International Journal of Information and Computation Technology.
ISSN 0974-2239 Volume 4, Number 3 (2014), pp. 259-264

http://www. irphouse.com /ijict.htm

Graph Based Evolutionary Approach to Classify Irregular Hindi Characters

Prateek Mishra[1] and Md. Tanwir Uddin Haider[2]

Department of Computer Science & Engineering,
National Institute of Technology Patna, Patna, Bihar, India.

Abstract

Devanagari Hindi is the most popular language in India. There are many conjuncts and compound characters in Hindi. A character may be written in many different styles. There may be various irregularities in the text while writing. These irregularities make machine recognition difficult and affect the performance of the recognition system. In this paper, the concept of Genetic Algorithm has been used for efficient recognition of irregular Hindi characters. The motivation behind using Genetics comes from the fact that different styles of characters can be associated with different possible irregularities. Furthermore these styles can be genetically combined to produce various new, unknown and irregular styles. In this paper, each character is represented by a graph. A genetic pool is developed by creating different regular and irregular characters with the help of genetic operations and possible irregularities. This pool is used to search the best possible match for an input graph. Genetic Algorithm can be more useful when the sample data is very large and the input data is rich with irregularities. In this paper, some cases where irregular characters may lead to poor recognition rate are also discussed and an approach using Genetic Algorithm is proposed to deal with this problem.

Keywords: Hindi Character Recognition, Genetic Algorithm, Irregular Devanagari Characters, Graph Theory.

1. Introduction

Hindi is the national language of India which is written in Devanagari Script. Over 300 million people in India, both in urban and rural areas, use Hindi language for mutual conversation and for documentation. In Hindi, a character may be written in different styles and in irregular manner which is very tough to be understood by the computer system. That is why the Optical Character Recognition (OCR) system tries to empower the computer system with the Human ability of recognizing the characters of a language (Pal, 2007). In the field of computational intelligence an improvements has been achieved but areas like computer vision are still to produce some sophisticated results. In case of character recognition, the problem is very challenging if the data contains many irregularities and the writing style is not known to the machine.

Some simple and common irregularities caused due to careless writing may lead to reduce the recognition rate of a recognition system dramatically (Kumar S., 2010). No matter what classification technique you have used irregular characters are pretty tough to deal with. This paper has been arranged in six Sections. Section 2 describes the features of Devanagari script and some common irregularities while writing Devanagari. Section 3 describes the major stages in an OCR system. Section 4 describes an approach towards the problem using Genetic Algorithm. Finally, Conclusion and Future research direction are discussed in Section 5 and References are listed in section 6.

2. Devanagari Hindi Language

2.1 Basic Features

India is a multi-script and a multi-lingual country. Devanagari is the most popular script in India. Devanagari Hindi has 44 basic characters comprising 11 vowels and 33 consonants (Ishida, 2002). Vowels can also be used as *modifiers* with consonants to form *conjuncts.* Also, two consonants can be combined to create *compound characters.* Each Hindi word contains one Headline at its upper part which is called as *Shirorekha*. Shirorekha is one of the important properties of Hindi language which separates it from other language. So developing a multi-lingual recognition system; *Shirorekha* can be used to identify Hindi words among others. These Modifiers are also called as *Vowel Symbols* or *Matra*. Modifiers and some combined characters are shown in table I and table II (Dongre, 2010).

Table 1: Vowels and Corresponding Modifiers.

Vowels:	अ	आ	इ	ई	उ	ऊ	ऋ	ए	ऐ	ओ	औ
Modifiers:		ा	ि	ी	ु	ू	ृ	े	ै	ो	ौ

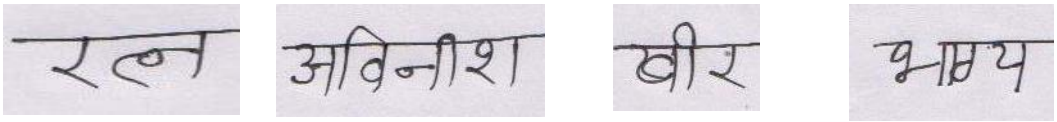
Table 2: Combination of two characters (Combined Characters)

क क्क	क ल क्ल	घ न घ्न	च ञ च्ञ	ञ च ञ्च	त न त्न	प त प्त	प ल प्ल
ब व ब्व	भ न भ्न	म ल म्ल	ल ल ल्ल	श न श्न	श व श्व	श ल श्ल	स न स्न

These conjuncts, modifiers and combined characters along with some irregularities make recognition system for such language very difficult. Moreover, there are some characters in Devanagari Hindi which looks very similar in shape. A fraction of error while writing them may create an ambiguity for the machine.

2.2 Common Irregularities

Touching and Fused Characters: This is very common error while writing Hindi and very difficult to deal with. This sort of irregularity can be found in all three zones of a Hindi character. That is two characters can be fused or touched with each other shown in figure 2a and 2c; upper and lower vowel symbols can be fused with character shown in figure 2b; an intra-character stroke touch can be there shown in figure 2d etc.



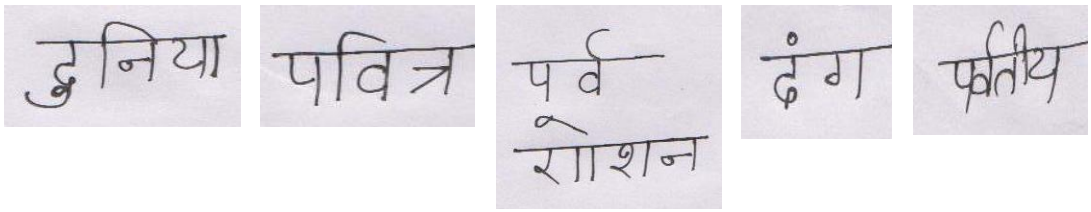
(A) **(b)** **(c)** **(d)**

Fig. 2: Some touching and fused characters (a) Fused characters (b) Fused upper vowel symbol (c) Touching characters (d) Touching internal strokes

Fusion of two characters creates confusion for classifier. Even the segmentation of two narrow characters is very cumbersome.

Merging Constant with Lower Modifier: Sometimes a character is chomped with the Modifier in such a way that it completely changes the appearance of the character produced. Judging this type of irregularity is very difficult. Figure 3a shows the fused lower vowel symbol.

Incomplete Representation of Modifiers: While writing hurriedly sometimes upper modifiers are left undeveloped. These undeveloped modifiers create ambiguity. Figure 3b shows the incomplete vowel symbol.



(a) **(b)** **(c)** **(d)** **(e)**

Fig. 3: (a) Fused lower vowel symbol (b) Incomplete vowel symbol (c) Isolated Modifier *(d) Incomplete Character (e) Narrow Writing*

Writing modifier in isolation: If writing carelessly a modifier could be written in such a way that it appears independent of its corresponding character. It may create ambiguity that whether the modifier belongs to the character in upper line or to the character in the lower line. Figure 4a shows an isolated modifier.

Incomplete Characters: Sometimes a character is not written to its complete shape. This creates an illusion for a character of being another character. Figure 4b shows an undeveloped character.

Narrow Writing: Some people write in such a way that the gap between two characters is very less (characters are touched). This creates problem while segmenting these characters (Bansal and Sinha, 1997). Figure 4c shows narrow writing.

3. Basic Optical Character Recognition (OCR) System

Generally, there are five major stages in an OCR System (Russ, 1998):

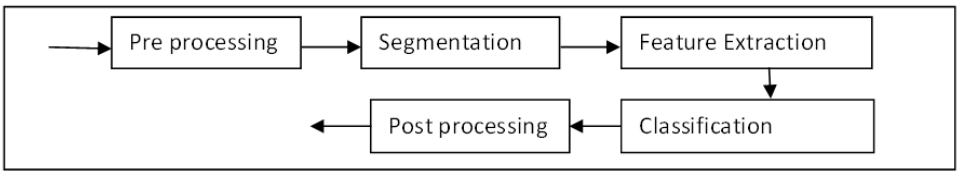


Fig. 4: Basic Steps in OCR System.

Since recognition of a character with different irregularities is the central concern. In this paper, it is assumed that the above steps are already done. That is the document has gone through various processing steps. The Image is enhanced; noises are eliminated; skewed characters have been made horizontal; characters are thinned and their size is normalized. The input image is assumed to be properly segmented assuming that the characters are left without being segmented if it is too tough (almost impossible) to segment. In this paper Genetic Algorithm is used for classification purpose.

Post-processing: When the recognition of a character is done this character is saved for post-processing to provide a semantic output. This stage is important in an OCR system as the recognition is meaningless if the output is not producing any sense. The output can be spell checked using a Devanagari dictionary and words can be edited. However, in real life scenario this is not very feasible because in India many Hindi words do not have meanings even in dictionary. E.g. name of a village or a city sometimes seems completely meaningless. This creates ambiguity for an editor. This is why developing a semantic editor for Devanagari is very hard.

4. Methodology

Overview of Genetic Algorithm: Genetic Algorithm is based on Darwinian paradigm of survival of the fittest and evolution of species by means of natural selection (Darwin, 1859). An individual is a feasible solution to the problem. Each individual is characterized by a Fitness function. Fitness Function is the measurement of an individual's struggle for life. Higher fitness is better solution. Based on their fitness, parents are selected to reproduce offspring for a new generation. Fitter individuals have more chance to reproduce. New generation has same size as old generation and

the old generation dies. Offspring has combination of properties of two parents (Mitchell, 1996) (Sivanandam, 2008).

Proposed Methodology: Since its evolution, Genetic Algorithm has been applied to the problems of various domains. In the field of character recognition Genetic Algorithm has been used to produce new and relatively unknown styles by combining two different styles. In cases where too many irregular characters are present Genetic Algorithm will prove very handy to recognize characters more accurately by associating different possible irregularities and creating different offspring.

Success of the genetic algorithm relies majorly on fitness function (condition), the representation mechanism of the chromosomes and the versatility of training data. However, Mutation should be considered the most interesting and powerful operation because it causes some very random and out of the pattern change in the next generation. In our approach each chromosome will be represented by a graph. The overall methodology is as follows:

After the segmentation part, the obtained binary images are converted into graphs. Prior to all this, two data sets are created. One contains the graphs of all Hindi characters written in correct and regular shape and other contains different graphs of the different characters written with possible irregularities. A complete and large Genetic pool (training data) is created (initial population) by applying genetic operations on the previous two data sets. This pool is used to search the best possible match for an input graph.

Now the graph of input character is matched with this data set. A probabilistic analysis is done while matching and if the match is successful then this input graph is also added to the training data set. The time when this training set becomes too large genetic algorithm is applied over this set to optimize the set and to select relatively fitter chromosomes using the fitness criteria.

Due to large size of the genetic pool, the pool is divided into clusters. Indexing is used to retrieve the appropriate cluster(s). As the input graph arrives, the graph is superimposed on a 3*3 grid. Now with each region of this grid, probability distribution fitting is performed to map this graph to a cluster. When one complete word is detected, a runtime backend dictionary is used to obtain semantic results.

Now moving towards the other case; if there are some words which cannot be segmented properly then they are passed to the classifier as one single input. Algorithm is used separately for these inputs.

5. Conclusions and Future Scope

India is a multilingual country so processing the document written in Indian languages is an interesting and ambitious problem to solve. Generally, a writer commits a lot of mistakes while writing Devanagari words. These writing discrepancies dramatically reduce the recognition rate. As a matter of fact, we cannot teach people to avoid bad writing. Therefore, a sophisticated approach is required to tackle the situation. Genetic Algorithm is one such approach.

An Intelligent OCR will be developed by applying the evolutionary approach in different stages of an OCR such as Image Enhancement, Feature Extraction and feature optimization and Classification. In this paper one such approach is proposed where different graphs will be created to represent chromosomes**.** Different graphs are obtained by applying genetic operations among different styles of regular and irregular graphs.

The efficiency of a classifier depends on how strong and versatile the training data is. But managing a large training data is cumbersome. A lot of research work has been done in the area of character recognition but processing the document written in local languages is still a challenge.

In our case also, there are some possible improvements which can be treated as the scope for future work. E.g. if the input document is very large then generating graphs for each character and matching them in optimal time is a challenge. Some sophisticated indexing mechanism is desirable to serve this purpose.

References

[1] U. Pal, N. Sharma, T. Wakabayashi and F. Kimura (2007), "Off-Line handwritten character recognition of devanagari script", Ninth International Conference on Document Analysis and Recognition.

[2] Vikas J Dongre, Vijay H Mankar (2010), "A review of research on devanagari character recognition", International Journal of Computer Applications, Volume 12– No.2, pp (0975–8887).

[3] Satish Kumar (2010), "An analysis of irregularities in devanagari script writing–a machine recognition perspective", International Journal on Computer Science and Engineering, Vol. 2, No. 2, pp.274-279.

[4] Richard Ishida, "An introduction to indic scripts," Available: http://www.w3.org/2002/Talks/09-ri-indic/indic-paper.pdf.

[5] Veena Bansal and R.M.K. Sinha (1997), "Segmentation of touching and fused devanagari characters", Technical Report, TRCS-97-247, I.I.T. Kanpur, India.

[6] Darwin Charles (1859), The Origin of Species, reprinted 1985, Penguin, London Available: http://www.tbi.univie.ac.ar/Origin/index.html

[7] M. Mitchell (1996), "An introduction to genetic algorithms", The MIT Press.

[8] Russ, John C. (1998), "The Image Processing handbook". Boca Raton, FL: CRC Press, 3rd Edition.

[9] S. N. Sivanandam, S. Deepa (2008), "Introduction to Genetic Algorithm", Springer.