A Recognition System for Devnagri and English Handwritten Numerals

G S Lehal[1] and Nivedan Bhatt[2]
[1]Department of Computer Science & Engineering, Thapar Institute of Engineering & Technology, Patiala, INDIA.
gslehal@mailcity.com
[2]Department of Computer Science & Engineering, Punjabi University, Patiala, INDIA.
nivedan@mailcity.com

Abstract

A system is proposed to recognize handwritten numerals in both Devnagri (Hindi) and English. It is assumed at a time the numerals will be of one of the above two scripts and there are no mixed script numerals in an input string. A set of global and local features, which are derived from the right and left projection profiles of the numeral image, are used. During experiments it was found that the Devnagri numeral set had a much better recognition and rejection rate as compared to the English character set and so the input numeral is first tested by the Devnagri module. The correct recognition enables the system to be set to the appropriate context (Devnagri/English numeral set). Subsequent identification of the other numerals is carried out in that context only.

1. Introduction

Handwritten numeral recognition has been extensively studied for many years and a number of techniques have been proposed [1-5]. However, handwritten character recognition is still a difficult task in which human beings perform much better. The problem of automatic recognition of handwritten bilingual numerals is even more tough. Recently there has been a growing interest in script and language identification for developing international OCRs for multi-lingual scripts[6-9]. A relevant research work in the field of bilingual character recognition of Indian scripts is by Chaudhary and Pal[9].They have developed an OCR system to read two Indian language scripts: Bangla and Devnagri. A stoke feature based tree classifier is used to recognize the basic characters. For some characters additional heuristics are used. In a multilingual environment like India, there are many situations where a single document may contain handwritten numerals in two or more language scripts. As English and Hindi are the two most prominent languages used in India, an integrated approach towards the recognition of numerals of both Devnagri and English script is helpful in OCR development in India. Also such approach will have commercial applications in postal and banking environments. In this paper a bilingual OCR system for isolated handwritten numerals of Devnagri(Hindi) and Roman scripts has been presented. To the best of our knowledge, this is the first paper on recognition of bilingual handwritten numerals in Devnagri and English.

This paper is based on the work done by Kimura and Shridhar[1]. Kimura and Shridhar have presented two algorithms to achieve good classification rates of handwritten numerals. The first algorithm is a statistical one which used a modified discrimination function(MQDF) with features derived from chain codes of the character contour and the second one is the structural algorithm which uses the left and right profile features. We have modified the structural classifier used in the above mentioned paper for the recognition of numerals of both Devnagri and English scripts. The use of such a classifier has allowed a use of a variety of different feature descriptors, which more precisely describe the character and yet remain under the domain of same classifier. Also the use of a single classification scheme has enabled fast and flexible learning.

2. Data processing for feature extraction

The input samples were taken in a specially designed form, which contained sets of guide boxes. The use of guide boxes for the samples allowed us to avoid the segmentation phase completely as the input was already in an isolated form. Some of the samples of handwritten Devnagri and English numerals are shown in fig.1. The form was processed by a scanner at 300dpi and gray-tone to two-tone was automatically performed by the scanner software. Each numeral occupied a field that measured 60Wx77H pixel units. Then a contour extraction algorithm was applied to obtain the closed contours of the numerals. As the input numerals can be of arbitrary size, the scaling factor had to be dynamically adjusted and then the normalization was applied to the numeral contour so that the enclosing frame measured 60Wx77H pixel units. The width varied from a minimum of 5 pixels to 60 pixels. This minimum width of 5-6pixel units(referred to as pen width) will be one of several feature descriptors used for the recognizing the numerals.

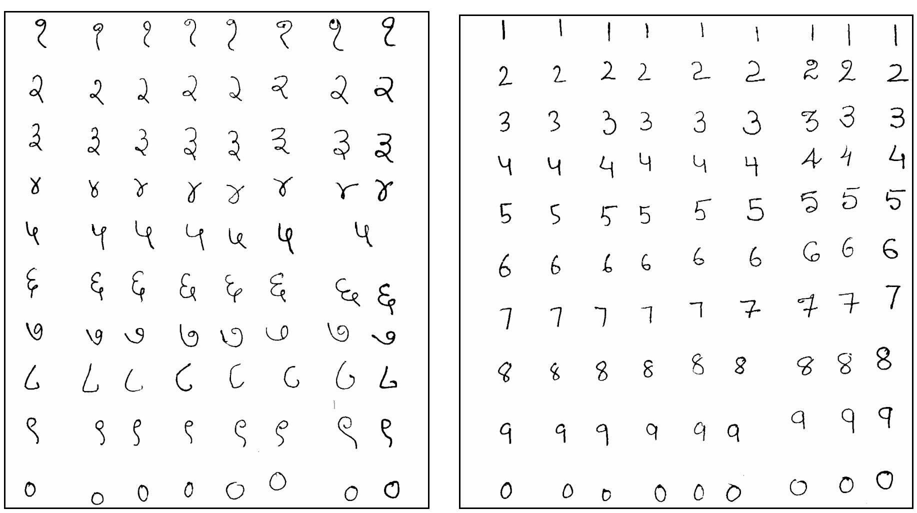


Fig. 1. Samples of the handwritten Devnagri and English numerals.

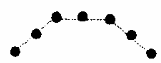
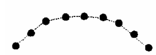
3. Feature Extraction

The input for the feature extraction phase is the normalized version of the candidate numeral's contour. The leftmost and rightmost foreground values are searched and the contour is split at those points to yield left are right profiles respectively. All the features required in our work are obtained from the shape variations in the profiles. The left profile lp[il and the right profile rp[i] are nothing but the collection of distances from the left and right edges. i ranges from 1 to the length of the numeral. These are the global features. In addition to these there is an additional set of global features called the first differences in the left and right profiles i.e. ldiff[i] and rdiff[i] where i ranges from 2 to the length of the numeral.

In addition to these, we need local features to account for local variations in the shape of the contour. To extract these features, the numeral contour is visualized as a complex structure which is composed of individual shape descriptors(features).The various categories of such features are described below:

CATEGORY 1)

Shape_ 1.1) A sequential occurrence of a minimum, maximum and a minimum i.e. rpmin, rpmax, rpmin in the right profile generates shape descnptors of type *mountain, plateau* and *bump* in the right profile for a particular range R1.

(a) mountain  (b) plateau  (C) bump 

Shape_ 1.2) A sequential occurrence of a minimum, maximum and a minimum i.e. Ipmin, Ipmax, lpmin in the left profile generates shape descriptors of type *mountain, plateau* and *bump* in the left profile. for a particular range R1.

Shape_ 1.3) A sequential occurrence of a maximum, minimum and a maximum i.e. rpmax, rpmin, rpmax in the right profile generates shape descriptors of type downward *mountain, plateau* and *bump* in the right profile for a particular range R2

a) mountain  b)plateau  c) bump 

Shape_ 1.4) A sequential occurrence of a maximum, minimum and a maximum i.e. Ipmax, lpmin, lpmax in the left profile generates shape descriptors of type downward *mountain, plateau* and *bump* in the left profile for a particular range R2.

CATEGORY 2)

Shape_2.1) A sequential occurrence of a minimum and a maximum i.e. rpmin, rpmax in the right profile generates shape descriptors of type *ascender* in the right profile for a particular range R1. (a) ascender 

Shape_2.2) A sequential occurrence of a minimum and a maximum i.e. lpmin, lpmax in the left profile generates shape descriptors of type *ascender* in the left profile for a particular range R1.

Shape_2.3) A sequential occurrence of a maximum and a minimum i.e. rpmax, rpmin in the right profile generates shape descriptors of type *descender* in the right profile for a particular range R2. (a)descender 

Shape_2.4) A sequential occurrence of a maximum and a minimum i.e. lpmax, lpmin in the left profile generates shape descriptors of type *descender* in the left profile for a particular range R2.

CATEGORY 3)

Shape_3.1) Discontinuity(successive pixels are widely separated) in the right profile for a particular range R1.

Shape_3.2) Discontinuity(successive pixels are widely separated) in the left profile for a particular range R1.

Shape_3.3) Sharp discontinuity in the right profile for a particular range R1.

Shape_3.4) Sharp discontinuity in the left profile for a particular range R2.

Shape_3.5)A smooth(relatively less discontinues) right profile for a particular range R1

Shape_3.6)A smooth(relatively less discontinues)left profile for a particular range R2.

CATEGORY 4)

Shape_4.1) Occurs when the right profile is more or less a straight line i.e. the difference between rpmax and rpmin is a small value for a particular range R1.

Shape_4.2) Occurs when the left profile is more or less a straight line i.e. the difference between lpmax and lpmin is a small value for a particular range R2.

Shape_4.3) Occurs when the difference between rpmax and rpmin is greater than a specific value for a particular range R1.

Shape_4.4) Occurs when the difference between lpmax and lpmin is greater than a specific value for a particular range R2.

CATEGORY 5) Width Features:

Shape_5.1) These occur when the width of the contour w[i]=lp[il - rp[il is approximately equal to the pen-width for a particular range R1.

Shape_5.2) These occur when the contour width stays nearly the same for particular range R2.

Shape_5.3) These occur when the width at the particular point is greater than any other particular point.

Shape_5.4) These occur when the width of the contour approaches the standard width value for the normalized version (which is 60 in our case).

CATEGORY 6)

Shape_6.1) These occur when the ratio of length to width is less than a particular value.

In order to classify the members of each numeral subclass, distinct groups containing features describing the numerals have to be found. It is worth mentioning that in each group of shape features of all the numeral subclasses there are two kinds of features present. First are those features, which uniquely define that particular numeral. Second are those, which are used specifically to avoid the confusion with the other similar numerals of its own or other character set. Although they may seem trivial in the manner they describe the numeral but in some cases they act as a decisive key for differentiating the two confusing numerals. For example the feature (shape_2.1) in the range 70 to 75 helps differentiating the Devnagri 3 from the English 3. The feature (shape_5.4) in the range 1 to 25 helps to differentiate between the Devnagri 4 and the distorted sample of English 8. Also the features (shape_1.2) in the range 1 to 55 and (shape 2.2) in range 40 to 75 helps differentiating a Devnagri 5 from a distorted English 4 and so on.

All the possible features for each numeral subclass are grouped together. These form the classified character groups for the numerals. The recognition process uses a tree structure to establish the identity of the numeral candidate. The character groups were kept as distinct as possible to enhance the between -class variability and while minimizing the within - class variability. Role played by the context in conjunction with the recognition process is explained in the next section. Groups for all the numeral subclasses of both the character sets have been tabulated in the appendix.

A verbal explanation for the group of features for Devnagri 3 is given below (Fig. 2).

I. It has a bump feature (category 1.4) in its left profile in the range 40 to 75.
II. It has a bump feature (category 1.1) in its right profile in the range 1 to 40 and 40 to 65.
III. It has an ascender feature (category 2.1) in its right profile in the range 1 to 10.
IV. It has a discontinuity (category 3.2) in its left profile in the ranges 1 to 25 and 40 to 75.
V. It has an ascender feature (category 2.1) in the range of 70 to 75.

4. Identification of the Script of the Numerals

It is assumed that we are dealing with situations where at a time we shall be having the numeral candidates of a particular script only. For identification of script, no separate routine is used. During the recognition process when the first numeral of a particular guidebox set (Fig 3) is recognized correctly, the context (Devnagri/English) is set to the domain of that particular numeral's character set. Subsequent identification of the remaining numerals in that particular guidebox set is carried out in that context only which drastically reduces the search space and hence increases the performance of the system. It was observed that the Devnagri numeral set had a very good recognition and rejection rate, as compared to the English set. Also the Devnagri numeral set's recognition module had good rejection rates for the numerals of the English character set. This property was exploited by adopting a *polling strategy* in which the input numeral is first tested by the Devnagri module. If the numeral is recognized then the context is set to Devnagri, else it is tested for English set and on recognition, the context is set to English. In case the numeral is rejected by both script sets, then the next numeral is tested, and this continues till one of the numeral is recognized. Subsequent identification of the other numerals in that particular guidebox set is carried out for character set of the recognized numeral. Numeral 0 is the same for both the character sets, thus in the case when the first numeral encountered in a particular guidebox set is a zero, subsequent numeral is checked before deciding the context

5. Results and Discussion

The performance of an OCR system is evaluated in terms of the parameters: recognition rate, confusion rate (indicating multiple membership of the numeral) and the rejection rate for a particular data set. In our case the results were tested on 1000 samples(in guideboxes) of both the Devnagri and English character set. For the Devnagri numeral set, a recognition rate of 89% and a confusion rate of 4.5% were obtained. The reason of the good performance for the Devnagri numerals is that these numerals are almost distinct from each other, thus increasing the inter-numeral subclass variability. Also the Devnagri numeral set's recognition module had good rejection rates for the numerals of the English character set. For the English numeral set we had a recognition rate of 78.4%, confusion rate of 18 % and rejection rate of 3.6%. The performance of the system can further be improved by including more features for English character set.

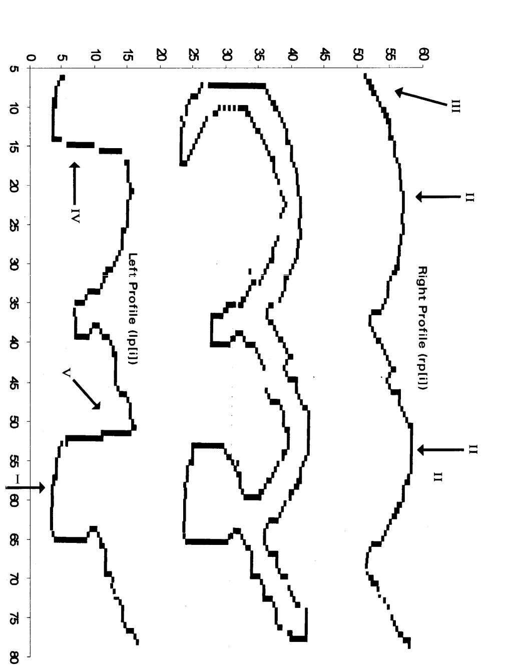


Fig 2: Projection profile of Devnagri 3

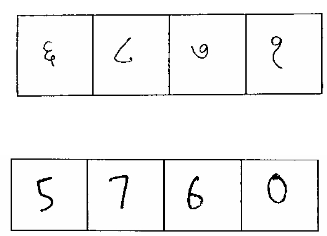


Fig. 3 : Guidebox containing (a) Devnagri and (b) English numerals

6. Conclusion

A recognition system that can read both Devnagri and English handwritten numerals has been described. The performance of the system for Devnagri character set is quite good but we are not so satisfied with the recognition rate for English character set. Work is currently being carried out for improving the performance for English character set.

References

1. Kimura, F., Shridhar, M. : Handwritten numeral recognition based on multiple algorithms, Pattern Recognition (1991) 976-983.

2. Cao, J., Ahmedi, M., Shridhar, M. : Recognition of handwritten numerals with multiple features and multistage classifier, Pattern Recognition (1995) 153-160.
3. Chi, Z., Wu, J., Yan, H.: Handwritten numeral recognition using self-organizing maps and fuzzy rules Pattern Recognition (1995) 59-66.
4. Lee, L., L., Lizarrage, M., G., Gomes, N., R., Koerich, A.: A prototype for Brazilian bankcheck recognition, International Journal of Pattern Recognition and Artificial Intelligence (1997) 469-487.
5. Stefano, C., D., Cioppa, A., D., Marcelli, A. : Handwritten numeral recognition by means of evolutionary algorithms, Proceedings ICDAR99, Bangalore, India, (1999) 804-807.
6. Spitz, A., L. :, Determination of script and language content of document images, IEEE Transactions on Pattern Analysis and Machine Intelligence (1997) 235-245.
7. Ding, J., Lam, L., Suen, C., Y.: Classification of Oriental and European scripts by using characteristic features, Proceedings ICDAR97, Ulm, Germany (1997) 1023-1027.
8. Lee, D., S., Nohl, C., R., Baird, H. S. : Language identification in complex, unoriented and degraded document images, Proceedings International Workshop on Document Analysis Systems, Malvern, Pennsylvania (1996) 76-98.
9. Chaudhuri, B., B., Pal, U. : An OCR system to read two Indian language scripts : Bangla and Devnagri (Hindi), Proceedings ICDAR97, Ulm, Germany (1997) 1011-1015.

Appendix: Shape features for Devnagri and English Numerals

Numeral	**Shape features**
Devnagri 1	Shape_ 1.1 (1 to 30), Shape_1.4 (1 to 30), Shape_1 .3 (30 to 75), Shape_1.4 (30 to 75), Shape_2.3 (30 to 45), Shape_2.4 (30 to 45), Shape_2.1 (45 to 75), Shape_2.2 (45 to 75), Shape_5.1 (35 to 75), Shape_5.3 (10, 35 to 75) Shape_5.2 (35, 45 to 75)
Devnagri 2	[Shape_3.4 (1 to 25) or Shape_3.2 (1 to 15)], Shape_3.2 (25 to 55), Shape_1.4(30 to 75), Shape_1.3 (30 to 75), Shape_2.1 (1 to 15), Shape_ 2.1(60 to 75), Shape_2.2 (60 to 75), Shape_ 5.1(60 to 75)
Devnagri 3	Shape_ 1.1(1 to 40), Shape_ 1.1 (40 to 65), Shape_ (25 to 55), Shape_1.4 (40 to 75), Shape_2.1 (1 to 10), Shape_2.1 (70 to 75), [Shape_ 3.4(1 to 25) or Shape_3.2 (1 to 25)]
Devnagri 4	Shape_1.3(25 to 55), Shape_1.2 (25 to 55), Shape_1.1(40 to 75), Shape_1.4 (40 to 75), Shape_ 2.1(40 to 65), Shape_2.4(40 to 65), [Shape_ 3.4(1 to 15) or Shape_3.3 (1 to 15)], Shape_ 5.3(15, 40), Shape_5.3 (65, 40), Shape_ 3.5(15 to 75), Shape_3.6 (15 to 75), Shape_ 5.4(1 to 10)
Devnagri 5	Shape_ 3.1(1 to 25), Shape_ 1.1 (1 to 40), Shape_ 2.2(1 to 40), Shape_2.2 (40 to 75), Shape_ 2.1(40 to 75), Shape_5.3 (25, 5) , Shape_1.2 (1 to 55), Shape_2.2 (40 to 75), Shape_ 5.3(35, 55 to 75).
Devnagri 6	Shape_ 1.4(1 to 40), Shape_ 1.4 (40 to 75), Shape_ 1.2(25 to 55), Shape_1.1 (40 to 75), Shape_2.4(1 to 10), Shape_3.1 (45 to 75), [Shape_ 3.1(1 to 25) or Shape_5.1 (1 to 15)], Shape_5.1 (70 to 75)
Devnagri 7	Shape_5.2(15,25), Shape_5.2 (20, 25), Shape_5.2(35,40), Shape_5.2 (60, 65) Shape_5.2(65,75), Shape_2.2 (40 to 75), Shape_2.3(40 to75), Shape_3.5 (15 to 75), [Shape_3.3(1 to 25)or Shape_3.4 (1 to 25)], Shape_3.6(15 to 75), Shape_5.3 (25, 45 to 75), Shape_5.4(1 to 15)
Devnagri 8	Shape_5.3(65,5to 35), Shape_2.3 (1 to 35), Shape_5.1(1 to 35), Shape_2.3 (1 to 35), Shape_2.4(1 to 65), Shape_3.5(1 to 30), Shape_3.3 (50 to 75), Shape_3.6(1 to 65)
Devnagri 9	Shape_5.3 (10, 35 to 75). Shape_5.1 (35 to 75), Shape_2.2 (30 to 75), Shape_2.1 (30 to 75), Shape_ 1.1 (1 to 25), Shape_ 1.4 (1 to 25), Shape_3.6 (1 to 75), Shape_3.5 (45 to 75)

English 0	Shape_2.1 (1 to 10), Shape_2.4 (1 to 10), Shape_2.3 (60 to 75), Shape_2.2 (60 to 75), Shape_3.5 (10 to 75), Shape_3.6 (10 to 75), Shape_5.2 (35, 40), Shape_5.2 (40, 45), Shape_5.3 (40, 15), Shape_5.3 (40, 65)
English 1	Shape_5.1 (15 to 65), Shape_3.5(1to65), Shape_3.6(15 to 65), Shape_4.1 (5 to 65), Shape_4.2 (15 to 65), Shape_5.2 (15,25 to65), Shape_5.2 (35, 40), Shape_5.2 (40, 45), Shape 6.1
English 2	Shape_2.4 (30 to 65), Shape_2.3 (15 to 65), Shape_5.1 (30 to 55), Shape_ 1.1 (1 to 40), [Shape_3.4 (1 to 25) or Shape_3.2 (1 to 25)] Shape_2.1 (1 to 10), Shape_2.3 (10 to 50), Shape_2.1 (50 to 75), Shape_2.2 (1 to 40)
English 3	Shape_1.1 (1 to 40), Shape_ 1.1 (40 to 75), Shape_1.3 (25 to 55), Shape_2.1 (1 to 15),[Shape_3.4 (1 to 25) or Shape_3.2 (1 to 25)], Shape_2.2 (1 to 40), Shape_2.4 (40, 75), Shape_2.3 (65 to 75)
English 4	Shape_3.1 (1 to 15), Shape_4.2 (55 to 75), Shape_4.1 (40 to 75), Shape_5.1 (50 to 75),Shape_3.2 (25 to 55), Shape_3.5 (50 to 75), Shape_5.3 (25, 50 to 75), Shape_5.2 (55, 60 to 75), Shape_3.6 (50 to 75)
English 5	Shape_2.3 (1 to 25), Shape_2.1 (25 to 60), Shape_2.3 (60 to 75), Shape_2.2 (5 to 40),[Shape_3.3 (1 to 15) or Shape_3.1 (1 to 15)], Shape_2.4 (40 to 75), Shape_ 1.1 (30 to 75),[Shape_3.2 (50 to 75) or Shape_5.1 (50 to 75)], Shape_ 1.2 (10 to 75), Shape_ 1.3 (1 to 50)
English 6	Shape_5.3 (65, 5 to 35), Shape_2.3 (1 to 35), Shape_2.3 (1 to 35), Shape_2.4 (1 to 65),Shape_2.2 (65 to 75),Shape_3.5 (1 to 30), Shape_3.6 (1 to 65), Shape_ 1.1(45 to 75)
English 7	[Shape_3.4 (1 to 25) or Shape_3.2 (1 to 25)], Shape_2.3 (5 to 75), Shape_2.4 (25 to 75),Shape_5.2 (25,30 to 75), Shape_2.2 (1 to 25), Shape_2.4(40 to 75) Shape_4.3 (25 to 75),Shape_4.4(25 to 75), Shape_5.1 (25 to 75)
English 8	Shape_ 1.1 (1 to 40), Shape_ 1.1 (40 to 75), Shape_ 1 .4 (1 to 40), Shape_3.5 (40 to 75),Shape_3.6 (5 to 75), Shape_ 1.3 (25 to 55), Shape_ 1.2(25 to 55) Shape_5.3 (15,40),Shape_5.3(65,40)
English 9	Shape_1.4 (1 to 25), Shape_5.3 (15,45 to 75), Shape_5.2 (45,50 to 75), Shape_5.1 (45 to 75), Shape_3.2 (15 to 45) ,Shape_3.5 (35 to 75)

Note:

a) The notation Shape_(yl to y2) implies that the shape primitive is expected to occur in the range yl to y2.

b) The notation Shape_5.2(yl,y2 to y3) implies that the difference of the widths between the point yl and all of the points in the range y2 to y3 is a small value.

c) The notation Shape_5.3(yl,y2 to y3) implies that width at the point yl is greater than all the points lying in the range y2 to y3.