A New Approach for Hindi Optical Character Recognition Based On Neural Networks

[1]Ajay Goel, [2]O.P.Sahu, [3]Sheifali Gupta, [4]Rupesh Gupta

***Abstract* —OCR is the acronym for Optical Character Recognition. This technology allows a machine to automatically recognize characters through an optical mechanism. Human beings recognize many objects in this manner our eyes are the "optical mechanism.**

Development of OCRs for Indian script is an active area of activity today. Optical character recognition (OCR) is the mechanical or electronic translation of images of handwritten, typewritten or printed text (usually captured by a scanner) into machine-editable text. In simple words OCR is a visual recognition process that turns printed or written text into an electronic character based file. OCR is a field of research in pattern recognition, artificial intelligence and machine vision. Though academic research in the field continues, the focus on OCR has shifted to implementation of proven techniques. A lot of work had been carried out for OCR at international scenario but in Indian context a concrete approach for character recognition is still required as scripts of Indian languages are from the group of most complex scripts and it is very hard to recognize them. Indian scripts present great challenges to an OCR designer due to the large number of letters in the alphabet, the sophisticated ways in which they combine, and the complicated graphemes they result in. The problem is compounded by the unstructured manner in which popular fonts are designed. There is a lot of common structure in the different Indian scripts. All existing OCR systems developed for various Indian scripts do not provide sufficient efficiency due to various factors. The objective of this paper is to discuss a more efficient character recognition technique. This paper introduces a new technical approach to recognize Indian script characters which are unpredictable due to different problems in other OCR's.

Index Terms—**OCR, Character, Skewed Character, Character Recognition, Artificial Neural Network**.

I. INTRODUCTION

The OCR is such an advanced technology, using which our day to day cumbersome job of manually creating soft-copies is resolved with a blink of eye. We are looking forward to get benefited using this technology for our own scripts, i.e. basically Indian scripts. But this is a great challenge for us to make this dream come true since Indian language's scripts are from the most complex scripts group and it is very hard to recognize them. So this paper is an attempt to resolve most of such complex problems regarding character recognition for Indian scripts and to come out with a scope of having a perfect or at least a near to perfect approach.

A. CURRENT STATE OF OCR TECHNOLOGY

The accurate recognition of Latin-script, typewritten text is now considered largely a solved problem. Typical accuracy rates exceed 99%, although certain applications demanding even higher accuracy require human review for errors. Other areas--including recognition of hand printing, cursive handwriting, and printed text in other scripts (especially those with a very large number of characters)--are still the subject of active research. Systems for recognizing hand-printed text on the fly have enjoyed commercial success in recent years. Among these is the input device for personal digital assistants such as those running Palm OS. The Apple Newton pioneered this technology. The algorithms used in these devices take advantage of the fact that the order, speed, and direction of individual lines segments at input are known. Also, the user can be retrained to use only specific letter shapes. These methods cannot be used in software that scans paper documents, so accurate recognition of hand-printed documents is still largely an open problem. Accuracy rates of 80% to 90% on neat, clean hand-printed characters can be achieved, but that accuracy rate still translates to dozens of errors per page, making the technology useful only in very limited applications. This variety of OCR is now commonly known in the industry as ICR, or Intelligent Character Recognition. Recognition of cursive text is an active area of research, with recognition rates even lower than that of hand-printed text[4][5]. Higher rates of recognition of general cursive script will likely not be possible without the use of contextual or grammatical information. For example, recognizing entire words from a dictionary is easier than trying to parse individual characters from script. Reading the Amount line of a cheque (which is always a written-out number) is an example where using a smaller dictionary can increase recognition rates greatly. Knowledge of the grammar of the

[1]Ajay Goel, [2]O.P.Sahu, [3]Sheifali Gupta , [4]Rupesh Gupta
[1]Assistant Professor, HIET, Kaithal (Haryana) E-mail: goelajay1@gmail.com
[2] Assistant Professor, NIT, Kurukshetra (Haryana) E-mail: ops_nitk@yahoo.co.in,
[3]Assistant Professor, HIET, Kaithal (Haryana) E-mail: sheifali@yahoo.com
[4]Assistant Professor, HCTM, Kaithal (Haryana) E-mail: rup_esh100@yahoo.co.in

language being scanned can also help determine if a word is likely to be a verb or a noun, for example, allowing greater accuracy. The shapes of individual cursive characters themselves simply do not contain enough information too accurately (greater than 98%) recognize all handwritten cursive script. It is necessary to understand that OCR technology has to be understood as a basic technology also used in advanced scanning applications. Due to this fact the reader should understand that an advanced scanning solution can be unique and patented and not easily copied despite being based on this basic OCR technology. For more complex recognition problems, intelligent character recognition systems are generally used, as artificial neural networks can be made indifferent to both affine and non-linear transformations [1] [13].

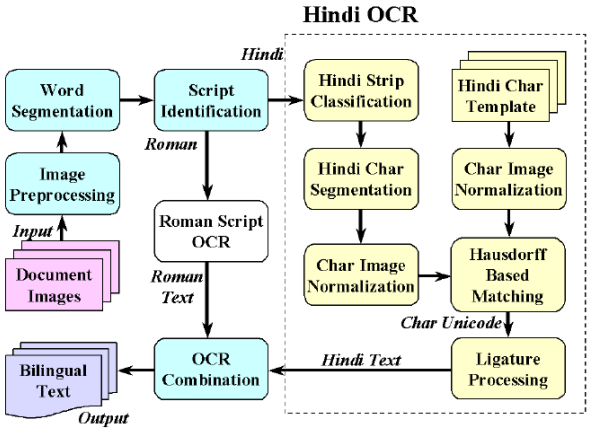


Figure 1: System Architecture

In Figure -1, Images are first preprocessed with denoising and deskewing [5]. After processing the recognized Hindi words, the output of the Hindi OCR will be combined with the OCR output of Roman script to provide a complete result.

B. FEATURE EXTRACTION BY THE SLIDING WINDOW METHOD

Feature extraction is the identification of appropriate measures to characterize the component images distinctly. Here and in what follows we assume that recognition of a full line is to be performed: in the postal domain such a line can contain a name, a street address or a city, state, and zip. For the sliding window method, the line image is first height-normalized to 64 pixels. Since the standard algorithms for scaling a bi-level image yield grey scale output, some form of binarization becomes necessary even for those images that start out as binary on the CEDAR CD-ROM [6]. We investigated two scaling algorithms, bilinear and bi-cubic resampling. Linear resampling was faster, but only marginally, and the results were indistinguishable as far as later stages and overall recognition accuracy are concerned. We also investigated two different binarization methods, global thresholding and local (adaptive) thresholding [11]. Here the impact was marked: local thresholding was considerably better than global. To obtain comparable results for the grey scale CEDAR data, we added a binarization stage prior to scaling, and investigated all eight combinations: first local vs. global thresholding, next bilinear vs. bi-cubic resampling, and finally again local vs. global thresholding. To summarize our results, the less global thresholding done the better. The scaled (re)binarized image is noise-cleaned: in particular underscores and dashed underscores are removed. Manual inspection reveals that this process leaves no visible traces of removal in about 97% of the cases. Next, the top and bottom of the writing is computed for every eight pixel wide column, and the results are low-pass filtered over five columns encompassing 40 vertical pixels. This results in images where ascenders or descenders are never cut off, but if they are not present, the 'n' zone, i.e. the body of the ascender- and descenders-less letters such as a c e i m n o r s u v w x z, fills the whole image. The effect of getting more of the n-zone into the feature vectors is enhanced by the next stage of nonlinear resampling, whereby a column of 64 vertical pixels is reduced to a set of 8 grey scale values, but the values close to the center are based on fewer pixels than far from the center. In effect, we condense the image by a factor of 12 at the edges, but only by a factor of 4 in the center. By this process, the image strip has been replaced by a sequence of 8-dimensional real vectors whose progression in time corresponds to progression in space along the x axis of the original line image. In the final stage of feature extraction this sequence v0, v1... vk is first triplicated, meaning that each vi is replaced by a concatenation vi−1, vi, vi+1. This yields a sequence of 24-dimensional vectors which are reduced by principal component analysis to 12-16 dimensions [2][3]. The main additions to our earlier feature extraction method which was successful in a bank check recognition system [10] are the initial global height normalization and the low pass filtering of local height.

C. FEATURE EXTRACTION BY SEGMENTATION

In the zip code field, numbers are rarely touching, and one field, such as state, rarely touches another, such as city or zip. These observations lead to a system where the primary unit of analysis is the connected component, and the expectation that word boundaries will also be connected component boundaries. Though this expectation is largely met, both cursive writing and touching handprint require that connected components be cut into smaller parts that we will call frags. In the segmentation algorithm developed by Jianchang Mao and Prasun Sinha at IBM Almaden, and used in the experiments, the central heuristic used for the cutting is the location of valleys (local minima) in the contour. Feature vector is computed for each frags using the contour direction features described in [9][12]. Even that the goal of the algorithm is to presegment characters, it is not surprising that for the majority of characters it creates a single frag, and therefore a single feature vector. The system over segments, but only slightly: over two thirds of character tokens yield a single frag, 20% yield two, and less than .15% yields four or more. The average

1999-4974©2010BUJICT

number of frags per character is 1.24, which makes single state character models a natural choice. This is in sharp contrast to the sliding window system, which must take into account that the width of characters varies widely, both within and across character classes, even after height normalization. To deal with across-class width variation, in the sliding window system models for different characters have different numbers of states ranging from 1 for dot (period) and 2 for i to 6 for w and 7 for m. For most characters we use Bakis models, with a self-loop for each state, a step transition to the next state, and a jump transition to the second state following it. If we enrich the model with a silent input state with transitions to any subsequent state [4], it becomes possible to fully absorb arbitrary width variation [8][7], and in certain classes, such as u or s, width variation was so extreme that we found it advantageous to do so. To summarize the differences between the two feature extraction methods, in the sliding window system many (often more than a dozen) relatively low dimensional feature vectors are extracted for each character, while in the pre segmentation-based system only a few (typically only one) vector will be extracted. Since this vector is much bigger (originally 88 dimensions, in most experiments reduced to 48 or 32 dimensions by principal component analysis), the overall bit rate of the two feature extraction front ends is quite similar, about 4-8 bytes per horizontal pixel. This bit rate, being an order of magnitude larger than that required for pen- or tablet-based recognition, offers a rough measure of the difficulty of extracting dynamic information from an image[13].

II. PROBLEM STATEMENT

A few OCRs are available for different Indian scripts. Most of them use "Feature Extraction Technique" for character recognition but they do not provide reliable output due to the following challenges: -

- Scalability & Shape of characters: The characters differ in sizes, shapes and styles from time to time and book to book as the technology and fonts for the printed text changes.
- Broken Characters: It is most challenging to recognize broken characters as these can not be recognized with feature extraction method which is being used for Indian languages.
- Noise in scanned or printed image: Noise may be introduced while printing or scanning the documents. It becomes tougher if the noise distort the text.
- Inclined/Skewed Characters: OCR misrecognizes or gives a garbage output for inclined characters. Skew detection & correction techniques are only applicable for the whole text area and not for individual character or word.

i. Small characters & punctuation marks: These are still a big challenge for Indian scripts OCR as punctuation marks and small characters like MATRAS are not recognized correctly.

ii. Identification of Indian languages scripts is challenging problems. In Optical Character Recognition [OCR], machine printed or handwritten characters/numerals are recognized. There are plentiful approaches that deal with problem of detection of numerals/character depending on the sort of feature extracted and different way of extracting them.

In Optical Character Recognition [OCR], a character numerals which has to recognized can be machine printed or handwritten characters/numerals [1]. There is extensive work in the field of handwriting recognition, and a number of reviews exist. Handwritten numeral recognition is an exigent task due to the restricted shape variant, unusual script style & different kind of noise that breaks the strokes in number or changes their topology [1]. Recognize of is gaining wider importance today & is one of the benchmark problem in document analysis. As handwriting varies when person write a same character twice, one can expect enormous dissimilarity among people. These are the reason that made researchers to find techniques that will improve the knack of computers to characterize and recognize handwritten numerals. Here it is not the end of problems. They are many more regarding character recognition.

III. METHODOLOGY

Research and development in the area of printed Indian script recognition was taken up by our group to address many specific and generic applications which need an OCR engine.

For increasing the efficiency of OCR system we need to develop a mechanism for proper and accurate character recognition. Here we propose a hybrid approach which uses "Skeleton", "Feature Detection & Categorization" and "Artificial Neural Network" methods to precisely recognize the character. All of the approaches have their own limitations but play a better role while using their combination and may give a very efficient output for Indian script.

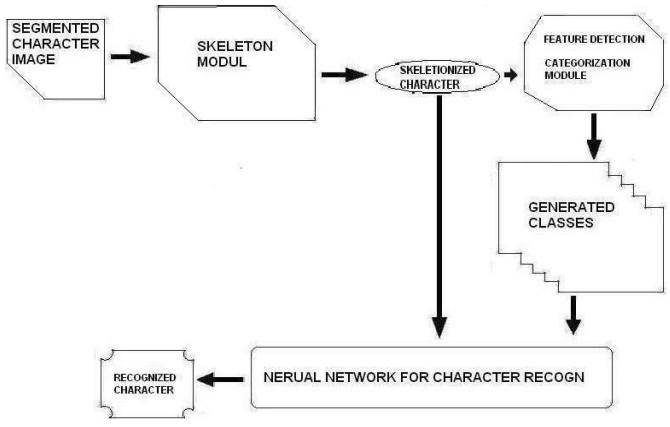


Figure 2: "Proposed System for Individual Character Reorganization"

Basically first the image of segmented character is sent to the "Skeleton Module" which generates the skeleton of that character. Now this skeleton is categorized on the basis of the characteristics & assigned to a particular class. These classes

1999-4974©2010BUJICT

are further used by a neural network to select the particular character among the class members for making decision regarding accurate selection of particular character among the class members. [13] [14]. . These classes, generated by these modules might be result oriented of our new character concern for individual character reorganization.

IV. SKELETONIZATION

Skeletonization is the process of peeling off of a pattern as many pixels as possible without affecting the general shape of the pattern. In other words, after pixels have been peeled off, the pattern should still be recognized. The skeleton hence obtained must have the following properties:

i. As thin as possible
ii. Connected
iii. Centered

Main advantage of skeleton zing is to reduce the amount of data required for transmitting the pattern without lose of safe information as well as to retain the characteristics of the shape useful for the specific structural description.The skeleton exhibits a very robust representation of local specificities of given shape and are not very sensitive to pattern distortions.

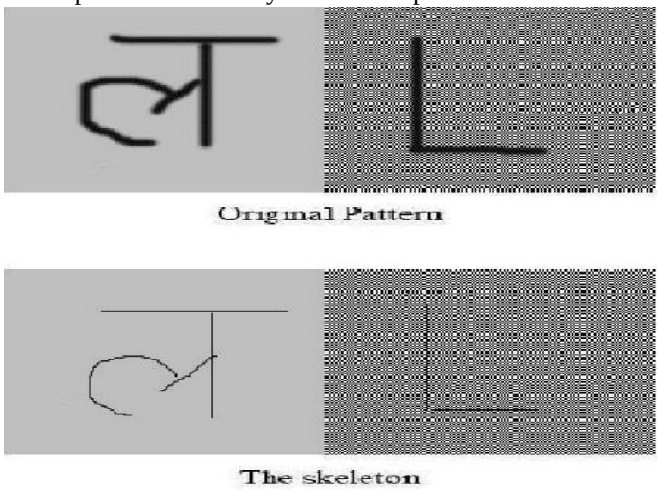
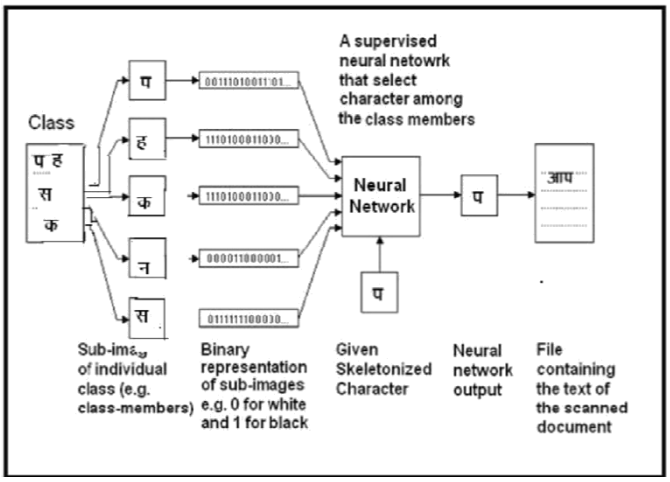


Figure 2 : " Example of A Skeltonized Image"

This approach is most suitable for Indian scripts because of the complexness of the scripts.

V. FEATURE DETECTION & CATEGORIZATION:

Different feature of given skeleton will be calculated like the aspect ratio, loops in the shape, curves and other attributes. The features are calculated irrespective of size of given character skeleton. On the basis of this a number of classes will be generated. Each class has its unique feature set. After categorization it will be easier to recognize them on the basis of their specific class characteristics [13].



Artificial Neural Network

Figure 3: "ARTIFICAL NEURAL NETWORK"

In this approach neural network is not used for pattern recognition like other OCRs but for making decision regarding accurate selection of particular character among the class members. This selection will be based on a complex neural network trained under supervised learning mechanism.

VI. CONCLUSION

This approach provides a concrete theoretical solution to the challenges mentioned in the problem statement. Size of given character doesn't matter as the aspect ratio of skeleton will be used. Every character has its own unique skeleton so it will be easier to distinguish it from other characters' skeletons. Broken characters which are a big problem in other OCRs can be easily handled here as the skeleton of the broken character will be very near to only that particular character. So it can be identified. The skeletons of inclined characters can be straightened easily. Small characters & punctuation marks have their particular skeleton. So that would help to consider as a character instead as a noise. Here the distance from the text area would play a measure role to consider them as a character. The above approach, attempted by us might be result oriented of our concern but things are still need to be done. This is only a hypothesis and still this approach needs to be practically resolved. For this, different algorithms for different methods related to this approach are available but they are to be combined together and a step by step method should be applied. We hope to come out with beautiful conclusions in the coming near future[11][13].

REFERENCES

[1] J.K. Baker, "Stochastic modeling for automatic speech understanding" Reprinted in A. Waibel and Kai-Fu Lee (eds) Readings in Speech recognition, Morgan Kaufmann, San Mateo CA, 1990, 297-307
[2] J.R. Bellegarda, D. Nahamoo, K.S. Nathan and E.J. Bellegarda, "Supervised Hidden Markov Modeling for On-line Handwriting Recognition" IEEE Proc. ICASSP, Adelaide 1994, Vol 5, 149- 152
[3] M.R. Bokser, "State of the Art in a Commercial OCR System: A Retrospective View" Paper presented at the IS&T/SPIE Symposium on Electronic Imaging, San Jose CA 1996

1999-4974©2010BUJICT

[4] T.H. Crystal and A.S. House, "Segmental durations in connected speech signals: current results" JASA 83 1988, 1553-1573
[5] A.J. Elms, The representation and recognition of text using Hidden Markov Models. University of Surrey PhD Thesis, 1996
[6] J.J. Hull, "Database for handwritten word recognition research" IEEE PAMI 16 1994, 550-554
[7] A. Kornai Formal Phonology. Garland Publishing, New York, 1995
[8] A. Kornai and S.D. Connell, "Statistical Zone Finding" IEEE Proc. 13th ICPR, Vienna 1996, Vol III, 818-822
[9] A. Kornai, K.M. Mohiuddin and S.D. Connell, "An HMM-based legal amount field OCR system for checks" IEEE Proc. SMC, Vancouver, BC 1995 Vol 3, 2800-2805
[10] A. Kornai, K.M. Mohiuddin and S.D. Connell, "Recognition of cursive writing on personal checks" Proc. 5th IWFHR, Essex 1996, 373-378
[11] K.M. Mohiuddin and J. Mao, "A Comparative Study of Different Classifiers for Hand printed Character Recognition" Pattern Recognition in Practice IV, 1994, 437-448
[12] H. Takahashi, "A Neural Net OCR Using Geometrical and Zonal-pattern Features" Proc. 1st ICDAR, 1991, 821-828
[13] Character Recognition using Neural Networks" by Deepayan Sarkar in the Proceedings of ICDIT'08.
[14] W.A.Woods et al "Speech understanding systems: final technical progress report" Bolt Beranek and Newman Inc. Report 3438, Cambridge MA, 1976.