* idea to decorrelate patterns using appropriate mathematical model of the ensemble behavior of the characters in the script is presented. The model employs the *superquadrics* family of curves.

3.4.1 Superquadrics

Superquadrics are a family of parametric shapes which are extensively used for the purpose of modeling basic shapes and solids. They carry the inherent potential to model a diverse range of objects and are intrinsically symmetric about their co-ordinate axes. Several researchers have used superquadrics for modeling various objects, motion analysis and object recognition [66, 67, 68, 69]. Advantages of employing a superquadric model include its flexibility in describing a large number of shapes and the compactness associated with its representation. However, it can't be used for modeling many naturally occurring objects because of its assumption of intrinsic symmetry.

Mathematically, a superquadric is given by the equation

$$\left|\frac{x}{a}\right|^{\frac{2}{\epsilon}} + \left|\frac{y}{b}\right|^{\frac{2}{\epsilon}} = 1 \tag{3.16}$$

where (a, b) represents the center of the superquadric in the $x-$ and $y-$coordinate system; ϵ is the modulating factor which controls the shape of the superquadric.

The solution to the above equation, in its parametric form is given as:

$$x = a\,\mathrm{sgn}(\cos\theta)\,|\cos\theta|^{\epsilon}$$

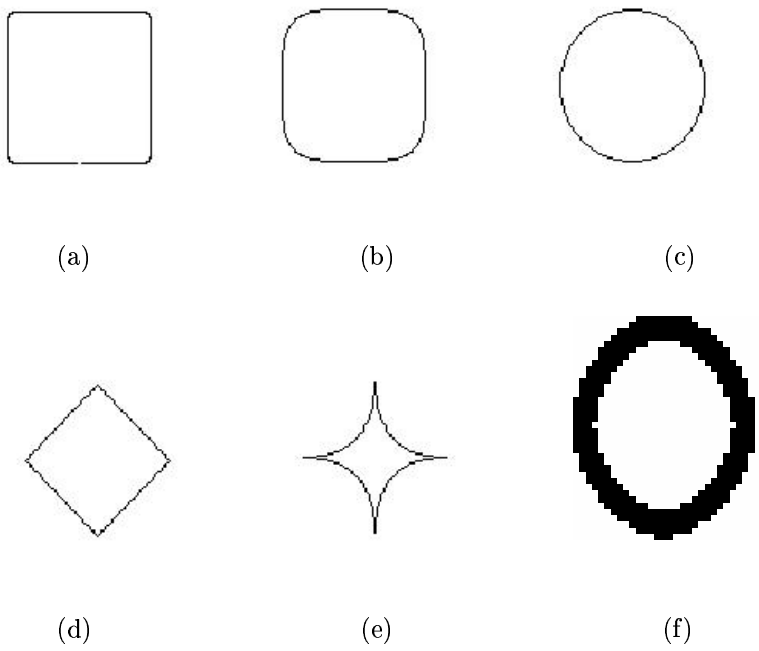


Figure 3.3: Superquadrics : (a) $\epsilon = 0.1$, (b) $\epsilon = 0.5$, (c) $\epsilon = 1$, (d) $\epsilon = 2$, (e) $\epsilon = 4$, and (f) $\epsilon = 1.13$ (one used for redundancy removal)

$$y = b\ \mathrm{sgn}(\sin\theta)\ |\sin\theta|^{\epsilon} \tag{3.17}$$

where $-\pi \leq \theta \leq \pi$. The exponent ϵ is called as the squareness parameter. The various shapes of superquadrics for different values of ϵ are shown in Fig. 3.3.

3.4.2 Redundancy modeling using superquadrics

Various instances of a signal, generally, have some correlation between them contributing to the redundancy in representation. It is advantageous to remove this redundancy, so that the analysis of the signal is more efficient. With regard to the human visual system, it has been shown that the neurons process only the non-redundant information. Explicitly, the neurons consider only the edge information, rather than considering the redundant, unvarying regions in the scene [70].

A similar route may be taken to process optical characters by removing the redundant information. This helps in efficient recognition in three ways: (a) the distinctive regions of the characters which make it unique are emphasized, (b) since many of the ON-pixels contributing to the redundancy in information are removed, the magnitude of the feature vector decreases, and (c) the correlation among the extracted features decreases.

Theoretically, consider the set of characters in the reference library to form a stochastic process $\mathcal{F}(x, y; \zeta)$, where (x, y) represent the spatial coordinates of the (character) images, while ζ is the class

value of the corresponding character.

The function $\mathcal{F}(x, y; \zeta)$ cannot be described in terms of a finite number of parameters, if all the associated independent variables are randomly varying (with the distribution also being not known)[71]. In stark contrast, if all these variables follow some mathematical certainty, the function can be described completely. The case of random arrival of test patterns lies somewhere in between these two extremes. This is because, though ζ is not predictable with high accuracy, it is known that (i) it belongs to one of the known set of values, (ii) it depends on x, y. The distribution and the statistics of ζ are also known.

Under such conditions, the nature of the process is determinable to a large extent. If a large number of realizations of the process $\mathcal{F}(x, y; \zeta)$ are available, it is possible to estimate the average value of the signal, $\eta(x, y)$, at each spatial location (x, y).

$$\eta_N(x, y) = \frac{1}{N} \sum_{i}^{N} \mathcal{F}(x, y; \zeta) \tag{3.18}$$

where $\eta_N(x, y)$ is a rough estimate of $\eta(x, y)$ and $\lim_{N \to \infty} \eta_N(x, y) = \eta(x, y)$. If N is large, thus representing a large ensemble of prototypes, a fair estimate of $\eta(x, y)$ can be made.

The estimate obtained from the ensemble of the reference patterns reveals that the redundancy in the characters (of the set of alphabets) is near-elliptical in shape. It is also observed from Fig. 1.2 that there exists a high factor of correlation between the characters, resulting in confusion among them. Though the redundancy among the characters helps in the formation of similarity-clusters, it does not aid in discriminating characters belonging to the same cluster. The removal of this redundancy, as mentioned earlier, will emphasize the distinctions of characters within the same cluster. Here, a superquadric-based scheme is used to model the redundancy in representation mathematically. That is the estimate $\eta_N(x, y)$, which is found to be near-elliptical, is approximated using a best-fit superquadric. This model is then employed to reduce the redundancy in the reference set, which results in better classification accuracy.

For Odiya script characters, it is found that a value of $\epsilon = 1.13$ gives the best approximation to $\eta_N(x, y)$. The superquadric with $\epsilon = 1.13$ with a width of 4 pixels is shown in Fig. 3.3(f)

3.5 Conclusion

In this chapter, a theoretical review of some of the existing and/or implemented techniques of feature extraction and classification has been presented. Features belonging to various groups have been presented: (i) Gross structural features: projection profiles, pixel counts, (ii) Moment-based features: Legendre moments, geometric moments, (iii) Correlation based features: image pixels, symmetric autowave distance, etc. A new feature, based on an analysis of the enclosed volume and the projected surface of the union of the template and the test image, is proposed. A new approach, to remove redundancy from patterns, for better recognition has also been proposed.

Chapter 4

Results and Discussion

"Those who admire modern civilization usually identify it with the steam engine and the electric telegraph. Those who understand the steam engine and the electric telegraph spend their lives in trying to replace them with something better." -Bernard Shaw

---

Summary:

An analysis of the applicability of various feature-classifier combinations is presented in this chapter. The effectiveness of the proposed techniques for skew correction and binarization is also shown. The proposal for removing the redundant points from the patterns is found to be beneficial in improving the recognition accuracy.

4.1 Introduction

This chapter presents the results corresponding to different techniques employed for carrying out the various subtasks involved in an OCR system. As subtasks, each of them yield partial results which are fed to the next stage in the hierarchy. Based on the analysis of the outputs at each stage, inferences and implications of the observed outcomes are presented as a measure towards the design of a better OCR system. An analysis of the pros and cons of various approaches and parameter dependence of each subtask is also presented. The OCR scheme was implemented in C++ on a Pentium-III (500 MHz clock speed) machine. The results related to the preprocessing stage and at the feature extraction and classification stages are given in the following sections.

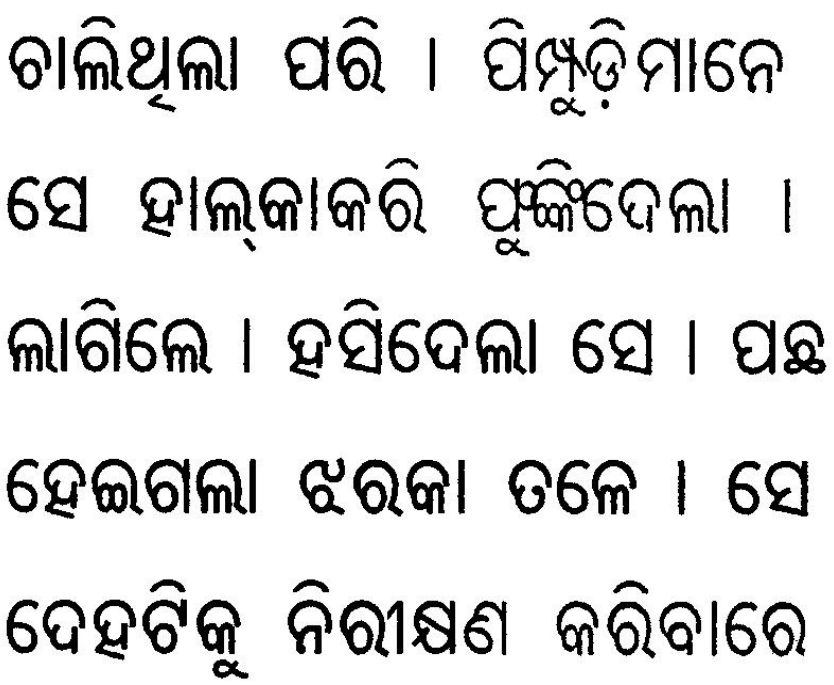


Figure 4.1: Binarization achieved by global thresholding

4.2 Preprocessing

The preprocessing stage of document analysis involves (i) noise removal, (ii) binarization and (iii) skew detection and correction (cf. Sec. 2.3). As documents of good quality are only employed, noise removal has not been addressed elaborately in the work presented. With respect to binarization and skew analysis, however, efficient modifications have been suggested in this thesis.

4.2.1 Binarization

In the process of binarization, a local or adaptive binarization scheme has been proposed. Here, the text page is roughly segmented into lines. Each line is then treated as a separate image and its histogram is studied to select the threshold for converting the gray image to a corresponding two-tone one. Fig. 4.1 shows the output of global binarization and Fig. 4.2 shows the corresponding output of the proposed method.

The advantage of local thresholding over global thresholding can be clearly observed by comparing the above mentioned figures. The reason for the failure of global thresholding is the uneven distribution of contrast between the foreground and background pixel values. On the other hand, it is observed that local thresholds take local properties (distribution of gray values) into account and hence a better

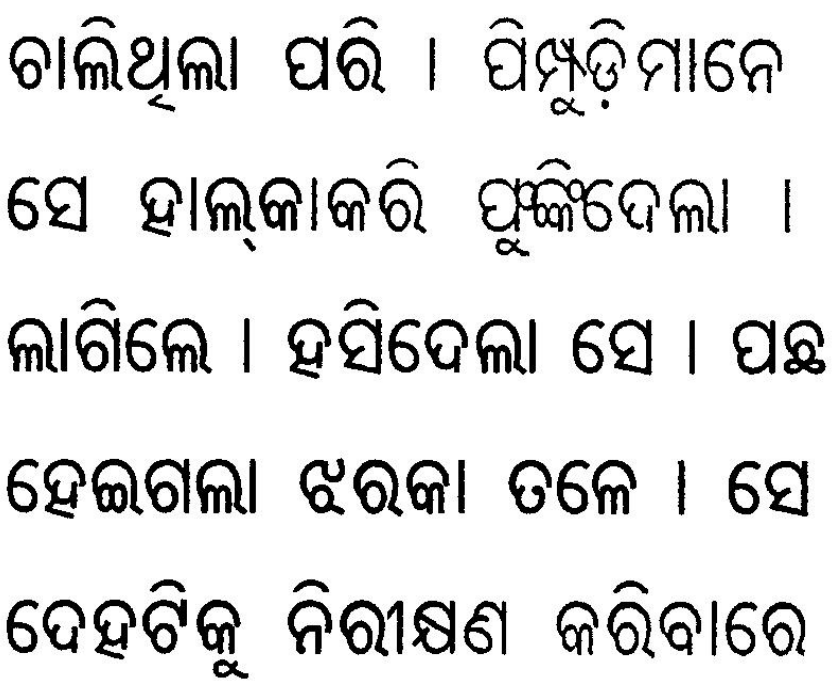


Figure 4.2: Binarization achieved by adaptive thresholding

binarization is achieved.

4.2.2 Skew Analysis

Kaushik *et al.* [25, 72] have proposed techniques to detect and correct the skew present in a given text document. The method (see Sec. 2.3.3) proposed by them is implemented in this work for detection of skew. However, their technique for skew correction is found to be unsatisfactory because it leads to a number of breaks in the characters. This may be observed from Fig. 4.4(a). This is because the rotation is performed on the binarized image.

In order to overcome this defect, rotation of the gray level image with bilinear interpolation is proposed and employed. The image, *after* such a (skew) correction mechanism, is subjected to binarization. The result of this processing (of the original skewed image in Fig. 4.3) is shown in Fig. 4.4(b). A comparison of Fig. 4.4(b) with Fig. 4.4(a) shows that skew correction before binarization is advantageous. It is observed that breaks in the characters in Fig. 4.4(b) are less compared to those in Fig. 4.4(a). Further, the character boundaries in the former are smoother.

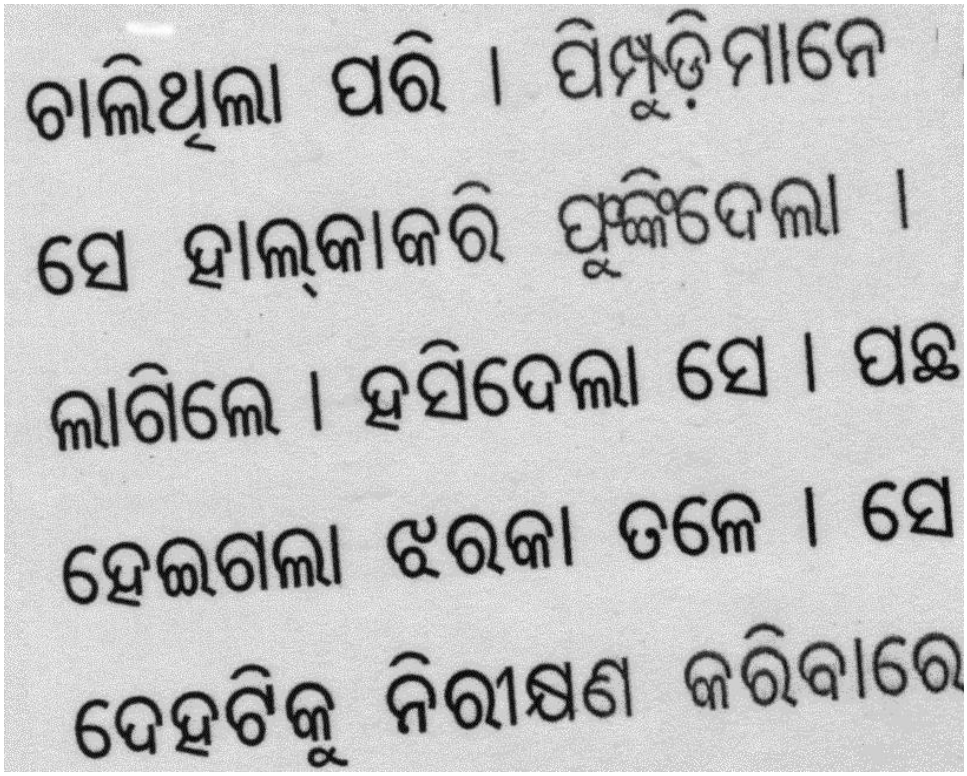
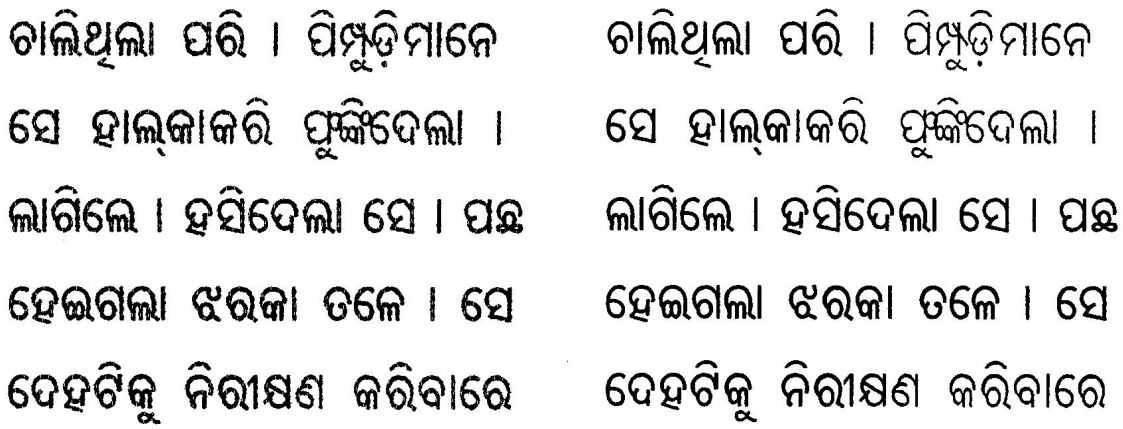


Figure 4.3: Original gray scale image with skew



(a) (b)

Figure 4.4: Skew correction achieved by: (a) rotating the binary image (b) rotating the gray image and then binarizing it.

4.3 Feature Extraction and Classification

After segmentation, the individual characters are available for recognition. The first level of analysis involves the grouping of the characters depending upon their aspect ratio. It is observed that with respect to the aspect ratio, Odiya characters generally fall into one of the following groups: (i) upto 0.2, (ii) 0.2 to 0.6, (iii) 0.6 to 1, (iv) 1 to 2, (v) 2 to 3.5, and (vi) above 3.5. The group into which the arriving character falls decides the size to which it is normalized. The group [0.6,1] encompasses all the basic Odiya characters, while the other groups are made up of matras, modifiers and compound characters.

A reference set of basic Odiya characters is created with 38 classes of frequently occurring characters, found by making a study of a number of documents collected from various sources. Each of the 38 classes is represented by a set of ten prototypes. The documents employed for arriving at the reference set belong to four different groups: (i) books published after 1995 with standardized script style, (ii) books published before the year 1990 earlier to the standardization of print-style (), (iii) manuscripts with computer generated fonts, and (iv) pages from *Sucharita*, a popular Odiya magazine. A set of more than 1600 test characters is formed with characters from all the above mentioned sources. It is ensured that the sizes and fonts of characters cover wide ranges.

In what follows, feature-specific results are presented. In other words, an analysis of the efficacy of individual features is presented with each of the classifiers discussed in Sec. 3.3. Before classification, the feature vectors are set to unit length with respect to all the features. The default parameters, unless explicitly mentioned, otherwise are as follows:

(a) For the k-NN and mk-NN classifiers, the value of k is set to 7.

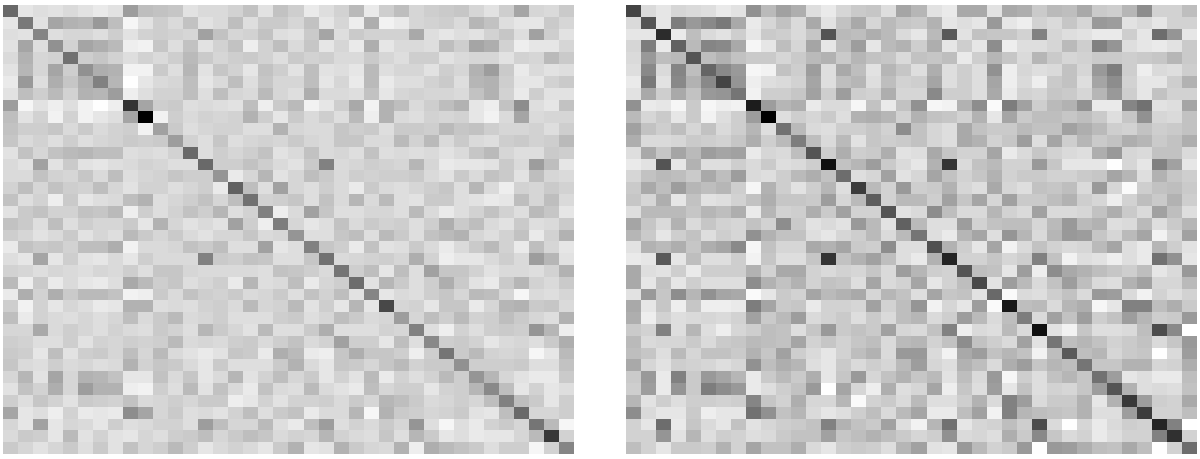
(b) The kernel type for SVM classifiers is Gaussian.

(c) The standard deviation of the Gaussian kernel in the SVM classifier ranges from 0.25 to 0.4. The actual value depends on the global standard deviation of all the components of the corresponding reference feature vectors.

4.3.1 Image Pixel Values

The characters employed for template matching have the following characteristics: (i) concatenated rows of the size-normalized character form the feature vector and (ii) no thinning operation is performed. The results corresponding to the various classifiers are shown in Tab. 4.1.

In order to identify the characters which fire false positives and false negatives, a class covariance matrix (CCM) is employed. The diagonal elements of a CCM give the intra-class variances of the employed features, whereas the off-diagonal elements give the inter-class variances of a given class with respect to others. In the inter-class case, the mean vectors represent the classes. It is observed that the diagonal elements of the CCM have dominant values implying that the correlation among vectors within



(a) (b)

Figure 4.5: CCM Image of the features (a) Image pixel values and (b) Projection Profile Vectors. The darkest point in the image corresponds to the highest value in the co-variance matrix.

a class is quite high. It also means that the inter-class correlation is relatively low. However, a closer look at the CCM image (in Fig. 4.5(a)) reveals that there do exist off-diagonal elements which have values close to their respective diagonal elements. Theoretically, the characters corresponding to these points are the ones which yield false positives and/or false negatives. This has also been empirically analyzed and the observations support this conclusion. Taking the class of characters corresponding to the ISCII value 209 as an example, the following are the observations:

- A total of 85 test characters are considered for recognition.
- 46 of the above characters are correctly classified *i.e.,* to class 209; 37 got mis-classified to class of 166.
- The values in the CCM shows that $\mathcal{C}_{209,209}$=0.0321 where as $\mathcal{C}_{166,209}$=0.2528. The other values corresponding to the same row or column in the CCM are very low in comparison to these values.
- Similar observations are made in other classes where there are more mis-classifications.

Table 4.1: Classification accuracies with image pixel values as features

Classifier	%Accuracy
NN	93.39
k-NN	84.36
mk-NN	91.69
SVM	90.43

Table 4.2: CLASSFICATION ACCURACIES WITH SAD AND MHUI AS FEATURES

Features	NN	k-NN	mk-NN
SAD	98.20	93.33	97.67
MHUI	91.17	86.27	90.53

Image pixel values, as features, yield good success rate because the amount of noise in printed character sets is relatively less. Further, the print quality and scanning resolution have been taken to be high, yielding relatively distortion-free characters. However, such an approach is not advisable in cases where characters of different fonts and orientations are to be recognized.

4.3.2 The SAD and the MHUI

The classification results with respect to the SAD and the MHUI (cf. Sec. 3.2.2) are presented in Tab. 4.2. A comparison of the SAD and the MHUI as features leads to the following observations:

(a) SAD forms a highly reliable and robust feature for Odiya characters.

Recognition accuracy drops to a large extent with MHUI.

(b) The computational requirements for SAD is enormous.

MHUI takes relatively lesser time for computation.

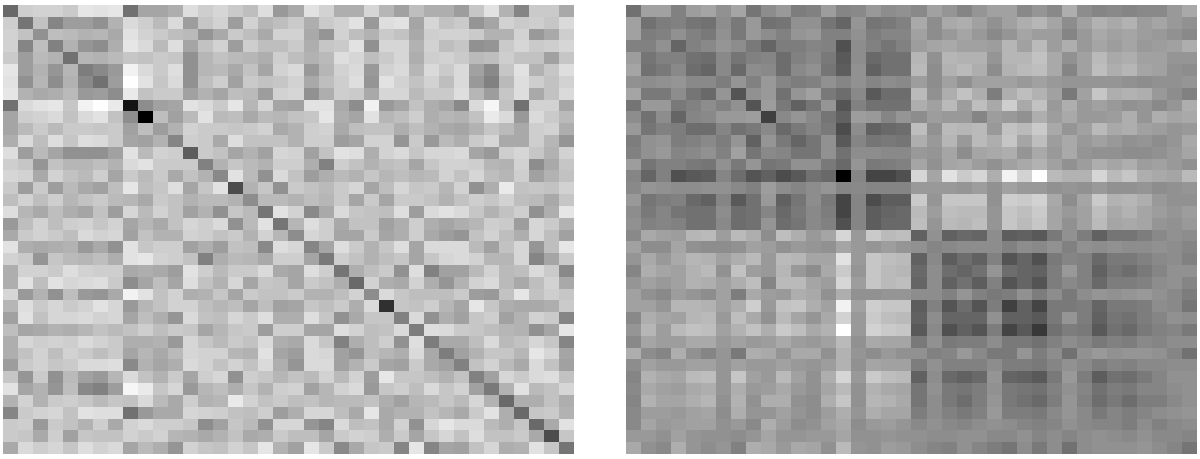
A quantitative analysis of the computational requirements of SAD and MHUI indicates the supremacy of the latter over the former with respect to the time consumed. The time taken to calculate the SAD is 58 minutes 10.99 sec on a test data set of 1619 characters with the reference set consisting of 380 characters. Whereas, the computation of MHUI consumed only 30 mins and 27.54 sec with the same data set.

4.3.3 Projection Profile

Results obtained with horizontal, vertical and two diagonal projection profiles of the 4 quadrants of the character are given in Tab. 4.3. Also presented are the results for thinned and normal characters. It may be observed that the accuracy of recognition reduces if the characters are thinned.

Table 4.3: CLASSIFICATION RESULTS WITH PROJECTION PROFILES

Type of Characters	NN	k-NN	mk-NN	SVM
Normal	95.43	84.81	89.73	96.29
Thinned	71.96	68.23	70.87	73.47



(a) (b)

Figure 4.6: CCM Image of the features (a) Second order Geometric Moments and (b) Legendre Moments. The darkest point in the image corresponds to the highest value in the co-variance matrix.

In spite of being a *simple* linear feature, the projection profiles are not suitable in many a cases because of the loss of localized spatial information, which, in turn, is the effect of the summing operation. In other words, it is not a **complete** feature (see Sec. 3.1). However, in this case, a high accuracy of recognition is observed. This may be attributed to the *noiseless* nature of the test and reference samples.

4.3.4 Geometric Moments

The normalized character is divided into 30 rectangular sectors, each of size 6×6. Second order geometric moments are extracted from these sectors and are concatenated to form a feature vector. Various classification techniques are applied to test their recognition accuracies. This technique is tried on both normal and thinned characters. Mahata *et al.* [25, 72] have reported the utility of thinning in the context of extracting such features (for Tamil text). However, in the case of Odiya characters, it is found that thinning has an adverse effect on the recognition accuracy. The results are tabulated in Tab. 4.4.

A rigorous study of geometric moments reveal the following with respect to their applicability for recognition of Odiya characters.

(i) With increase in the order of the moment, the recognition accuracy initially improves and then

Table 4.4: Classification Results with Geometric Moments

Type of Characters	NN	k-NN	mod-kNN	SVM
Normal	82.33	72.27	81.76	87.65
Thinned	70.79	64.03	69.84	71.23

drops. With further increase in the order, the accuracy again picks up and saturates. This is illustrated graphically in Fig. 4.7(a), where moments upto an order of 10 have been experimented with. It should be noted here that the moments of different orders are considered independently.

(ii) When **all** moments upto a given order are considered as features (that is, if the given order is 3, then moments corresponding to orders 1,2 and 3 are considered), it is observed that the phenomenon reported in the previous case repeats (See Fig. 4.5(c)). However, in this case, the drop in recognition accuracy falls drastically for order=6.

(iii) The CCM (Fig. 4.6) supports the results obtained with the recognizers. The dominance of the diagonal elements in the covariance matrix reveals that this feature is suitable.

Though the dimensionality of the feature vector increases with increase in the order, the drop in accuracy may be attributed to the extra attention paid to each and every pixel whereby undue importance is paid to the intra-class variation instead of considering intra-class similarity and inter-class variations.

Assuming that each of the pixels in a character pattern is independent of the other, for a pattern of size 36×30, there exists 1080 degrees of freedom. This means a vector of features, each independent of the other, can take a maximum dimension of 1080. From signal theory, it is known that the pixel values in the spatial domain are not uncorrelated. Hence the existence of such a high degree of freedom cannot be assumed. So the dependence of one component of the feature vector on another component is unavoidable. If the interdependence of the features contributes constructively, *i.e.,* more of the strong features are added in the vector, the accuracy of classification increases. However, if the interdependence is a destructive one, it results in a large number of weak components in the feature vector. This in turn leads to a high rate of misclassification of the test characters. This could explain the fall and then the rise in the accuracy curve with increase in the dimensionality of the feature vector. Since the number of degrees of freedom is unknown, it is always better to restrict the dimensionality to a small value. Accordingly, the first peak in the plot of accuracy vs. dimension should be chosen to be the optimum point whose index should indicate the optimum dimensionality of the feature vector.

In order to ensure better recognition accuracy, a statistical study of all the components of the selected feature vector should be carried out. To ensure better performance, the intraclass variance should be low and the interclass variance should be high. Feature vectors displaying (a) high similarity among intra-class members and (b) high variance with respect to members of other classes need to be chosen.

Increase in the order of the geometric moment leads to the following:

(a) Increase in the dimensionality of the input feature vector. This extends an invitation to the curse of dimensionality thereby increasing the complexity of certain classes of classifiers.

(b) Estimation error of the higher order moments increases, because the estimate is based upon a finite sample (reference) set.

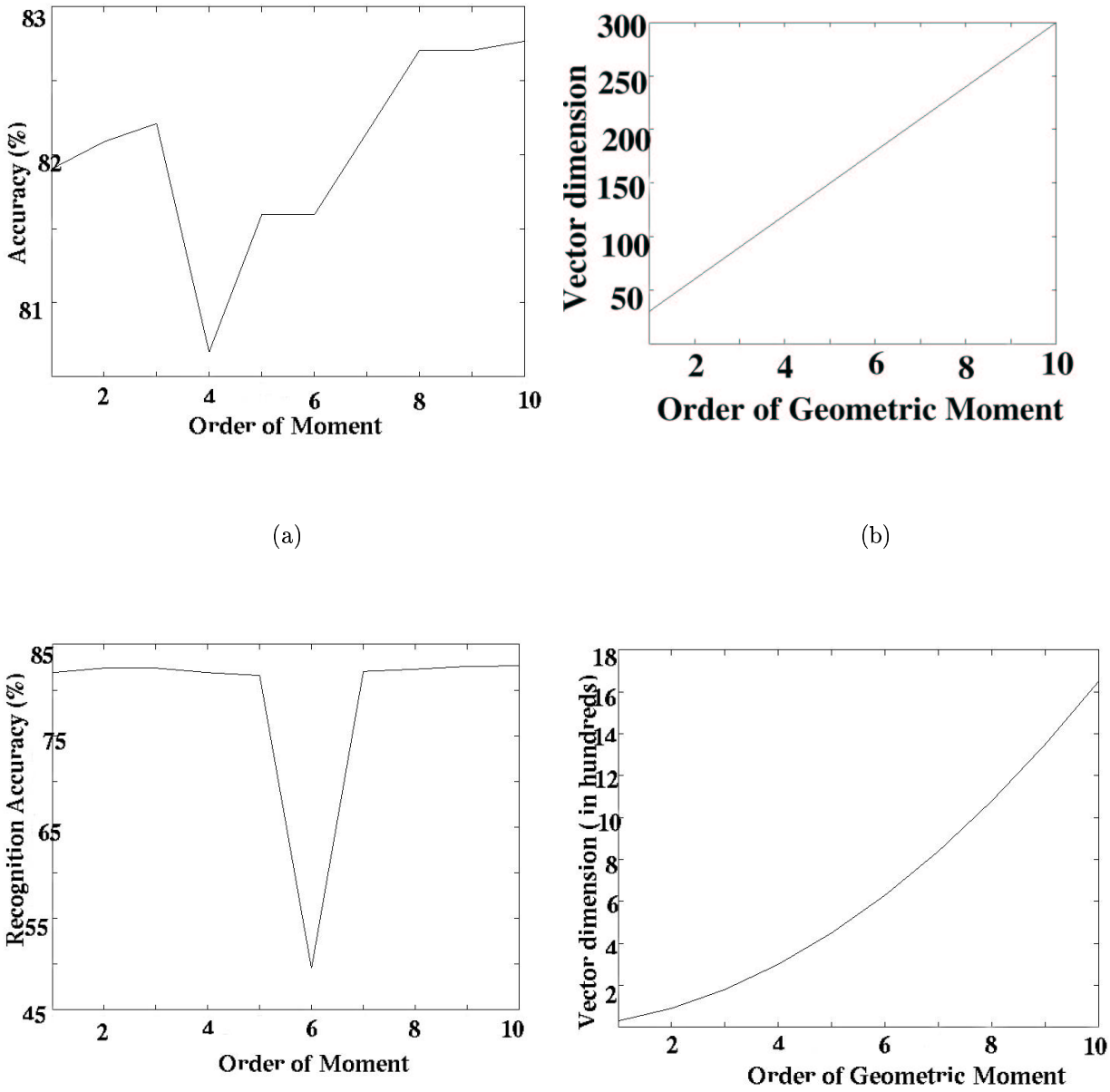


Figure 4.7: Classification accuracy as a function of the order of Geometric Moments: (a) percentage accuracy of recognition v/s order of moment, (b) vector dimension v/s order of moment, (c) percentage accuarcy of recognition v/s moments up to some order, and (d) vector dimension v/s moments up to some order.

Table 4.5: Classification Accuracies with Legendre Moments

Input Type	NN	k-NN	mod-kNN
Normal Chars	58.0	54.37	56.29
Sectors of Normal Chars	70.22	64.63	69.54
Thinned Chars	46.43	41.93	48.95

(c) The interdependence among the various features is unknown. This may lead to a positive or negative outcome. As the dimension increases, this becomes difficult to analyze.

(d) With so many input features, it may become difficult for the classifier (especially in a neural classifier) to generalize. This is because the intraclass variations get over-emphasized.

Hence, in order to avoid the above disadvantages, it is advisible to employ as small a number of moment coefficients, as possible.

4.3.5 Legendre Moments

Legendre moments are employed with the following inputs: (a) normal, non-thinned characters, (b) thinned characters and (c) individual sectors (see Sec. 3.2.6) of characters. The recognition accuracies with features corresponding to each of these are presented in Tab. 4.5 for various classifiers.

It is observed from Tab. 4.5 that features corresponding to Legendre moments *do not* form a satisfactory representation for Odiya characters. The CCM also is reflective of this: the number of off-diagonal elements having values closer to their diagonal counterparts is high. However, it is observed that a sector-based analysis yields a better performance than a global analysis.

4.3.6 Pixel Counts

The results with respect to ON-pixel counts in rectangular and sector blocks are shown in Tab. 4.6. Contrary to the claims made by Ganesh Murthy *et al.* [73] (for Roman characters), it may be observed here, that radial blocks do not offer any advantage as compared to rectangular blocks.

Table 4.6: Classification Accuracies with Pixel Count Features

Features	NN	k-NN	mod-kNN	SVM
Radial Blocks	75.79	66.15	76.84	78.13
Rectangular Blocks	81.9	76.13	82.07	82.23

4.3.7 Structural features and DCT coefficients

The existence of structural features, namely, vertical straight lines and/or loops in certain Odiya characters and their absence in others, is the motivation behind grouping the basic characters. This helps in analyzing the characters within a hierarchical paradigm. A grouping based on the presence of vertical line and loops is effectuated as described in Sec. 3.2.7.

After this primary grouping, DCT coefficients are extracted from the patterns and a comparison is done with prototypes of the same group. The procedure is followed for both thinned as well as non-thinned characters. The accuracy of recognition of the set of test characters is 93.02 % and 95.79 % in normal and thinned cases, respectively, using a nearest neighbour classifier.

4.3.8 Results of Redundancy Removal

A technique to remove the redundant points from characters has been proposed in Sec. 3.4. In order to analyze the efficacy of this process, the two-level recognizer of Sec. 4.3.7 is run on the data sets with and without decorrelation. It is observed that the intuitive advantage offered is confirmed by the results obtained. Employing a NN classifier, it is observed that the percentage accuracy goes up to 97.71 % in the case of decorrelated characters. It may be noted from the previous section that the percentage accuracy with normal characters is 93.02 %.

4.4 Conclusion

In this chapter, a comprehensive analysis of the results obtained as outcomes of the experiments carried out during the development of the OCR system has been presented. It has been shown that the proposed binarization and skew correction techniques perform more efficiently than those presented in the literature. The relevance of various structural and statistical features in the context of recognition of Odiya characters has been analyzed. A quantitative measure of this relevance is obtained by examining the class covariance matrix. Whereas the dominance of the diagonal elements is an indication of the validity of a feature, that of the off-diagonal elements indicates the relative unfitness for characterizing the characters. Exploiting the *superquadric nature* of Odiya characters, a context-dependent decorrelation method is proposed and found to be effective.

Chapter 5

Conclusions

"The true measure of the gift of ingenuity is not the ability to postulate complex solutions to seemingly complex problems, but rather to distill and design relatively simple, practical, and even elegant solutions to these problems. This should be the essence of all scientific research."
-Anonymous

In this thesis, an attempt to develop an integrated system for recognition of printed Odiya characters has been presented. In the course of the design process, methods which were found to be effective with Odiya characters have been employed as such. However, in cases where methods reported in the literature were found to be sub-optimal, appropriate modifications have been proposed and better efficiency has been achieved.

The major contributions of the thesis may be summarized as follows:

1. A ***maiden*** attempt to develop an integrated OCR system for printed Odiya characters has been made.

2. Selection and analysis (empirical and quantitative) of features which suit Odiya script have been presented.

3. A novel local-threshold scheme has been proposed and has successfully been implemented for binarization.

4. A modified approach to skew correction has been proposed and employed.

5. A new, computationally efficient feature, based on the union of patterns, has been formulated and employed.

6. A new method to decorrelate Odiya characters using superquadrics has been presented.

The results presented demonstrate the efficacy of the pre-processing techniques, the features and the classifiers used in this work. The proposed modifications in the binarization and the skew correction schemes have been shown to be superior to the existing methods. A comparative analysis of the various features and classifiers employed lead to the following observations:

(a) The highest recognition accuracy is obtained with the SAD-NN combination. However, the computation of SAD is time consuming.

(b) Striking a trade-off between computation time and recognition accuracy, it is seen that the hierarchical two-level scheme, employing a structural grouping followed by the DCT-NN pair, yields the same order of recognition accuracy (as in (a)) with far less computational cost.

(c) Removal of redundancy employing the *superquadric nature* of the Odiya ensemble enhances the recognition accuracy of the system.

5.1 Future Directions

The work presented here throws open a set of new avenues for research and/or development. While some of them point towards improvements with respect to the analysis of the language and the script, the others point towards modifications for computational efficacy. A few of them are enumerated below:

(1) The system presented here does not take into account patterns in Odiya script which rarely occur in common usage. However, a *complete and robust* system should include these as well.

(2) The performance of the system can be improved to a great extent if 'context' is brought into account. Explicitly, if a Odiya dictionary and Odiya grammar are incorporated into the system, so as to interpolate (and extrapolate) characters/words, then recognition accuracy can be improved by employing a spell-check mechanism and time taken for the task can be reduced by *guessing* characters/words.

(3) The idea of removing redundancy using an ensemble analysis has been effectively employed. If, instead of employing the entire set of characters to come out with the redundancy, the spatial correlation is analyzed among *similar* groups of characters, a better estimate of redundancy can be obtained. This, in turn, may lead to performance enhancement.

(4) A more systematic analysis of tree-structured classifier can be performed.

(5) Analysis of online and handwritten Odiya text can be done.

Bibliography

[1] B. B. Chaudhuri and U. Pal, "A complete Printed *bangla* OCR System," *Pattern Recognition*, vol. 31, no. 5, pp. 531–549, 1998.

[2] J. Mantas, "An overview of character recognition methodologies," *Pattern Recognition*, vol. 19, pp. 425–430, 1986.

[3] I. K. Sethi and B. Chatterjee, "Machine recognition of hand-printed Devanagari numerals," *Journal of the IETE(India)*, vol. 22, no. 8, pp. 532–535, 1976.

[4] I. K. Sethi, "Machine recognition of constrained hand–printed Devanagari," *Pattern Recognition*, vol. 9, pp. 69–75, 1977.

[5] R. M. K. Sinha and H. N. Mahabala, "Machine recognition of Devanagari script," *IEEE transaction on Systems, Man and Cybernetics*, vol. 9, no. 8, pp. 435 – 441, 1979.

[6] R. M. K. Sinha, "Role of context in Devanagari script recognition," *Journal of the IETE(India)*, vol. 33, pp. 86–91, 1987.

[7] R. M. K. Sinha, "Role of contexual postprocessing for Devanagari text recognition," *Pattern Recognition*, vol. 20, pp. 475 – 485, 1987.

[8] R. R. Karnik, "Identifying Devanagari characters," in *Proceedings of the International Conference on Document Analysis and Recognition*, pp. 669 – 672, 1999.

[9] V. Bansal and R. M. K. Sinha, "On how to describe shapes of Devanagari characters and use them for recogntion," in *Proceedings of the International Conference on Document Analysis and Recognition*, pp. 410 – 413, 1999.

[10] L. M. Pravin, "Recognition of documents printed in Devanagari," Master's thesis, Department of Electrical Engineering, Indian Institute of Science, Bangalore, 1999.

[11] S. H. Srinivasan and K. R. Ramakrishnan, "The independent components of characters are 'strokes'," in *Proceedings of the International Conference on Document Analysis and Recognition*, pp. 414–417, 1999.

[12] P. Comon, "Independent Component Analysis, A new Concept?," *Signal Processing*, vol. 36, pp. 287–314, 1994.

[13] A. Hyvarinen and E. Oja, "A fast fixed point algorithm for independent component analysis," *Neural Computation*, vol. 9, pp. 1483–1492, 1997.

[14] A. K. Dutta and S. Chaudhuri, "Bengali alpha-numeric character recognition using curvature features," *Pattern Recognition*, vol. 26, pp. 1757 – 1770, 1993.

[15] A. Bishnu and B. B. Chaudhuri, "Segmentation of hand written text into characters by recursive contour following," in *Proceedings of the International Conference on Document Analysis and Recognition*, pp. 402 – 405, 1999.

[16] A. Negi, B. Phanikumar, and B. K. Trinadh, "Offline printed Kannada script recognition system," in *International Workshop on Performance Evaluation Issues in Multi-lingual OCR*, Sept. 1999.

[17] T. V. Ashwin, "A font and size independent OCR for printed Kannada using SVM," Master's thesis, Department of Electrical Engineering Indian Institute of Science Bangalore, 2000.

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

[19] G. Siromoney, R. Chandrasekaran, and M. Chandrasekran, "Machine recognition of printed Tamil characters," *Pattern Recognition*, vol. 10, pp. 243–247, 1978.

[20] M. Chandrasekaran, R. Chandrasekran, and G. Siromoney, "Context dependent recognition of hand-printed Tamil characters," in *Proceedings International Conference on Systems, Man and Cybernatics (India)*, vol. 2, pp. 786–790, 1984.

[21] R. Chandrasekaran, M. Chandrasekran, and G. Siromoney, "Computer recognition of Tamil, Malayalam and Devanagari characters," *Journal of the Institute of Electronics and Teleccommunication Engg. (India)*, vol. 30, pp. 150–154, 1984.

[22] P. Chinnuswamy and S. G. Krishnamoorthy, "Recognition of hand–printed Tamil characters," *Pattern Recognition*, vol. 12, pp. 141–152, 1980.

[23] S. Sundaresan and S. S. Keerthi, "A study of representation for pen based handwriting recognition of Tamil characters," in *Proceedings of the International Conference on Document Analysis and Recognition*, pp. 422–425, 1999.

[24] I. Guyon, P. Albrecht, Y. L. Cun, J. Denker, and W. Hubbard, "Design of a neural network character recognizer for a touch terminal," *Pattern Recognition*, vol. 24, no. 2, pp. 105–119, 1991.

[25] K. Mahata, "Optical Character Recognition for printed Tamil script," Master's thesis, Department of Electrical Communication Engineering, Indian Institute of Science Bangalore, 2000.

[26] S. N. S. Rajasekaran and B. L. Deekshatulu, "Recogntion of printed Telugu characters," *Computer Graphics and Image Processing*, vol. 6, pp. 335–360, 1977.

[27] B. B. Chaudhuri, O. A. Kumar, and K. V. Ramana, "Automatic generation and recognition of Telugu script characters," *Journal of the Institute of Electronics and Telecommunication Engineers(India)*, vol. 37, pp. 499–511, 1991.

[28] M. B. Sukhaswami, P. Seetharamulu, and A. K. Pujari, "Recogntion of Telugu characters using neural networks," *International Journal of Neural Syatems*, vol. 6, pp. 317–357, 1995.

[29] J. J. Hopfield and D. W. Tank, "Computing with neural circuits: a model," *Science*, vol. 223, p. 625, 1986.

[30] R. Lippmann, "An introduction to computing with neural nets," *IEEE ASSP magazine*, vol. 4, 1987.

[31] G. Siromoney, R. Chandrasekaran, and M. Chandrasekran, "Machine recognition of Brahmi script," *IEEE transaction on Systems, Man and Cybernetics*, vol. 13, 1983.

[32] G. S. Lehal and R. Dhir, "A range free skew detection technique for digitized Gurumukhi script documents," in *Proceedings of the International Conference on Document Analysis and Recognition*, pp. 147–150, 1999.

[33] S. Antani and L. Agnihotri, "Gujarati character recognition," in *Proceedings of the International Conference on Document Analysis and Recognition*, pp. 418–421, 1999.

[34] P. B. Pati, A. G. Ramakrishnan, and U. K. A. Rao, "Machine Recognition of Printed Odiya Characters," in *Accepted for Proceedings of Int. Conf. on Inf. Tech.*, (Bhubaneswar, India), December 2000.

[35] U. Pal and B. B. Chaudhuri, "Script line separation from Indian muli-script document," *IEEE transaction on Pattern Analysis and Machine Intelligence*, vol. 19, 1997.

[36] B. B. Chaudhuri and D. D. Majumdar, *Two Tone Image Processing and Recognition.* New Delhi: Wiley Eastern Limited, 1993.

[37] T. Akijama and N. Hagita, "Automatic entry system for printed documents," *Pattern Recognition*, vol. 23, pp. 1141 – 1154, 1990.

[38] M. Chen and X. Ding, "A robust skew detection algorithm for gray scale document image," in *Proceedings of the International Conference on Document Analysis and Recognition*, pp. 617 – 620, 1999.

[39] L. O. Germon, "The document spectrum for page layout analysis," *IEEE transaction on Pattern Analysis and Machine Intelligence*, vol. 15, pp. 1162 – 1173, 1993.

[40] A. Haziume, P. S. Yeh, and A. Rosenfield, "A method for detecting the orientation of aligned components," *Pattern Recognition Letters*, vol. 4, pp. 125–132, 1986.

[41] H. S. Hou, *Digital Document Processing.* New York: John Wiley, 1983.

[42] X. Jiang, H. Bunke, and D. W. Kliajo, "Skew detection of document images by focused nearest neighbour clustering," in *Proceedings of the International Conference on Document Analysis and Recognition*, pp. 629 – 632, 1999.

[43] D. S. Le, G. R. Thoma, and H. Weschlar, "Automatic page orientation and skew detection of binary document image," *Pattern Recognition*, vol. 24, pp. 1325 – 1344, 1994.

[44] U. Pal and B. B. Chaudhuri, "An improved skew angle estimation," *Pattern Recognition Letters*, vol. 17, pp. 899 – 904, 1997.

[45] U. Pal and B. B. Chaudhuri, "Skew angle detection of digitized Indian script documents," *IEEE transaction on Pattern Analysis and Machine Intelligence*, vol. 19, 1997.

[46] T. Pavlidis and J. Zhau, "Page segmentation and classification," *Computer Vision, Graphics and Image Processing*, vol. 54, pp. 484 – 496, 1992.

[47] S. C. Hinds, J. L. Fisher, and D. P. Ameto, "A document skew detection method using runlength encoding and hough transform," in *Proceedings of the International Conference of Pattern Recogntion*, vol. 1, pp. 464–468, 1990.

[48] H. Yan, "Skew correction of document images using interline cross correlation," *Graphics Models and Image Processing*, vol. 55, pp. 538 – 543, 1983.

[49] K. Mahata and A. G. Ramakrishnan, "Precision skew detection through principal axis," in *International Conf. on Multimedia Processing and Systems*, (IIT Chennai), 2000.

[50] R. C. Gonzalez and R. E. Woods, *Digital Image Processing.* New York: Addison Wesley Publishing Co., 1993.

[51] A. K. Jain, *Fundametals of Digital Image Processing.* Prentice Hall of India Pvt. Ltd., 1995.

[52] A. K. Jain and Y. Zhong, "Page Layout Segmentation based on Texture Analysis," *Pattern Recognition*, vol. 29, pp. 743–770, may 1996.

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

[54] D. Wang and S. N. Srihari, "Classification of newspaper image blocks using texture analysis," *Computer Vision, Graphics and Image Processing*, vol. 47, pp. 327–352, 1989.

[55] E. E. Gose, J. W. Bacus, and L. Ackerman, "A comparision of some computer-measured and human-measured pattern recognition properties," *Journal of Cybernetics*, vol. 1, pp. 68–74, 1971.

[56] E. E. Gose, R. JohnsonBaugh, and S. Jost, *Pattern Recognition and Image Analysis.* New Delhi: Prentice Hall of India Pvt. Ltd., 1999.

[57] R. O. Duda and P. E. Hart, *Pattern classification and Scene analysis.* John Wiley and Sons, 1973.

[58] A. K. Jain, R. P. W. Duin, and J. Mao, "Statistical pattern recognition : A Review," *IEEE transaction on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 4–37, january 2000.

[59] P. A. Chou, "Optimal partitioning for classification and regression trees," *IEEE transaction on Pattern Analysis and Machine Intelligence*, vol. 13, no. 4, pp. 340–354, 1991.

[60] B. Vadim, K. Valentin, and H. Hermann, "A wave approach to pattern recognition (with application to optical character recogntion)," *International Journal of Bifurcation and Chaos*, vol. 4, no. 1, pp. 193–207, 1993.

[61] H. H. (ed.), *Neural and Synergitic Computers.* Berlin: Springer, 1988.

[62] L. Kuhnert, K. I. Agladze, and V. I. Krinsky, "Image processing using light sensitive chemical waves," *Nature*, vol. 337, pp. 244–247, 1989.

[63] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 955–974, 1998.

[64] V. Vapnik, *The Nature of Statistical Learning Theory.* New York: Springer, 1995.

[65] R. Collobert and S. Bengio, "On The convergence of SVMTorch, an Algorithm for Large Scale Regression Problems," tech. rep., Dalle Molle Institute for Perceptual Artificial Intelligence, Martigny, Switzerland, 2000.

[66] E. Bardinet, L. D. Cohen, and N. Ayache, "Tracking and motion analysis of the left ventricle with deformable superquadrics," *Medical Image Analysis Journal*, vol. 1, no. 2, pp. 129–149, 1996.

[67] A. Gupta and R. Bazcsy, "Surface and volumetric segmentation of range images using biquadrics and superquadrics," in *Proceedings of the 11th IAPR International Conference on Pattern Recognition*, pp. 158–162, 1992.

[68] D. Metaxas and D. Terzopoulos, "Constrained deformable superquadrics and nonrigid motion tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 337–343, 1991.

[69] D. Terzopoulos and D. Metaxas, "Dynamic 3d models with local and global deformations: Deformable superquadrics," *IEEE transaction on Pattern Analysis and Machine Intelligence*, vol. 13, no. 7, pp. 703–714, 1991.

[70] D. Marr, *Vision: A computational investigation into the human representation and processing of visual information.* Freeman and Co, 1982.

[71] A. Papoulis, *Probability, Random Variables and Stochastic Processes.* Singapore: McGraw-Hill Inc, 1991.

[72] A. G. Ramakrishnan and K. Mahata, "A complete OCR for Tamil printed text," in *Proceedings of Tamil Internet 2000*, (Singapore), 2000.

[73] C. N. S. G. Murthy and Y. V. Venkatesh, "Encoded pattern classification using constructive learning algorithms based on learning vector quantization," *Neural Networks*, vol. 11, no. 2, pp. 315–322, 1998.

Publications Related to the Thesis

1. Peeta Basa Pati and A.G. Ramakrishnan, "OCR in Indian Scripts: A Survey," Communicated to *Journal of IETE (India)*, July 2000.

2. Peeta Basa Pati, A.G. Ramakrishnan and U. K. Arvind Rao, "Machine Recognition of Printed Odiya Script," Accepted for presentation at International Conference on Information Technology – 2000, Bhubaneshwar, India, Dec. 2000.

3. Peeta Basa Pati and A.G. Ramakrishnan, "OCR for Printed Odiya Text," Communicated to *Sadhana (India)*, Oct. 2000.