

# Machine-printed and hand-written text lines identification

U. Pal, B.B. Chaudhuri \*

*Computer Vision and Pattern Recognition Unit, Indian Statistical Institute, 203 B.T. Road, Calcutta 700 035, India*

Received 27 March 2000; received in revised form 23 August 2000

---

## Abstract

There are many types of documents where machine-printed and hand-written texts intermixedly appear. Since the optical character recognition (OCR) methodologies for machine-printed and hand-written texts are different, to achieve optimal performance it is necessary to separate these two types of texts before feeding them to their respective OCR systems. In this paper, we present a machine-printed and hand-written text classification scheme for Bangla and Devnagari, the two most popular Indian scripts. The scheme is based on the structural and statistical features of the machine-printed and hand-written text lines. The classification scheme has an accuracy of 98.6%.

*Keywords:* Optical character recognition; Document processing; Indian language; Machine-printed and hand-written text; Bangla and Devnagari text

---

## 1. Introduction

Optical character recognition (OCR) concerns automatic recognition of text characters in a document page. Some of the potential applications of OCR include office automation, reading aid for the blind, natural language processing, multimedia design, etc. Attempt to recognize machine-printed text in a fair quality document is a success story and several commercial systems are available in the market that perform efficiently and accurately. Systems for good hand-written text recognition are also available in the market. Unfortunately, these systems can perform on Latin-based script only

(Govindan and Shivaprasad, 1990; Impedovo et al., 1992; Mori et al., 1984, 1992). Some systems on Arabic, Chinese, Japanese and Korean scripts have also been reported (Amin, 1998; Nagy, 1988). However, Indian scripts are largely neglected, only a few papers have dealt with OCR of Bangla and Devnagari (Chaudhuri and Pal, 1998; Pal and Chaudhuri, 1997; Sinha and Mahabala, 1979).

From the comprehensive surveys of Govindan and Shivaprasad (1990), Impedovo et al. (1992), Mori et al. (1992) and Nagy (1988) it can be understood that machine-printed and hand-written character recognition schemes are quite different from each other in almost all steps like preprocessing, character segmentation, size normalization, feature extraction, matching and classification as well as post-processing like error detection and correction. So, if a document contains both machine-printed and hand-written text portions, they should be separated and fed to their

respective OCR systems to achieve optimal performance.

Intermixed appearance of hand-written and machine-printed texts in a single document is common in several kinds of documents, especially in table form documents. A form document is a combination of two parts. One is the preprinted machine-printed text, and other is hand-written fill-in text. Other examples of mixed document are question papers where answers are to be written by hand on blank space in box or over dotted lines, fax cover page, where recipient's name and address are generally written by hand, etc. (Li and Srihari, 1995).

There exist a few papers on the separation of machine-printed and hand-written texts but they deal with English, Chinese and Japanese scripts. In 1993, Imade et al. (1993) described a method to segment a Japanese document into machine-printed Kanji and Kana, hand-written Kanji and Kana, photograph and printed image. By extracting the gradient and luminance histogram of the document image, they used a layered feed-forward neural network model in their system. Franke and Oberlander (1993) reported a method to check whether a data field in a form document is hand-written or printed. In 1995, using directional and symmetrical features as the input of a neural network, Kuhnke et al. (1995) developed a method to identify machine-printed and hand-written English characters. Recently, Fan et al. (1998) described a method for the classification of machine-printed and hand-written text lines from English, Japanese and Chinese scripts. They used spatial features and character block layout variance as the prime features in their approach. None of the above pieces of work deals with Indian script documents.

This paper deals with separation of machine-printed and hand-written texts in *Devnagari* and *Bangla*, two popular scripts in south Asia. *Devnagari* and *Bangla* are the first and second most popular scripts in the Indian sub-continent. The structure of these two scripts is different from those of English, Chinese and Japanese. In the separation scheme, we used a robust and fast technique based on structural and statistical features of machine-printed and hand-written text lines in these scripts. To the best of our knowledge,

this is a pioneering work of its kind on Indian language scripts.

The organization of the paper is as follows. In Section 2 properties of *Bangla* and *Devnagari* scripts are presented. Preprocessing like text digitization, noise cleaning, different text column segmentation, text mode (portrait or landscape) detection as well as line segmentation from the document are described in Section 3. Section 4 deals with text classification scheme. Finally, experimental results and discussions are provided in Section 5.

## 2. Properties of *Bangla* and *Devnagari* scripts

Hindi and *Bangla*, the most popular languages in the Indian sub-continent, are together used by a total of about 500 million people. Also, Hindi and *Bangla* are, respectively, fourth and fifth most popular languages in the world. The script form of Hindi is called *Devnagari*, while that of *Bangla* is called *Bangla*. *Devnagari* script is used to write Hindi, Nepali, Marathi and Sindhi languages while *Bangla* script is used to write *Bangla*, Assamese and Manipuri languages. *Bangla* and *Devnagari* scripts are derived from the ancient Brahmi script through various transformations. Because of their same origin, these two scripts have some structural features in common. These common features help us to build up the system.

The properties of *Bangla* and *Devnagari* scripts that are useful for the present work are given below.

(a) There are 11 vowels and 39 consonant characters in *Bangla* while 11 vowels and 38 consonants in *Devnagari* alphabets. They are called *basic characters*. The set of basic characters in these scripts are shown in Fig. 1. Sometimes two or more characters combine and generate a complex shape in both *Bangla* and *Devnagari*. The resultant shape may be called as *compound character*. The concept of upper/lower case character is absent in these scripts.

(b) From Fig. 1 it is noted that many characters of *Bangla* and *Devnagari* alphabets have a horizontal line at the upper part. In *Bangla*, this line is called *matra*, and in *Devnagari* it is called

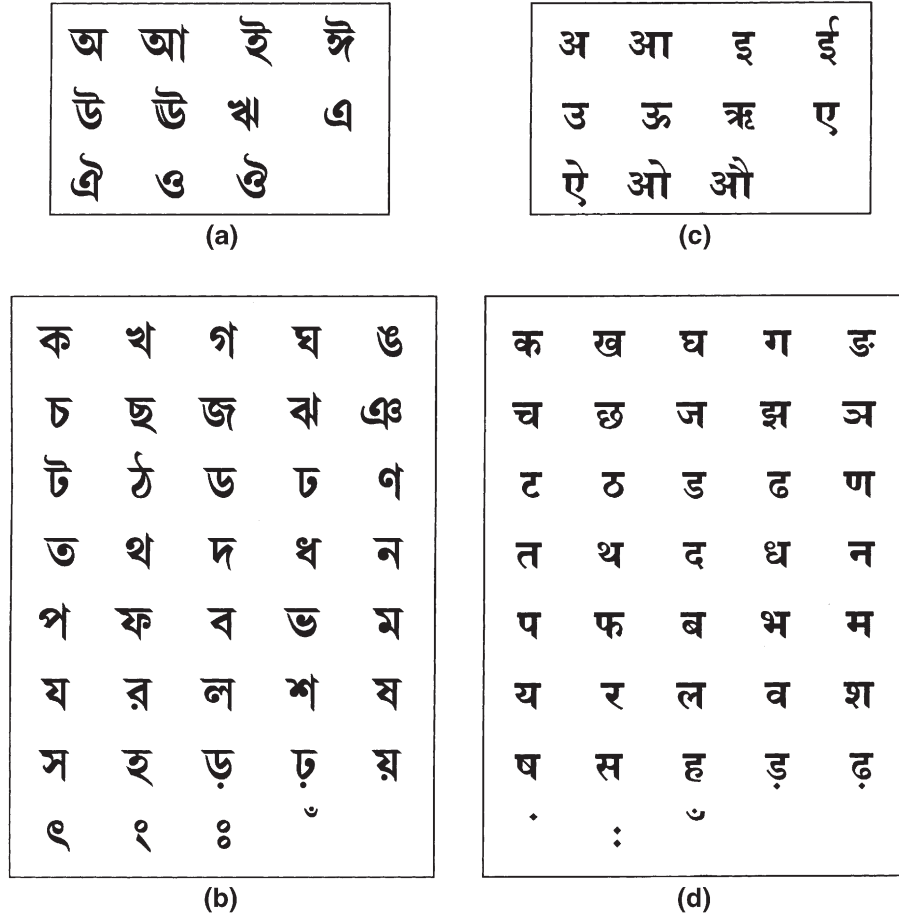


Fig. 1. Basic characters of Bangla and Devnagari script: (a) vowels of Bangla script; (b) consonants of Bangla script; (c) vowels of Devnagari script; and (d) consonants of Devnagari script.

*Sirorekha*. However, we shall give a common name, called *head-line*. When two or more Bangla (Devnagari) characters sit side by side in proper alignment to form a word, the *head-line* portions touch one another and generate a big head-line, which is used as a feature in our separation scheme.

(c) In Bangla (Devnagari) a vowel other than অ (अ) following a consonant takes a modified shape, which depending on the vowel is placed to the left, right or both (in case of Bangla), top or bottom of the consonant. See Fig. 2, where modified vowel shapes and their attachment with a consonant character ক (क) are shown. They are called *modified characters*. Modified character shapes are used in the classification scheme.

(d) A Bangla or Devnagari text line may be partitioned into three zones. The *upper zone* denotes the portion above the head-line, the *middle zone* covers the portion of basic (and compound) characters below head-line and above base-line, and the *lower zone* is the portion below base-line where some of the modifiers can reside. A typical zoning is shown in Fig. 3.

### 3. Preprocessing

For the experiment, text digitization has been done by a flatbed scanner (manufactured by HP, Model number ScanJet 4C). We have used a

Bangla Vowel	আ	ই	ঈ	উ	ঊ	ঋ	এ	ঐ	ও	ঔ
Modified Shape	।	ি	ী	ু	ূ	ৃ	ে	ৈ	ো	ৌ
When attached to consonant ক	কা	কি	কী	কু	কূ	কৃ	কে	কৈ	কো	কৌ
Devnagari vowel	आ	इ	ई	उ	ऊ	ऋ	ए	ऐ	ओ	औ
Modified shape	।	ि	ी	ु	ू	ृ	े	ै	ो	ौ
When attached to consonant क	का	कि	की	कु	कू	कृ	के	कै	का	कौ

Fig. 2. Shapes of Bangla (Devnagari) vowel modifiers when attached to the basic character ক(क).

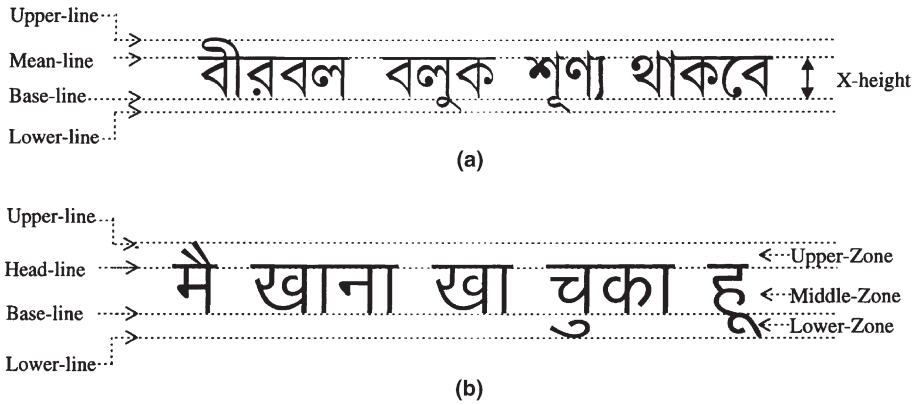


Fig. 3. Different zones of (a) Bangla and (b) Devnagari script lines.

histogram-based thresholding approach to convert the digitized gray tone images into two-tone ones. For a clear document, the histogram shows two prominent peaks corresponding to white and black regions. The threshold value is chosen as the midpoint of two histogram peaks. The two-tone image is converted into 0–1 labels where the label 1 represents the object (black) and 0 represents the background (white).

For accurate text classification, the system should properly detect individual text columns and should accurately segment the lines from each text column. Different columns of a text document are detected using the run length smoothing approach due to Wang et al. (1982).

After detection of each text block, the mode of the text (portrait or landscape mode) of the document is determined. A text block is in portrait (landscape) mode if the text lines in that block are horizontal (vertical). To determine the mode of a text block we use the property that the white space between characters is much smaller than the white space between the lines. We compute the horizontal and vertical projection profiles of a text column. The projection profiles are obtained by accumulating the number of black pixels that appeared in the same row or column, then summarizing the accumulated data to form a histogram. In a projection profile, a text line will appear as a black hill and a white stream between two text

lines will appear as a valley. From the projection profile the position of valleys and hills are found. A spatial feature, called hill–valley-distance (HVD), is extracted to estimate the orientation of the text block. To find the valley and hill points, the projection profiles are smoothed by a run-length based smoothing approach. To smooth the horizontal profile, we scan the horizontal profile column-wise and if the length of a white run is less than a threshold  $T$  (computation of this threshold value is discussed in the next paragraph), the white run is changed into black. Vertical profile is also smoothed in a similar way, only scanning mode is now row-wise. The smoothed version of the vertical and horizontal profiles of Fig. 4(a) is shown in Figs. 4(b) and (c). In each smoothed hill and valley region, the top-most hill point and bottom-most valley point are noted. Top-most hill points and bottom-most valley points are shown by  $h$  and  $v$  in the smoothed version of the projection profiles. Let  $h_1$  and  $h_2$  be the lengths of two consecutive hills and  $v_1$  be the length of the valley between these two hills (length of a hill means the length of the top-most point of the hill from the base, and length of the valley means the length of the bottom-most point of the valley from the base). We compute the HVD as follows:

$$\text{HVD} = (h_1 + h_2)/2 - v_1.$$

We compute average HVD values for both horizontal and vertical profiles. Let these values be  $W_h$  and  $W_v$ , respectively. We decide that the mode of the text block is portrait if  $W_h > W_v$ . Otherwise, the orientation is landscape.

The value of threshold  $T$  can be estimated as follows. For most printed text documents, the character size is not smaller than 6 points. Then, the minimum spacing between two lines is 12 points. If the document is digitized at  $P$  dpi, then the minimum distance between two text lines will be  $P \times 12/72$  pixels =  $P/6$  pixels (since 72 points = 1 inch). For a document digitized at 300 dpi we have  $T = 50$ .

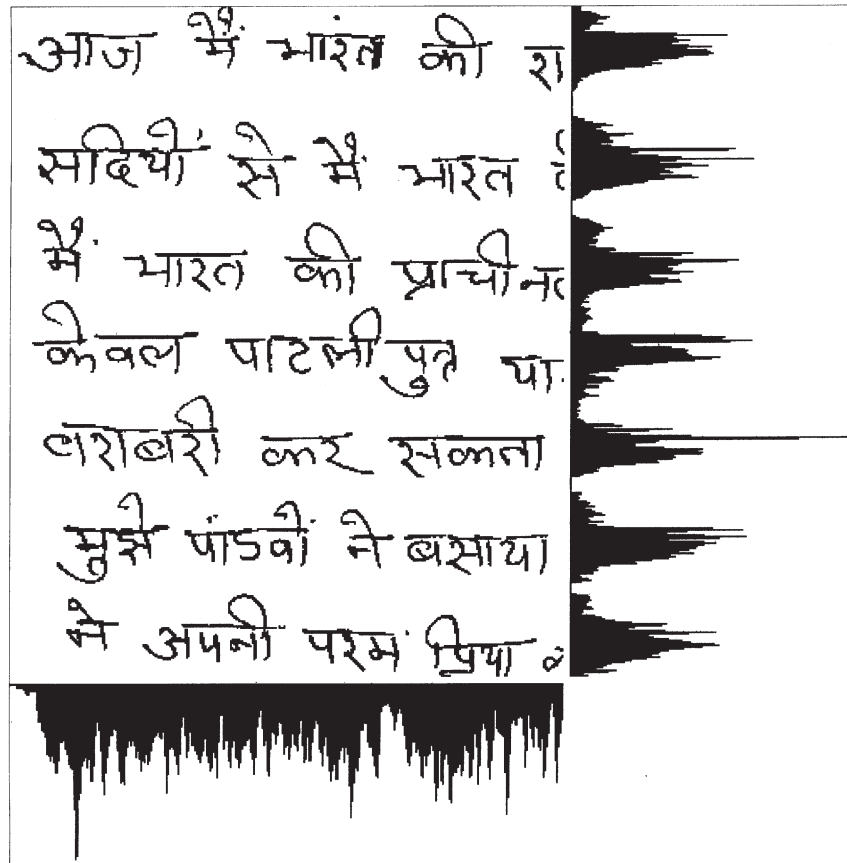
The lines of a text block are segmented by noting the valleys of the projection profile. The position where profile height is least denotes one boundary line. A text line can be found between two consecutive boundary lines.

#### 4. Classification of machine-printed and hand-written text lines

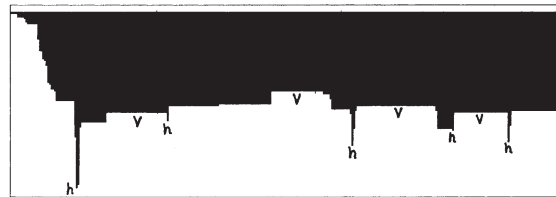
Our separation scheme is a three-tier tree classifier where in the nodes of the tree we use some simple and easily detected features of machine-printed and hand-written texts. Selected features for classification are based on (a) statistical analysis of machine-printed and hand-written texts, and (b) insensibility of font and style variations. For the statistical analysis, we have collected hand-written of 1500 individuals with different educational and professional status. We note that the handwritings obtained from 91% individuals do not have long head-line. We also note that text lines are not properly horizontal in the handwritings obtained from 57% individuals. The flow-diagram of the classification scheme is shown in Fig. 5. We shall discuss here the classification technique of portrait mode document. The classification technique for landscape documents can be done in a similar way. Different level features used in the scheme are as follows.

##### 4.1. First level feature

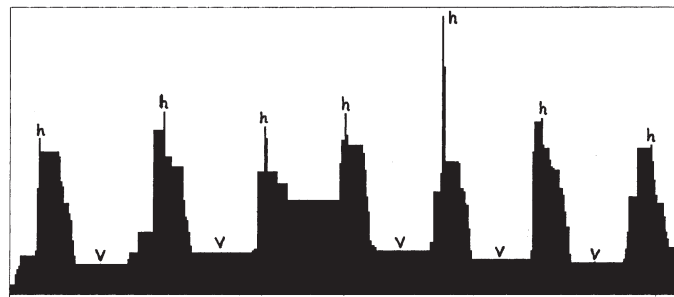
Since characters of a word sit side by side in proper alignment in a machine-printed text line, the head-line portions of the characters of a word touch one another and generate a big head-line. To test the likelihood of head-line in a machine-printed word, we note that out of the 50 basic characters in Bangla there are 32 characters with head-line while in Devnagari out of 49 basic characters 42 characters have head-line. We have computed character occurrence statistics in Bangla language (Chaudhuri and Pal, 1995). From these statistics, we note that out of 12 most frequent characters, only one character has no head-line. So, it is likely that most Bangla words will have a head-line. We can make a better quantitative analysis of the presence of head-line, as follows. According to the computed statistics (Chaudhuri and Pal, 1995), the average length of Bangla words is about six characters. We noted vowel modifiers are small in width and contribute very little to the head-line of the word. Also, we noted that compound characters are very infrequent, occurring in



(a)



(b)



(c)

Fig. 4. (a) Horizontal and vertical projection profiles are shown for writing-mode detection of a Devnagari text block. (b) Smoothed version of the vertical projection profile. (c) Smoothed version of the horizontal projection profile.

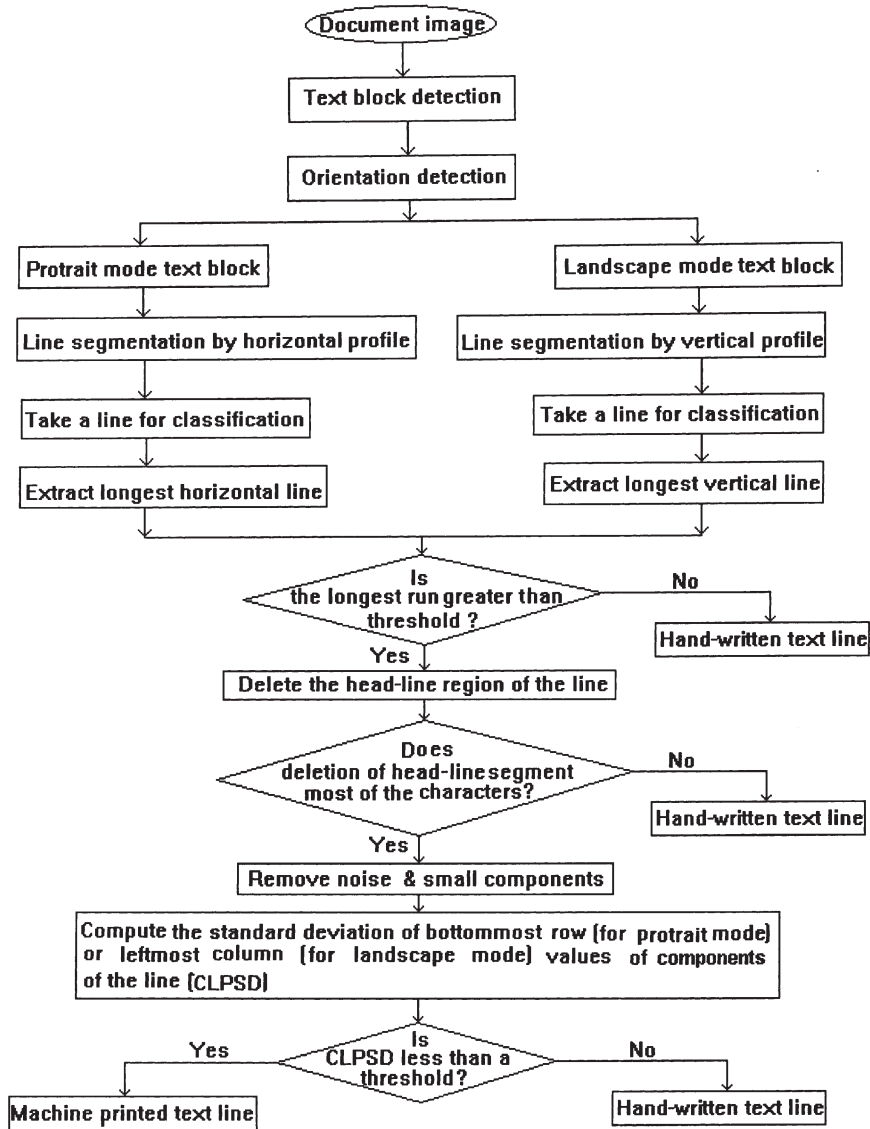


Fig. 5. Flow diagram of the classification scheme.

5% cases only. As a result, we assume that on an average four basic characters only contribute to the head-line of the word. We also assume that each character is equally likely in a word. In Bangla 41 characters can appear in the first position of a word. Out of these 41 characters, 30 characters have head-line. Hence the probability of getting a character with head-line in the first po-

sition of a word is  $P_1 = 30/41$ . Then the probability of getting a character without head-line in the first position is  $1 - P_1 = 11/41$ . As argued above, the characters which can contribute to the head-line in the other positions of a word are mostly consonants. Since 28 out of 39 Bangla consonants have head-line, the probability of getting a consonant with head-line for other positions

in a word is  $P_2 = 28/39$ . Then probability of getting a character without head-line in other positions is  $1 - P_2 = 11/39$ .

Thus, probability of all four characters without head-line in a word is  $(1 - P_1)(1 - P_2)^3 = 0.006$  (assuming that all characters are equally likely and independently occurring in a word). Hence, probability that a word will have at least one character with head-line is  $1 - 0.006 = 0.994$ . Analyzing in the same way we get for Devnagari, the corresponding probability of 0.997. The practical situation is better than these estimates since characters are not equally likely in a word and most frequently used characters have head-line.

Thus, it is quiet reasonable to use this head-line feature for the separation. At first, we use this head-line feature for classification. The hand-written text lines can be separated from machine-printed text lines by computing the longest row-wise horizontal run. We have noted that machine-printed text lines always generate a long horizontal run. As mentioned earlier, in about 91% of the handwritings such a long run is not present. But when somebody writes very carefully and slowly, long head-line may appear (9% cases only) in their handwritings. So, the hand-written text lines may or may not generate such a long run. If the length of the longest run is less than  $T_1$ , then we declare the line as hand-written one. Otherwise, no decision is taken and we use second level feature for separation. For illustration see Fig. 6. In this figure, the first and third lines are hand-written while the second line is machine-printed. For the second and third lines, the longest horizontal run

is greater than  $T_1$  although the third one is a hand-written line, while for the first line this run is less than  $T_1$ . The value of  $T_1$  is set experimentally as twice the height of middle zone of a text line.

#### 4.2. Second level feature

We noted that characters in Bangla or Devnagari machine-printed word are connected through the head-line. If we delete the head-line region from a text line then for machine-printed text lines all characters in that line get isolated. On the other hand, for most hand-written text lines all characters will not be isolated because of the irregular alignment of the characters in words and lines. Thus, if characters are not isolated by the deletion of head-line region, we declare the line as hand-written one. Else, it may be a machine-printed line or hand-written line (that written slowly and carefully). See Fig. 7 for illustration. Here, three text lines and their position after deletion of head-line region are shown. From Figs. 7(a) and (b) it can be noted that the characters are topologically segmented due to the deletion of head-line region although Fig. 7(a) is machine-printed and Fig. 7(b) is hand-written text line. But for the hand-written text line shown in Fig. 7(c), characters are not topologically segmented after deletion of head-line region. Thus, we can classify this line as hand-printed without using extra feature.

The head-line region deletion is done as follows. From the text line we find the row ( $L_r$ ) where longest horizontal run occurs, and we compute the upper envelope and lower envelope of  $L_r$ . The

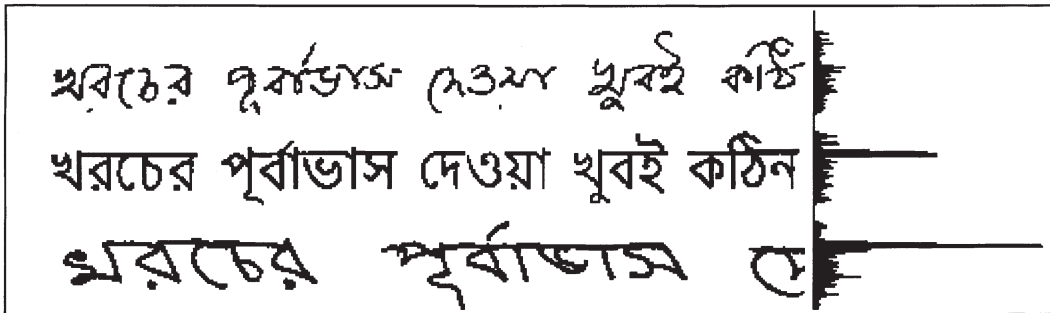
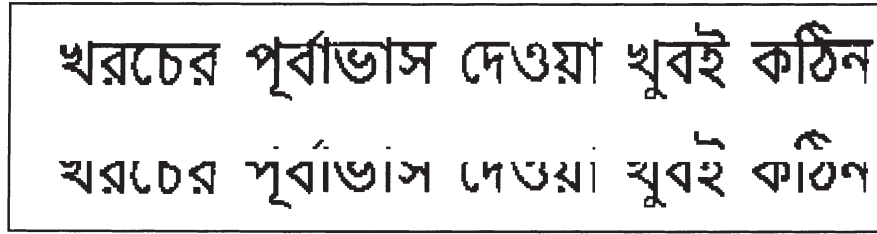


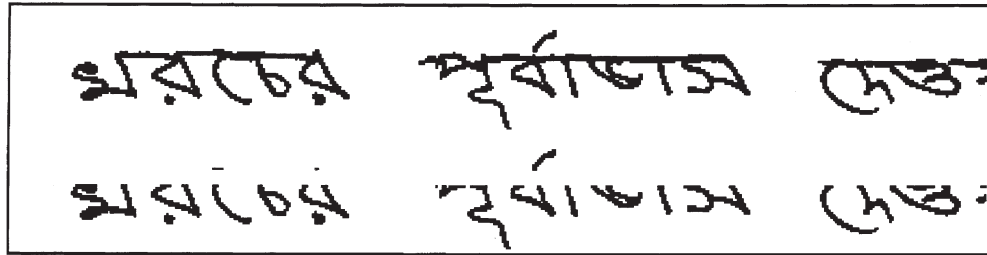
Fig. 6. Example of the longest horizontal run of three text lines is shown. (Here, second line is machine-printed while other lines are hand-written.)





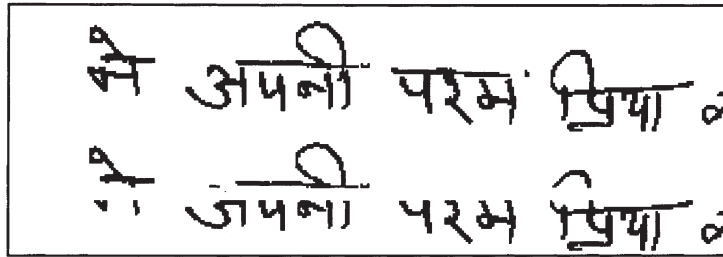
খরচের পূর্বাভাস দেওয়া খুবই কঠিন  
 খরচের পূর্বাভাস দেওয়া খুবই কঠিন

(a)



খরচের পূর্বাভাস দেওয়া  
 খরচের পূর্বাভাস দেওয়া

(b)



শ্রী অদলী পরম প্রিয়া  
 শ্রী অদলী পরম প্রিয়া

(c)

Fig. 7. Examples of three text lines and their positions after deletion of head-line region are shown (here, (a) is the example for machine-printed text line while other two for hand-written).

region between upper envelope and lower envelope is the head-line region and deletion of head-line is nothing but the deletion of the region between upper and lower envelopes. To get the upper envelope, from the row  $L_r$  column-wise upward vertical scan is made. For a column the upward scanning is stopped as soon as it hits a white (non-image) pixel, and its co-ordinate is noted. So, for an image having  $m$  columns, we get  $N_i$  ( $i = 1, 2, \dots, m$ ) points for upward scan. The row which contains maximum number of these  $N_i$  points, is called as upper envelope. The lower envelope is obtained in a similar way but the mode of scanning is downwards. The result of head-line deletion is shown in Fig. 7.

In actual implementation the head-line is not rubbed off. We do not consider the portion of the text line between the upper envelope and lower envelope.

#### 4.3. Third level feature

We use this feature, when all characters of a line are topologically segmented due to deletion of head-line regions at the second level discussed above. Here, to identify a line we note the distribution of lowermost points of isolated components in middle zone and lower zone after deletion of head-line. We noted that the distribution of characters in lowermost points is regular in

machine-printed texts, and random in hand-written texts. This property is used at the third level feature for the identification of machine-printed and hand-written text lines.

In machine-printed text, we note that the lowest points of most of the characters of a text line lie only on two horizontal lines. For the characters to which a lower modifier is attached, the lowest points lie on lower-line. Otherwise, the lowest points lie on the base-line. For examples of base-line and lower-line, see the printed text lines shown in Fig. 3. Here, the lowest points of the characters lie either on base-line or lower-line. This is not true in hand-written text line because of non-alignment.

For a text line we compute two sets of lowest points  $B$  and  $L$  corresponding to base-line and lower-line. If the lowest point of a component does not lie on any one of these two lines then we include this point in one of the two sets as follows. Let  $B_r$  and  $L_r$  be the row numbers corresponding to base-line and lower-line. Now, a component with lowest row  $C_r$  belongs to the set  $B$  if  $|B_r - C_r| \leq |L_r - C_r|$ . Otherwise, it belongs to  $L$ . Let  $b_1, b_2, \dots, b_m$  be  $m$  lowest row values of  $m$  components belonging to set  $B$  and let  $l_1, l_2, \dots, l_p$  be  $p$  lowest row values of  $p$  components that belong to set  $L$ . We noted that for machine-printed lines most of the elements of set  $B$  are equal i.e., they lie on the same row. This distribution is true for the set  $L$  also. But these are not true in hand-written text lines. Now, a spatial feature called *character lowest point standard deviation* (CLPSD) is defined as

$$\text{CLPSD} = \sqrt{\frac{1}{m} \sum_{i=1}^m (b_i - \bar{b})^2} + \sqrt{\frac{1}{p} \sum_{i=1}^p (l_i - \bar{l})^2},$$

where

$$\bar{b} = \frac{1}{m} \sum_{i=1}^m b_i \quad \text{and} \quad \bar{l} = \frac{1}{p} \sum_{i=1}^p l_i.$$

A line is classified as machine-printed if the value of CLPSD is smaller than a threshold value  $r_1$ . Otherwise, it is called hand-printed. The threshold  $r_1$  is computed as

$$r_1 = 0.1$$

× Average height of components considered.

Due to the dots of some characters like ঞ, ণ, ঙ, ঝ, etc., in Bangla and, ञ, ण, etc., in Devnagari, or due to some punctuation marks like comma, or due to salt and pepper noise sometimes we may get high CLPSD value in a machine-printed text line and hence it may be wrongly identified as hand-written line. To tackle this situation, we find lowest points only of those components whose bounding box widths are greater than half of the average bounding box width of all components in the line. Hence, small and irrelevant components like dots of the characters as well as noise and punctuation marks are mostly filtered out.

## 5. Results and discussion

To demonstrate the feasibility and validity of the proposed approach, a wide variety of document images were tested. We applied our separation scheme on 600 different document images. In some documents the printed text lines were of various sizes, fonts and styles. The images were scanned from question papers, money-order form, application form and several other documents containing printed and hand-written text. On an average 54% of the document script lines were hand-written. We observed that accuracy of the system is 98.6%.

From the experiment, we noted that most of the identification errors are obtained from very short lines containing one word only. To detect this we find the position of the word. If the position of this word is extreme left, then we assume that the short line is the continuation of the previous line and the line is identified as that of the previous one. In this way we can reduce the classification error rate.

Since the features used in the classification scheme are independent of size, font and style variations of the script, the proposed scheme does not depend on size, font and style of the characters in the text line.

From the computed statistics we note that most of the hand-written text lines do not generate long

head-line. By the first level feature, which is very easy to compute, most of the lines are identified without using the second and third level features. Hence, the proposed approach is very fast. We noted that the time required for the separation of different script lines from a document image of  $512 \times 512$  in a SUN 3/60 (with microprocessor MC68020 and SUN O.S. version 3.0) machine is about 4 s. The experiments were programmed using C language.

This approach can be used for the separation of machine-printed and hand-written lines of other Indian languages, e.g. Marathi, Assamese, Panjabi, etc., because of their script similarity with Devnagari and Bangla.

## References

- Amin, A., 1998. Off-line Arabic character recognition: the state of the art. *Pattern Recognition* 31, 517–530.
- Chaudhuri, B.B., Pal, U., 1995. Relational studies between phoneme and grapheme statistics in current Bangla. *J. Acoust. Soc. India* 23, 67–77.
- Chaudhuri, B.B., Pal, U., 1998. A complete printed Bangla OCR system. *Pattern Recognition* 31, 531–549.
- Fan, K.C., Wang, L.S., Tu, Y.T., 1998. Classification of machine-printed and hand-written texts using character block layout variance. *Pattern Recognition* 31, 1275–1284.
- Franke, J., Oberlander, M., 1993. Writing style detection by statistical combination of classifiers in form reader applications. In: *Proc. 2nd Internat. Conf. Document Analysis and Recognition*, pp. 581–584.
- Govindan, V.K., Shivaprasad, A.P., 1990. Character recognition – a review. *Pattern Recognition* 23, 671–683.
- Imade, S., Tatsuta, S., Wada, T., 1993. Segmentation and Classification for mixed text/image document using neural network. In: *Proc. 2nd Internat. Conf. Document Analysis and Recognition*, pp. 930–934.
- Impedovo, S., Ottaviano, L., Occhinegro, L., 1992. Optical character recognition – a survey. *Internat. J. Pattern Recognition and Artificial Intell.* 5, 1–24.
- Kuhnke, K., Simoncini, L., Kovacs, -V.Z.M., 1995. A system for machine-written and hand-written character distinction. In: *Proc. 3rd Internat. Conf. Document Analysis and Recognition*, pp. 811–814.
- Li, J., Srihari, S.N., 1995. Location of name and address on fax cover page. In: *Proc. 3rd Internat. Conf. Document Analysis and Recognition*, pp. 756–759.
- Mori, S., Suen, C.Y., Yamamoto, K., 1992. Historical review of OCR research and development. *Proc. IEEE* 80, 1029–1058.
- Mori, S., Yamamoto, K., Yasuda, M., 1984. Research on machine recognition of hand-printed characters. *IEEE Trans. Pattern Anal. Machine Intell.* 6, 386–405.
- Nagy, G., 1988. Chinese character recognition: a twenty-five year retrospective. In: *Proc. 9th Internat. Conf. Pattern Recognition*, pp. 163–167.
- Pal, U., Chaudhuri, B.B., 1997. Printed Devnagari script OCR system. *Vivek* 10, 12–24.
- Sinha, R.M.K., Mahabala, H., 1979. Machine recognition of Devnagari script. *IEEE Trans. System, Man and Cybernetics* 9, 435–441.
- Wang, K.Y., Casey, R.G., Wahl, F.M., 1982. Document analysis system. *IBM J. Research and Development* 26, 647–656.