OCR in Indian Scripts: A Survey

Peeta Basa Pati* and A.G. Ramakrishnan†, Fellow IETE
Department of Electrical Engineering
Indian Institute of Science
Bangalore 560 012, India

Abstract

India is a multi-lingual country. A significantly large number of scripts are used to represent these languages. A desire of vision researchers is to develop an integrated Optical Character Recognition (OCR) system which will be able to process all such scripts. Such a development, if objectified, will, not only enable faster flow of information across the country, but also have a profound impact on its scientific and economical development. Courageous endeavors have been successfully made towards the development of systems capable of recognizing machine-printed, or hand-written characters and/or numerals. However, most Indian scripts do not have an integrated OCR system. Further, the development of a unified system which is capable of processing all Indian scripts is still a dream. This article presents a survey of the current literature on the development of OCR's in Indian scripts. Reviewing the basics of and the motivation towards the development of OCR system, the article analyzes the various methodologies employed in general purpose pattern recognition systems. A critical analysis of the work towards OCR systems in Indian Languages, with pointers towards possible future work is also presented.

*e-mail: pati@ee.iisc.ernet.in
†Corresponding Author: e-mail: ramkiag@ee.iisc.ernet.in, Ph: +91 80 2293 2556

1 Introduction

India is a country which takes pride in 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, Governmental bodies of 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 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

considered to be 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. 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-today activities of life. These include:

1. A recognizer for analyzing the digits of the number plates of vehicles in motion; this would help traffic monitoring systems.
2. A reader to input text in electronic publishing.
3. Automated bill payment processing in customer-centric departments such as telephone and electricity.
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.

2 Methodologies in OCR

The first step towards automatic recognition of documents is the conversion of document manuscripts to digital images. The presence of pictures and/or graphics, in a typical document image, cannot be ruled out. Hence, the text areas of a given document have to be separated to enable subsequent processing. The text areas, thus extracted, are then processed to obtain lines of text first, words next, and finally individual characters. *Features* are extracted from these individual characters and are compared with those of a set of *reference* characters to decide the classes to which they belong. Thus, the steps involved in OCR can be formalized as follows(Fig. 1):

- **Layout Analysis:** This extracts the text-only portions of the document page from the rest of the contents of the page such as, graphics and pictures.
- **Preprocessing:** Preprocessing involves noise removal, skew detection and correction and 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.

Identifying and extracting the right features with minimal error is one of the most important tasks in automatic recognition of documents. The features that exhibit better

discriminating capabilities are chosen for use in the recognition phase. A broad taxonomy of features relevant to character recognition may be given as follows [1]:

- **Correlation-based features:**

  Correlation-based features involve distance measures which are computed with 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 a *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 reduce the dimensionality of the feature vector. Some of the transforms which have been 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 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.

The process of **Feature Selection**, which involves the choice of the right features for the given problem is done based on the structural and statistical properties of all the input patterns. Once the right features representative of the characters are extracted, they are fed to a classifier in order to identify the class to which the input character belongs. Some of the standard classification techniques employed for this purpose are [2]:

1. Neighborhood approach
2. Statistical approach
3. Neural network approach

The neighborhood approach for classification identifies the neighbors of the current feature point in the feature space. This is done by defining a distance measure. Some variants of this classifying technique are (i) the Nearest Neighbor(NN) classifier [3] (ii) k-NN classifier [4] (iii) condensed-NN classifier [5] (iv) reduced-NN classifier [6] (v) edited-NN classifier [7]. It may also be noted that classifiers with many other modified nearest neighbor rules have also been mentioned in the literature. Neighborhood based classifiers do not require any *a priori* knowledge of occurrence. They are simple to implement and involve less computation cost.

Statistical pattern classifiers rely on the statistics of the ensemble of features of the reference inputs. Typical statistical classifiers are the Bayesian classifier and the parzen window based classfier [2]. A disadvantage of such classifiers is the requirement that the probability of occurrence of a pattern should be known *a priori*. A relatively recent addition to the group of statistical classifiers is the Support Vector Machine (SVM) [8, 9]. SVM's are understood to be a set of optimal classifiers, where the decision boundary separates the individual class spaces.

Current research in pattern classification, however, focusses on neural network-based classifiers [10, 11]. This may be attributed to the benefits that such systems have to offer, like (i) trainability (ii) higher recognition speed (iii) better recognition accuracy (iv) no necessity to know the *exact* input-output relations (v) non-requirement of *a priori* knowledge of pattern distribution. Such systems are assumed to emulate the human brain in order to accomplish the task of pattern classification. They *learn* a distributed representation of the input-output relationship of the classifier from the *training* (reference) patterns. However, such systems are also not devoid of disadvantages. Some of the disadvantages of such a system are (i) large training periods (ii) difficulty in tuning the parameters to achieve optimal performance (iii) need to retrain the networks when new patterns are added to the library of reference.

3 OCR in Indian Scripts

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, in these scripts.

However, the availability of such commercial products for even printed text in Indian scripts is still a rarity. This manifests the necessity of research and subsequent development of systems capable of processing Indian scripts. Efforts towards developing such systems are being made around the country and elsewhere. Research work at institutions like Indian Statistical Institute, Calcutta and Centre for Development of Advanced Computing, Pune have resulted in the development of OCR systems for *Devanagari* and

Bangla scripts. Tamil Gnani, an OCR system for Tamil, is a product with very good performance (Test Report STQC-IT Delhi/OCR/002.[1]) This has come out of the research work being carried out at Indian Institute of Science, Bangalore.

An analysis of the languages used in India shows that most of the Indian scripts are almost phonetic in nature. All Indian scripts have the following characteristics:

1. They have a set of basic characters: the vowels and the consonants.
2. Indian scripts consist of composite characters, which are compositions of smaller units. A single composite character represents either a complete syllable, or coda of one syllable and the onset of another. A single character, known as an orthographic unit, consists of a single vowel, a set of consonants C1, C2, $\cdots$ followed by a vowel, V, or more rarely, a set of consonants followed by a *halant*.
3. A vowel-consonant combination is represented by the consonant associated with an extra entity called the *matra*. Each vowel has its own representative *matra* and hence a set of *matras* exist.
4. Most Indian scripts have their own representation for the numerals.

In some cases, the *matras* can be separated from the consonants and viewed as separate entities. However, there also exist cases where the matras combine with basic characters to form an entirely new symbol. Specific rules govern the modifications that the basic character undergoes when combined with the *matras*. The *matras* and the modified characters also have to be stored as reference patterns in order to have complete script recognition.

[1]STQC Directorate, Department of Information Technology, Ministry of Communications & Information Technology, Government of India.

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 Oriya scripts. Of these, usable systems for Devanagari, Bangla and Tamil are nearing maturity. Work is going on for development of bilingual OCR in Tamil. Efforts towards the development of OCR systems for other scripts are also on.

3.1 Bangla

Bangla is the official language of West Bengal state in India and of Bangladesh. It uses the Bangla Script. It is used by over 200 million people worldwide as their first language.

Dutta and Chaudhuri [12] 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, they have tried to expoit these from a thinned character and have used them for the purpose of recognition. 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 they extract 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. They also consider the number of strokes meeting at any junction and latter's normalized position in each of the strokes as components of the feature vector.

The classification is achieved in two steps. The strokes and the junctions which are

common over a set of characters in the alphbet 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.* [13] have contributed significantly to the development of OCR system in Bangla script. They have used the characteristic structural features of Bangla characters for efficient feature selection and extraction. In an effort to design a robust classifier, they make use of a tree-based decision sytem. They also implement a knowledge-based error-correction mechanism to improve the accuracy of the recognition system. They have used ISCII codes for representing their recognized output characters.

Chaudhuri *et al.* further suggest the exploitation of the *shiro-rekha* (*i.e.,* the horizontal straight line that acts as the head-line for the characters) for various preprocessing tasks. They have shown that it can be used for skew detection and segmentation of lines and words. They also identify it as a cue for distinguishing Bangla and Devanagari scripts from other scripts in a multi-script document.

In order to extract the relevant features of a character, they divide the character into three horizontal zones. This is based on the positions of the head and base-lines of the main character. The features they extract 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 OCR in Bangla has been attempted by Bishnu *et al.* [14].

3.2 Devanagari

Devanagari, the official script of India and Nepal, is the one used for representing Sanskrit, Nepali, Hindi, Marathi and many other languages used in the northern part of the country. It is approximately used by 700 million people around the world.

One of the earliest attempts towards the development of OCR in this script is by Sethi and Chatterjee, a group at IIT Kharagpur [15]. In this work, the authors have made an attempt to solve the problem of recognition of hand-printed Devanagari numerals. They extract four primitives (*i.e.,* the horizontal, vertical, left and right line segments) and the interconnections among these. Then they employ a decision tree-based classifier to recognize the characters. Later, using a similar technique, Sethi [16] has reported an attempt to recognize constrained hand-printed characters.

Sinha and Mahabala[17], another group at IIT Kanpur, tackle the problem through the syntactic pattern recognition approach. In this work, they quantify structural descriptions by the primitives and the interrelation between them, which are used later for recognition. Later, Sinha has analysed the role of contextual postprocessing in Devanagari documents [18, 19].

Chaudhuri *et al.* [13] 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. They effectively adapt the system developed for the Bangla characters (discussed in Sec. 3.2) for use with the Devanagari script. Such an attempt has been reported to yield high recognition accuracy. They have represented the recognized characters with ISCII code.

By effectively exploiting the intersections between the headlines on the top and the droppers from the headlines, Karnik [20] has identified individual characters. 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 quandrants and counting the number of ON-pixels in each quandrant. 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 case of compound characters, the same feature is employed with appropriate context-based changes. On the other hand, the representation scheme for the *matras* 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 [21] have successfully 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. They demonstrate the efficacy of a knowledge-based post-processing scheme by comparing recognition accuracies with and without it.

Lingayath [22] obtains the quadrants of the size-normalized characters. He, then, employs the horizontal and vertical widths of the four quadrants as features. Comparing the performance of various classifiers (see Sec. 2), he demonstrates the superiority of the Support Vector Machines (SVM's) over others. Further, he claims, with the selected set of features and the SVM classifiers, that the recognition accuracy decreases if the characters are thinned before the features are extracted. Apart from this, he also analyzes the results

with a Minimum Distance Classifier(MDC) to conclude that a high recognition rate may be achieved in this case, if features are extracted after thinning. Lingayath used ISCII codes for representing the recognized characters.

Srinivasan and Ramakrishnan [23], analyzing the applicability of Independent Component Analysis (ICA) [24, 25] 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.

3.3 Kannada

Kannada is the state language of the southern state of Karnataka. It's spoken by over 50 million people worldwide. It is used in the states of Karnataka, Andhra Pradesh, Tamil Nadu and Maharashtra. It uses the Kannada script.

Atul *et al.* [26] extract geometric moments such as zernike and pseudo-zernike moments ([27] is a good reference on moments) and have used a probablistic neural network classifier for recognition of printed Kannada characters. They report that the pseudo-zernike moments give better recognition accuracy with the same classifier than the zernike moments. They also have considered the histogram of the contour directions and the projection profile as features. Classifying the features with the quadratic discriminant classifier, they report the recognition accuracy obtained to be disappointingly low.

Ashwin [28] 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 Zernike moments and some structural features on the recognition accuracy of the system.

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 classfiers 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.

Vijay [29, 30] has evaluated various kinds of structural and transform based features with neural network based as well as distance based classifiers. He finally has reported that features extracted employing Haar wavelet transform with RBF classifier give the highest recognition accuracy (about 99%) for base Kannada characters. He has employed ISCII codes for representation of the output.

3.4 Tamil

Tamil is the official language of the southern state of Tamil Nadu and also of Singapore, Sri Lanka and Mauritius. It is spoken by 65 million people worldwide. It is spoken in Tamil Nadu and its neighboring states. Worldwide, it is also spoken in Bahrain, Malaysia,

Qatar, Reunion, Thailand, United Arab Emirates and United Kingdom. It uses Tamil script for its representation.

Shiromoney *et al.* [1, 31] have proposed an encoded character string dictionary for recogniton of Tamil characters. This principle of theirs is taken by Chandrasekaran *et al.* [32, 33] for recognition of constrained handprinted Tamil charcters and later for multifont Tamil character recognition.

Chinnuswamy and Krishnamoorthy [34] tried to recognize the hand printed Tamil characters by employing labelled graphs. These graphs describe the structural composition of the characters in terms of line like primitives. They use a correlation-based, template matching technique, where the character under consideration is matched with the prototypes.

Mahata [35, 36] considers a three–level hierarchical feature extraction and classification system. Adapting the strategy of Chaudhuri *et al.* [13], he divides words into three horizontal segments (cf. Sec. 3.2). The first of the three levels assigns the character to one of the four possible groups. This is accomplished by analyzing the spatial spread of the character in the three segmental areas. 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. The second level of feature extraction involves the extraction of second order geometric moments from each of these sectors. Classification at this stage is realized with the help of a neural network. If this results in ambiguities, then, at the third level, DCT-based features and a neural network classifier execute the final decision-making process.

Extending the work of Mahata [35, 36], Dhanya has tried to address the issue of recognizing the characters present in a bilingual Tamil document [37]. She has taken an

approach where she initially tries to identify the script of a word using directional Gabor filters [38, 39]. Later she has explored various features to show that a hierarchical scheme with grouping of classes based on the spatial spread of the characters at the primary level with DCT as a feature at the secondary level give the best possible performance [40]. Using the results obtained by Dhanya, Aparna [41] has reported the development of an OCR system for Tamil script. She has used both ISCII code and Unicode for representing the output of the recognition engine.

3.5 Telugu

Telugu is the official language of the south-central state of Andhra Pradesh. It is spoken by over 75 million people worldwide. It is spoken in Andhra Pradesh and neighboring states. Worldwide, it is spoken in Malaysia, Fiji, Singapore, Bahrain and United Arab Emirates. It is represented by the Telugu script.

Rajasekaran and Deekshatulu [42] are the pioneers in the recognition work of Telugu characters. They have identified around 2000 characters in Telugu script. After a careful analysis of these characters, they found that all these characters have been developed from 25 basic characters with the help of some builders and primitives. They took the initiative to break the characters into their primitives and basic characters and recognize them separately. In their approach, the primitives are separated from the basic characters using a syntax aided recognition scheme. They have tried to implement an on-the curve method to describe the basic characters. Then the 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.

Chaudhuri *et al.* [43] have tried to recognize machine generated Telugu charatcters.

They have introduced the concept of T-tuples for the tracing of the skeletonized characters. The primitives, at first, are recognized by this curve tracing method followed by the recogntion of the basic character using the same algorithm.

Sukhaswami *et al.* [44] have taken a neural network approach to the recognition of both printed and hand–written Telugu characters. In order to achieve robustness to noise, they have exploited the associative memory property of Hopfield networks [45, 46]. In an effort to overcome the huge memory requirement of such networks, they propose the Multiple Neural Network Associative Memory (MNNAM) as an alternative. 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 of 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.

3.6 Others

There are many Indian scripts on which work has started in the recent past. Pati [47, 48] has experimented with various kinds of features (both statistical and transform based) with distance based classifiers. He has also employed SVM as a classifier and compared its performance with distance based classifiers. Later this work has been extended for the development of an OCR system for mono-lingual Odiya script [49]. For extending this OCR engine for making it compatible to bilingual documents, Pati *et al.* [50] have employed directional Gabor filters for script identification at the word level. This scheme

has been shown to work successfully for combinations of Roman with Tamil and Devanagari as well. Mohanti [51] has proposed a recognition scheme of Odiya alphabets based on Kohonen's neural network. Here, the pixel values of the test character are fed directly to the network and classified based on the network output. Chaudhuri *et. al.* [52] have presented a two level recognition scheme for printed Odiya characters.

Shiromoney *et al.* [53] have proposed a scheme for recognition of Brahmi script, an ancient Indian script. They accomplished this by adapting their own scheme for Tamil script (Sec. 3.4). Chandrasekaran *et al.* [33] have tried their method (Sec. 3.4) on Malayalam script. The work of Lehal and Dhir [54, 55, 56] towards the development of an OCR system for the Gurumukhi script is significant. Work on a Gujarati script analyzer is being carried out by Antani and Agnihotri [57].

Chaudhuri *et al.* [58] have proposed an algorithm for the identification of scripts in a multi-script document. Here, they have made a restrictive assumption that each text line consists of only a single script. They have extracted the nature of the projection profile of the text line and based on its characteristics, have been able to separate the texts of different scripts in such a document.

4 Discussion

In this section, an overall analysis of the methodologies employed by OCR researchers of Indian scripts is presented. It also poses certain questions to current researchers and points towards possible directions for future research.

A general overview on the methods employed to analyze compound characters indicates the existence of two orthogonal schools of thought. One school supports the view of

analyzing the *matras* or modifiers after separating them from the base character. The other, however, prefers to analyze the compound characters as single entities. In view of the fact that the *matras* or modifiers may also be touching the base characters, a decision on the choice among the two schools has to be context-dependent. Explicitly, while *matras* which can be separated from the base character should be analyzed separately, those which *change* the shape of the base character should be analyzed as single entities (*i.e.,* along with the base characters).

It has long been known that the primate visual system learns patterns by analyzing their structural features ((*e.g.* strokes, loops etc.) and their inter-relations [59, 60]. This may be exploited in the design of a machine recognition system by choosing and extracting appropriate structural features. To this end, the shapes of the curvatures, junction points, strokes and the inter-relations between them have been utilized as features. Moreover, such features are more relevant to Indian scripts because: (a) the curvilinear structure of most Indian script characters and (b) position of modifiers or *matras* decide the exact nature of characters. Most of the work carried out in Indian scripts employ such structural features.

Apart from the use of such structural features for recognition, researchers have also employed them effectively for pre-processing purposes. For instance, the use of the top-line has been used to simplify both the pre-processing and recognition of Devanagari and Bangla characters (cf. [13, 22]).

Future researchers should also note that statistical features also play important roles in describing patterns [28, 61]. In the schemes analyzed, statistical features, if they are used, have been used as ambiguity resolvers and not as primary features. Though explorations towards such features have remained dormant, it is believed that they are of appreciable

significance towards the analysis of Indian scripts [1, 62].

Features which analyze the back-ground geometry of characters by fitting appropriate curves to it (like circles, ellipses, etc.) and moment-based features have been utilized by researchers dealing with Roman and oriental scripts. OCR researchers of Indian scripts have also used such features [26, 35].

Hidden Markov Models (HMM) [63] are shown to be of intense utility for recognition of both on- and off-line hand-written as well as machine-printed character recognition [64, 65, 66, 67, 68]. The gray valued templates are also used for segmentation and recognition of characters by Lee *et al.* [69]. These provide an advantage over the binary templates in which some information has already been lost during the process of binarization. This may also be tried by future Indian-script OCR researchers.

Almost all possible types of classifiers have been successfully utilized in Indian context. It has been demonstrated by many that neural network-based classifiers are superior over their counterparts. Nevertheless, classical statistical approaches to classification [2] remain largely unexploited. As these classification techniques require *a priori* information on the occurrence of the individual classes, it helps in embedding the *context* or *nature* of the given language into the classifier. However, this advantage also has turned out to be the stumbling block towards the realization of such a classifier owing to the nonavailability of such information. Of late, rigorous analysis carried out by the Central Institute for Indian Languages and the Indian Statistical Institute has opened up ways to explore such classifiers.

Knowledge-based error-correction approach is shown to provide better recognition accuracy [21, 13]. This is a potential area for investigation, which will enable development of self-contained OCR systems. Script identification in multi–lingual documents is still

in its infancy. It is the first and, yet, one of the most important subtasks in developing multi-script OCR systems. The goal of future work has to be focussed towards the design of a unified multi-script automatic document analysis system.

One of the important system related issues to be addressed after the recognition of characters is the issue of coding the output. In the Indian scenario, there are a lot of font based coding schemes used by various manufacturers of software that facilitate word processing or web page creation. However, it is extremely difficult to go from one font based coding to another and this makes the interoperability of the outputs a very difficult problem to handle. Ideally, one must use a character based coding, and ISCII is an ideal choice for that. The issues of character coding must be kept distinct from the issues of font coding, and there must be a legislation that, irrespective of the font coding used by any commercial software, there must be a provision to export the output in a standard character based coding. Unfortunately, even the Unicode for some of the Indian languages, is partly font based, and hence is not a convenient coding system, when it comes to natural language processing, such as, searching, sorting, etc. from the text database.

It is crucial to standardize the sort order for every Indian language, including in the character set all the additional symbols already used to represent borrowed words from English and other Indian languages. Then the coding used to store the output information from an OCR or a handwriting recognition system or a word processing system, must be strictly character based, and should follow the standard sort order. This will be a major step in facilitating our languages to be used increasingly on computers and simplifying inter-operability of different softwares and language technology tools. NLP experts, computer scientists and computational linguists must be consulted, and a committee consisting of specialists in the above areas, must make a recommendation to the

concerned Governmental bodies and the Unicode consortium. In the meanwhile, it may be meaningful to use ISCII and/or Unicode to store the recognized output of character recognition systems.

5 Conclusion

In this article, an attempt has been made to describe the present status of OCR work in Indian scripts. The motivation behind such a work and the methodologies used are described in detail. Most of the research work already reported are explained in brief. Some areas which are either under-explored or left untouched are brought to the limelight. Some future directions of research in this area are also proposed.

Though many researchers have suggested many different methods of feature extraction and classification, none have been found to be self–contained. So some of the researchers have stressed on the need for different levels of feature extraction and classification. Many are of the opinion that knowledge should be used to take rectifiable steps to increase the accuracy of recognition. Some have taken these suggestions and have developed working OCR models.

From the above study, it can be said that the machine recognition of Indian script documents requires a lot more work and a multi-lingual system to take care of most or all of these scripts still remains to be a realizable dream.

References

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

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

[3] T. Cover and P. Hart, "Nearest neighbor pattern classification," IEEE Transactions on Information Theory, vol. 13, pp. 21–27, January 1967.

[4] J. Goin, "Classification bias of the k-nearest neighbor algorithm," IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 6, pp. 379–381, May 1984.

[5] P. Hart, "The condensed nearest neighbor rule," IEEE Transactions on Information Theory, vol. 14, pp. 515–516, May 1968.

[6] G. Gates, "The reduced nearest neighbor rule," IEEE Transactions on Information Theory, vol. 18, pp. 431–433, May 1972.

[7] I. Tomek, "An experiment with the edited nearest-neighbor rule," IEEE Transactions on Systems, Man, an Cybernetics – 6, vol. 6, 1976.

[8] 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.

[9] N. Cristianini and J. Shawe-Taylor, AN INTRODUCTION TO SUPPORT VECTOR MACHINES (and other kernel-based learning methods). cambridge University Press, 2000.

[10] C. M. Bishop, Neural Networks for Pattern Recognition. Oxford Univ Press, 1995.

[11] S. Haykin, Neural Networks : A Comprehensive Foundation. Prentice Hall, 1999.

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

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

[14] A. Bishnu and B. B. Chaudhuri, "Segmentation of hand written text into characters by recursive contour following," in Proceedings of the 5^{th} International Conf. on Document Analysis and Recognition, pp. 402 – 405, 1999.

[15] 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.

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

[17] 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.

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

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

[20] R. R. Karnik, "Identifying Devanagari characters," in Proc. of the 5^{th} International Conf. on Document Analysis and Recognition, pp. 669 – 672, 1999.

[21] V. Bansal and R. M. K. Sinha, "On how to describe shapes of Devanagari characters and use them for recogntion," in Proc. of the 5^{th} International Conf. on Document Analysis and Recognition, pp. 410 – 413, 1999.

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

[23] S. H. Srinivasan and K. R. Ramakrishnan, "The independent components of characters are 'strokes'," in Proc. of the 5^{th} International Conf. on Document Analysis and Recognition, pp. 414–417, 1999.

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

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

[26] 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.

[27] R. Mukundan and K. R. Ramakrishnan, "Fast computation of legendre and zernike moments," Pattern Recognition, vol. 28, pp. 1433–1442, 1995.

[28] 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.

[29] B. V. Kumar, "Machine recognition of printed kannada text," Master's thesis, Indian Institute of Sciences, Bangalore, INDIA – 560 012, 2002.

[30] B. V. Kumar and A. G. Ramakrishnan, "Machine recognition of printed kannada text," in Int. Workshop DAS 2002, Princeton, NJ, USA, August 2002 (D. Lopresti, J. Hu, and R. Kashi, eds.), vol. 2423 of Lecture Notes in Computer Science, pp. 37–48, Springer-Verlag, 2002.

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

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

[33] R. Chandrasekaran, M. Chandrasekran, and G. Siromoney, "Computer recognition of Tamil, Malayalam and Devanagari characters," Journal of the IETE (India), vol. 30, pp. 150–154, 1984.

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

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

[36] A. G. Ramakrishnan and K. Mahata, "A complete ocr for printed tamil text," in Proceedings of Tamil Internet, pp. 165–170.

[37] D. Dhanya, "Bilingual ocr for tamil and roman scripts," Master's thesis, Indian Institute of Sciences, Bangalore, INDIA – 560 012, 2001.

[38] D. Dhanya and A. G. Ramakrishnan, "Script identification in printed bilingual documents," in Int. Workshop DAS 2002, Princeton, NJ, USA, August 2002 (D. Lopresti, J. Hu, and R. Kashi, eds.), vol. 2423 of Lecture Notes in Computer Science, pp. 13–24, Springer-Verlag, 2002.

[39] D. Dhanya, A. G. Ramakrishnan, and P. B. Pati, "Script identification in printed bilingual docuements," Sadhana, vol. 27, pp. 73–82, 2002.

[40] D. Dhanya and A. G. Ramakrishnan, "Optimal feature extraction for bilingual ocr," in Int. Workshop DAS 2002, Princeton, NJ, USA, August 2002 (D. Lopresti, J. Hu, and R. Kashi, eds.), vol. 2423 of Lecture Notes in Computer Science, pp. 25–36, Springer-Verlag, 2002.

[41] K. G. Aparna and A. G. Ramakrishnan, "A complete tamil optical character recognition system," in Int. Workshop DAS 2002, Princeton, NJ, USA, August 2002 (D. Lopresti, J. Hu, and R. Kashi, eds.), vol. 2423 of Lecture Notes in Computer Science, pp. 53–57, Springer-Verlag, 2002.

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

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

[44] 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.

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

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

[47] P. B. Pati, A. G. Ramakrishnan, and U. K. A. Rao, "Machine Recognition of Printed Odiya Characters," in Proc. of Int. Conf. on Inf. Tech., pp. 227–232, 2000.

[48] P. B. Pati, "Machine recognition of printed odiya text," Master's thesis, Indian Institute of Sciences, Bangalore, INDIA – 560 012, 2000.

[49] N. K. Pati, P. B. Pati, and A. G. Ramakrishnan, "An Optical Character Recognizer in Odiya," in Proc. of Int. Conf. on Inf. Tech., pp. 590–591, 2003.

[50] P. B. Pati, S. S. Raju, N. K. Pati, and A. G. Ramakrishnan, "Gabor Filters for document analysis in Indian bilingual documents," in Proc. of Int. Conf. on Intelligent Sensing and Information Processing, pp. 123–126.

[51] S. Mohanti, "Pattern recognition in alphabets of oriya language using kohonen neural network," Int. J. Pattern Recogn. Artif. Intell., vol. 12, pp. 1007–1015, 1998.

[52] B. B. Chaudhuri, U. Pal, and M. Mitra, "Automatic recognition of printed oriya script," in Proc. of the 6^{th} Int. Conf. on Document Analysis and Recognition, pp. 406–409, 2001.

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

[54] G. S. Lehal, "Feature extraction and classification for ocr of gurumukhi script," Vivek, vol. 12, pp. 2–12, 1999.

[55] G. S. Lehal and R. Dhir, "A range free skew detection technique for digitized Gurumukhi script documents," in Proc. of the 5^{th} Int. Conf. on Document Analysis and Recognition, pp. 147–150, 1999.

[56] G. S. Lehal and C. Singh, "A gurumukhi script recognition system," in Proc. of the 15^{th} Int. Conf. on Pattern Recognition, pp. 557–560, 2000.

[57] S. Antani and L. Agnihotri, "Gujarati character recognition," in Proc. of the 5^{th} Int. Conf. on Document Analysis and Recognition, pp. 418–421, 1999.

[58] U. Pal and B. B. Chaudhuri, "Script line separation from Indian muli-script document," in Proc. of the 5^{th} Int. Conf. on Document Analysis and Recognition, pp. 406–409, 1999.

[59] S. Kahan, T. Pavlidis, and H. S. Baird, "On the recognition of printed characters of any font and size," IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 9, no. 2, pp. 274–288, 1987.

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

[61] 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.

[62] S. Mori, C. Y. Suen, and K. Yamamoto, "Historical review of OCR research and development," Proceedings of IEEE, vol. 80, no. 7, pp. 1029–1058, 1992.

[63] L. Rabiner, "A tutorial on Hidden Markov Models and selected applications in speech recognition," Proceedings of IEEE, vol. 77, no. 2, pp. 257–286, 1989.

[64] B. Al-Badr and S. Mahmoud, "Survey and bibliography of Arabic optical text recognition," Signal Processing, vol. 41, no. 1, pp. 49–77, 1995.

[65] I. Bazzi, R. Schwartz, and J. Makhoul, "An omni-font open vocabulary OCR system for english and arabic," IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 21, no. 6, pp. 495–504, 1999.

[66] W. Cho, S.-W. Lee, and J. H. Kim, "Mdodelling and recognition of cursive words with Hidden Markov Models," Pattern Recognition, vol. 28, pp. 1941–1953, 1995.

[67] J. Hu, M. K. Brown, and M. Turin, "Hmm based online handwriting recognition," IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 18, no. 10, pp. 1039–1045, 1996.

[68] M. Mohamed and P. Gader, "Hand-written word recognition using segmentation-free Hidden Markov Modeling and segmentation based dynamic programming techniques," IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 18, no. 5, pp. 548–554, 1996.

[69] L. Seong-Whan, L. Dong-June, and P. Hee-Seon, "A new methodology for gray-scale character segmentation and recognition," IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 18, no. 10, pp. 1045–1050, 1987.