A Machine Learning Approach to Detection of Core Region of Online Handwritten Bangla Word Samples

S. Baral, S. Bhattacharya, A. Chakraborty, U. Bhattacharya, S. K. Parui
Computer Vision and Pattern Recognition Unit, Indian Statistical Institute, Kolkata, India
baral_sudarshan@yahoo.com, soumik862003@gmail.com, chakraborty.abhi89@gmail.com, ujjwal@isical.ac.in, swapan.parui@gmail.com

***Abstract*—Core region detection of handwritten cursive words is an important step towards their automatic recognition. Several preprocessing operations such as height normalization, slant estimation etc. are often based on this core region. This is particularly useful for word recognition of major Indian scripts, which have large character sets. The main parts of majority of these characters belong to the core region that is bounded above by a headline and bounded below by an imaginary base line. Only a few such characters or their parts appear either above or below the core region. A few approaches are available in the literature for detection of such a core region of offline handwritten word samples of Latin script. Also, a similar region is often determined for recognition of images of printed Indian scripts. However, none of these approaches have studied detection of core region of an unconstrained online handwritten word. In this article, we propose a novel method for detection of the core region of online handwritten word samples of Bangla, a major Indian script. For this we first perform smoothing on the samples and then segment a stroke into substrokes. We compute certain novel positional features from each such substroke. Using these features, a multilayer perceptron (MLP) is trained by backpropagation (BP) algorithm. On the basis of the output of the MLP, we determine the position of both the headline and the baseline. We have tested this approach on a recently developed large database of online unconstrained handwriting Bangla word samples. The proposed approach would also work on similar samples of Devanagari, another major Indian script. Experimental results are encouraging.**

I. Introduction

For automatic recognition of handwritten word samples, certain preprocessing steps need to be accomplished before their actual recognition takes place. One such step is the determination of the core region of a word sample. The core region represents the horizontal strip having most of the information (a major portion of the trace) present in a handwritten word sample [1]. Most of the existing algorithms [1-5] for core region detection, involve a number of parameters, which are determined manually or empirically. The earliest available work on detection of core region of handwritten words can be found in [5], which analyzed the horizontal density histogram by identifying the lines with the maximum horizontal density of foreground pixels per line of the input word image. In this study, the horizontal density histogram was searched for features such as maxima and first derivative peaks. These features along with a number of heuristic rules were used to find the core region of word images. In [3] and [4], the authors considered the horizontal density distribution rather than the horizontal density histogram to avoid the influence of local strokes in a word image. In a recent study [6], the horizontal black runs of the word image is considered to compute certain reinforced horizontal black run profile histogram which in turn provides the estimated core region.

All of the existing studies on core region detection of handwritten words considered offline image samples. To the best of our knowledge, there does not exist any such study on online handwritten word samples. Also, no such study could be found on detection of core region of either online or offline handwritten cursive words of an Indian script although these scripts need special attention due to their individual characteristics, which differ significantly from Latin or any other script. A few existing studies on online and offline handwritten cursive words of Indian scripts can be found in [7-13].

In the present paper, we describe our recent study on detection of the core region of on-line handwritten word samples of Bangla, the second most popular Indian script used in India and Bangladesh. We, in fact, explore if it is possible to avoid manual or empirical parameter setting in order to detect the core region. For this we use a multilayer perceptron (MLP) classifier and the well-known backpropagation (BP) training algorithm on the basis of certain novel positional features extracted from the sub-strokes of an input word sample. The present study shows that selection of parameter values requiring manual intervention can, in fact, be learnt efficiently using the BP algorithm.

In order to detect the core region, we separately detect the headline and the baseline (shown in Fig. 1) of a word sample. The horizontal strip bounded above by the headline and bounded below by the baseline, becomes the core region. One purpose of core region detection here is to normalize the size of the sample. We scale the sample so that the height of the core region is 100. Scaling is done keeping the aspect ratio unchanged. The results of this study will be helpful for future studies towards recognition of handwritten cursive words of Indian scripts.

2167-6445/14 $31.00 © 2014 IEEE
DOI 10.1109/ICFHR.2014.83

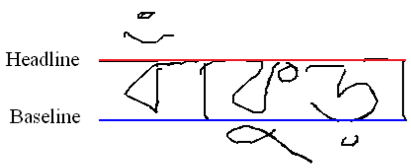


Figure 1. The location of headline and baseline of a Bangla word are shown by red and blue colored horizontal lines respectively.

Another advantage of having the core region is that it makes the recognition task a little simpler in the sense that in Bangla script there are only 5 symbols (characters or their parts) occurring above the headline and another 5 symbols (characters or their parts) occurring below the baseline. These symbols are shown in Fig. 2. Thus, recognition of symbols occurring outside the core region becomes easier. Also, an occurrence of such symbols has some association with the symbols occurring in the core region in the corresponding vertical strip. This enhances classification accuracy for symbols occurring in the core region. In the core region, there may appear 40 basic character symbols, 3 modifier symbols and 200 (approximately) compound character symbols.

The novelty of the present work lies in the proposed feature vector and in the use of machine learning to determine the optimal parameters for detection of the headline and the baseline, without going into manual or empirical setting of parameter values.

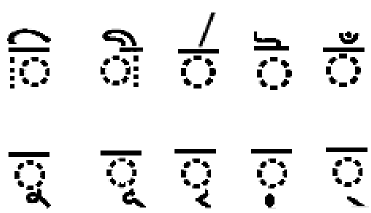


Figure 2. The first row shows shapes of 5 symbols which may occur above the core region of a Bangla word and the second row shows the shapes of another 5 symbols (including 'hasanta', a mute inherent modifier) occurring below the core region. The dotted portions show the regions where another character or its part always occur. The continuous horizontal line above each dotted portion shows the position of the headline.

II. Bangla Online Handwritten Word Database

Bangla script has 50 basic characters and their shapes are shown in Fig. 3(*a*). There are 11 basic vowels and these vowels appear in the beginning of the list in Fig. 3(*a*). All the vowels barring only the first one and two among the consonants have modified shapes known as modified characters. Shapes of modified characters are shown in Fig. 3(*b*). Additionally, there are approximately 200 compound characters of complex shapes formed due to merging of two or more basic characters. A study on Bangla corpus [14] reported that the percentage of occurrence of compound characters is only 7.34%. However, a majority of these compound characters occur rarely in a Bangla corpus. Shapes of a few compound characters are shown in Fig. 3(*c*). The present lexicon of 589 words consists of 50 basic characters, 12 character modifiers and 15 frequently occurring compound characters. These 77 characters constitute more than 96% of an existing Bangla corpus.

In the present study, we use a database of 9,424 online unconstrained handwritten Bangla word samples. These handwriting samples were provided by 16 native writers. Each of them wrote a pre-selected text consisting of 589 words aligned along 60 lines. These writers belong to different groups with respect to age, education, sex and income. The present lexicon consists of words of varying lengths of 2 to 7 characters. This database is annotated at sub-stroke level using a semi-automatic annotation tool and stored in an XML format. The annotation tool and the XML format has recently been reported in [7].

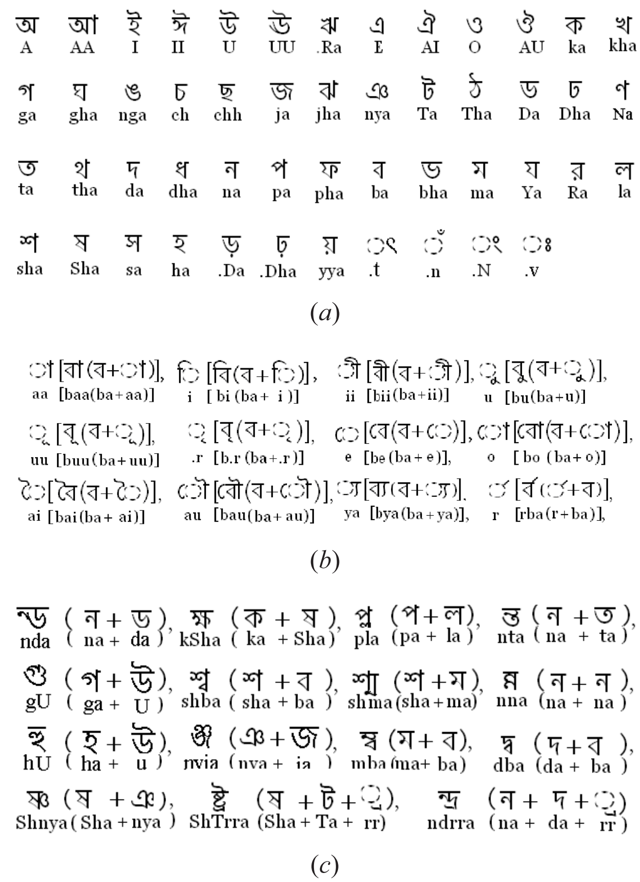


Figure 3. Alphabetic characters of Bangla – (*a*) basic characters, (*b*) character modifiers and (*c*) a few popularly occurring compound characters and their compositions being shown within parentheses.

The present database is divided into training and test sets. The training set consists of word samples written by 10 people whereas the test set consists of handwritten samples from the remaining 6 writers. Thus, there are 5890 training word samples and 3534 test samples.

III. PREPROCESSING

Here, the input word sample is a sequence of 2-dimensional points with original pen-up and pen-down information. The sequence of points between a pen-down and the next pen-up situations forms a stroke. The input word is first subjected to a set of preprocessing operations. Initially, the whole sequence of points in the word sample is translated upwards so that its topmost point touches the line y = 0. Next, the points are scaled so that the vertical distance between the top-most and the bottom-most points becomes 100 units (the value is chosen heuristically). Scaling is the same in both x and y directions so that the original aspect ratio is preserved. Then the repeated points are removed from the resulting sequence of points forming the scaled word. Next, the following smoothing operation is employed by keeping the first and the last points of all strokes intact. Each of the remaining points A_i is replaced by the average of the three consecutive points A_{i-1}, A_i and A_{i+1}. This smoothing is repeated thrice to get rid of the local distortions that may be present in the stroke preserving the essential shape of the stroke. The sequence of points in the stroke is now $\{A_i = (x_i, y_i), i = 1, 2, \cdots, M\}$. We next generate a sequence $\{P_i = (x_i, y_i), i = 1, 2, \cdots, N\}$ of equidistant points lying on the polyline defined by $\{A_i = (x_i, y_i), i = 1, 2, \cdots, M\}$. Here, $N < M$, $P_1 = A_1$ and $P_N = A_M$ and the distance between two consecutive points P_i and P_{i+1} (for all $i<N$) along the polyline is approximately 8 units (the value is selected empirically). Here, it may be noted that the height of the scaled word sample is 100 units. At the end of this preprocessing stage, the sequence of points is divided into several subsequences corresponding to the set of strokes.

IV. CORE REGION DETECTION

The proposed approach of core region detection of online handwritten word is based on segmentation of the word sample. The segmentation approach is based on the estimation of the position of the busy zone of the input word. A similar segmentation approach for unconstrained online handwritten Bangla words was reported earlier in [8]. This algorithm (Algorithm-1) is briefly described below.

A. Word Segmentation

Here, we first decide the horizontal lower boundary $y = HT\_Lim$ of the busy zone. Next, we compute a histogram of the y-values of the sample points lying above this lower boundary and obtain its modal value (M). We estimate the headline as $y = min\ (y \mid y \in \{y \mid freq\ (y) > B * M\})$, where B is empirically set to the value 0.5. We find all the local minima on the trajectory of the word which lie in a horizontal strip around this estimated headline. Various strokes of the input word are segmented into one or more sub-strokes at these local minima points. Details of this segmentation algorithm are provided below.

Algorithm-1 [Segmentation algorithm]

Step 1: Compute y_max, y_min and H = y_max - y_min based on the sample points of the input word. Translate the word to set y_min = 0. Here, y increases downwards.
Step 2: Scale the word setting its height (H) = 100.
Step 3: Remove repeated points and apply smoothing.
Step 4: Set $HT\_Lim = [A * H]$, where A $(0 < A < 1)$ is selected empirically.
Step 5: Compute frequency distribution of all those y-values for which $y < HT\_Lim$.
Step 6: Set M = modal value of the above frequency distribution.
Step 7: Obtain $S = \{y \mid freq\ (y) > B * M\}$, where B $(0 < B < 1)$ is selected empirically.
Step 8: Set $y\_Top = min\ (y \mid y \in S)$
Step 9: Obtain the set T of all local-minima of the word trajectory lying inside the strip between the horizontal lines $y = y\_Top - C*H$ and $y = y\_Top + C*H$, where C $(0 < C < 1)$ is selected empirically.
Step 10: Segment the word at each point of T.

Here, we set A = 0.75, B = 0.5 and C = 0.15 based on extensive simulation runs using training samples of the present database.

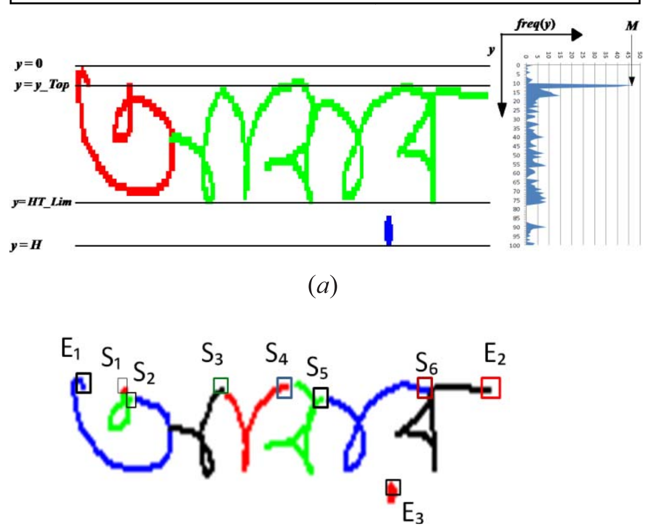


Figure 4. (a) An online handwritten cursive Bangla word sample scaled to the height 100. The histogram of the number of points in diferent horizontal lines is shown to the right of this word. (b) The segmentation points are indicated by S_1, S_2, S_3, S_4, S_5 and S_6 while the pen-up positions on the scaled word are shown by E_1, E_2 and E_3.

In Fig. 4(*a*), a sample of a cursive handwritten Bangla word consisting of 3 strokes and scaled to the height H (=100 units) is shown. The segmented word sample consisting of nine sub-strokes is shown in Fig. 4(*b*). The segmentation points are denoted by S_i, $i = 1, \ldots, 6$, while the end points of the original strokes are E_1, E_2 and E_3. In the proposed approach, the core region is computed based on these sub-strokes. Individual strokes or sub-strokes are indicated by varying colors in Figs. 4(*a*) and 4(*b*) respectively.

B. Proposed Approach Based on Machine Learning

The present approach is based on the determination of the y_{min} and y_{max} values of all the sub-strokes generated by Algorithm-1. The proposed algorithm for core region detection is described by Algorithm-2 provided below. The input here is the sets of points forming the sub-strokes and N = number of sub-strokes.

Algorithm-2 [Core region detection algorithm]

Step1: Compute y_{min} and y_{max} values for each sub-stroke.
Step2: Sort the y_{min} values in the ascending order and let y_min_i ($i = 1,2, \ldots, N$) denote the sorted values.
Step3: Similarly, sort the y_{max} values in the descending order and let y_max_i ($i = 1, 2, \ldots, N$) be the sorted list.
Step4: Compute
$gap\_min_i = (y\_min_{i+1} - y\_min_i), \forall\ i=1, \ldots, N/2$
i.e., gap_min_i values are computed for first $(N/2+1)$ sub-strokes in their sorted (ascending order) list w.r.t. the y_min values.
Step5: Form 10-dimensional feature vector $\boldsymbol{x}$ consisting of first ten gap_min_i values. If $N/2 > 10$, then we consider only the first 10 values, i.e., $gap\_min_1, \ldots, gap\_min_{10}$ to form the feature vector $\boldsymbol{x}$ and if $N/2 < 10$, some of the y_min_i values on the top are repeated to obtain ten gap_min_i values forming the feature vector $\boldsymbol{x}$.
Step 6: Feed the feature vector $\boldsymbol{x}$ to an already trained MLP network. If the response of the MLP is i, then the headline passes through y_min_{i+1}.
Step7: Compute
$gap\_max_i = (y\_max_i - y\_max_{i+1}), \forall\ i=1, \ldots, N/2$
i.e., above computation is done for $N/2$ sub-strokes sorted in the descending order w.r.t. the y_max values.
Step8: Form 10-dimensional feature vector $\boldsymbol{y}$ consisting of first ten gap_max_i values. If $N/2 >$ or < 10, then we follow the similar strategy of Step 5.
Step 9: Feed the feature vector $\boldsymbol{y}$ to another trained MLP network. If the response of this MLP is i, then the baseline passes through y_max_{i+1}.

In Fig. 5, a few word samples having varying number of sub-strokes are shown. For each sub-stroke the y_{min} and y_{max} points are indicated by red and green colored points respectively.

Now we propose an approach to re-estimation of the headline on the basis of the red points and the baseline on the basis of the green points. We consider a machine learning approach for the task. In particular, we use multilayer perceptron (MLP) architecture along with the backpropagation (BP) algorithm for its training. For this, the feature vector and the target value for a given training word sample are defined as follows. We first describe the strategy for re-estimation of the headline. It is trivial to check that a word sample having N sub-strokes provide N number of y_min_i values. Here, our assumption is that the headline passes through one such y_min_i for some index i. (Such an assumption is acceptable for the Indian scripts like Bangla, Devanagari in their printed form). This index i is the target value (target class) for our BP algorithm. Now since the headline occurs in the upper portion of a word sample, it must pass through y_min_i for where $i \leq N/2$. We train an MLP classifier to determine the index i on the basis of the feature vector defined as $\boldsymbol{x} = (gap\_min_1, gap\_min_2, \ldots, gap\_min_{N/2})$. Now since the value of N varies from sample to sample, we need to make the feature vector of uniform size. The largest value of N found in our database of handwritten word samples is 20. Therefore, we make the feature vector size 10.

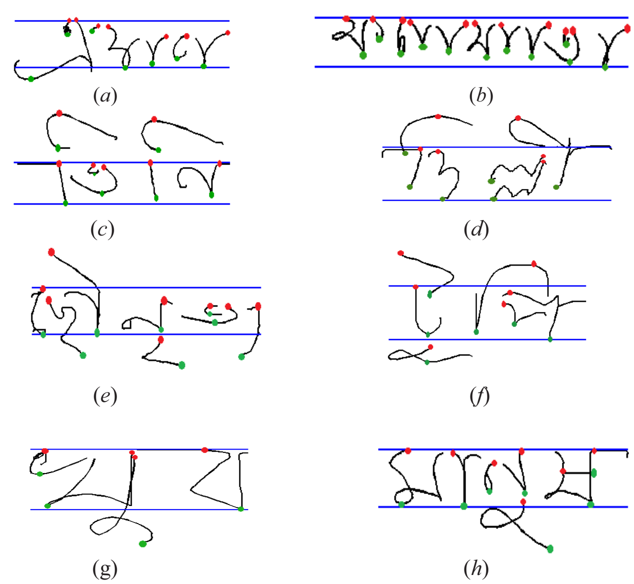


Figure 5. Core regions of some Bangla words after their segmentation. (*a*), (*b*) word samples having only middle zone; (*c*), (*d*) word samples having upper and middle zones; (*e*), (*f*) word samples having upper, lower and middle zones; (*g*), (*h*) word samples having middle and lower zones.

If $N/2 = M$ is greater than 10, then we ignore the excess number of y_min_i values and consider only the first eleven values y_min_1, y_min_2, y_min_3, $\ldots$, y_min_{11} providing 10 gap_min_i values in the sequence. On the other hand, if $N/2$ is less than 10, some y_min_i values on the top are repeated.

For example, if there are only 9 such values y_min_1, y_min_2, y_min_3, …, y_min_9, then we consider 11 values as y_min_1, y_min_1, y_min_2, y_min_2, y_min_3, …, y_min_9, where y_min_1 and y_min_2 are repeated. Similarly, if initially there are only 8 y_min values, the sequence of 11 y_min values are formed as y_min_1, y_min_1, y_min_2, y_min_2, y_min_3, y_min_3, y_min_4, …, y_min_8. These eleven y_min values are considered to obtain the feature vector $\boldsymbol{x}$ = (gap_min_1, gap_min_2, ..., gap_min_{10}). Thus, for each word sample in the training set, we have the feature vector $\boldsymbol{x}$ and index i indicating that the headline passes through y_min_i. The idea here is based on the observation that the gaps between consecutive y_min values of the sub-strokes occurring in the upper zone, are small and so are the gaps between consecutive y_min values of the sub-strokes occurring in the middle zone. But the gap between the largest y_min value of the sub-strokes occurring in the upper zone and the smallest y_min value of the sub-strokes occurring in the middle zone is relatively larger. This is true for the Indian scripts like Bangla and Devanagari. If gap_min_i is sufficiently large, the headline passes through y_min_{i+1}. If no gap_min_i is sufficiently large, the headline passes through y_min_1, that is, it passes through the top-most point in the word sample. The task of the MLP here is, based on the 10 gap_min values, to find the index i such that the headline passes through y_min_{i+1}. The parameters of the MLP are learnt on the basis of a training set of feature vectors x = (gap_min_1, gap_min_2, ..., gap_min_{10}) and the corresponding indices i.

The method described in the paragraph above is repeated to find the baseline of a word sample using y_{max} values instead of y_{min} values. Also, gap_max values are used instead of gap_min values. The feature vector is defined here as $\boldsymbol{y}$ = (gap_max_1, gap_max_2, ..., gap_max_{10}). The largest gap_max in the bottom area of the word sample determines the baseline. If no gap_max value is sufficiently large, the baseline passes through y_max_1, that is, it passes through the bottom-most point in the word sample.

The architecture of the MLP network is as follows. The input layer has 10 nodes while the output layer has 11 nodes. The size of the hidden layer is taken as 15. The same architecture is used for the detection of the baseline. Thus, we build two different trained MLP networks for the detection of the headline and the baseline.

V. Experimental Results

The whole database of word samples is annotated in the sense that the position y_min_i of the headline and the position y_max_j of the baseline are provided for each word sample. The training set of word samples used for the first MLP classifier for the detection of the headline, consists of 5890 samples and the test set consists of 3534 samples. The accuracy of placing the headline in the desired position obtained for the test set is 91.6%.

The same training and test sets of word samples were used for training of the second MLP. The accuracy of placing the baseline in the desired position obtained for the test set is 90.2%.

The proposed approach to core region detection is based on features extracted from the sub-strokes. We did not consider the strokes since individual strokes can be very long and the evidence obtained from a smaller number of strokes (the number of sub-strokes is usually larger) may sometimes be inadequate for the purpose of detection of headline or baseline. Fig. 6 shows such an example of a word sample whose strokes are shown in (*a*) and sub-strokes are shown in (*b*). Since in Fig. 6 (*a*) the number N of green points (y_max points) is only 3 and $\boldsymbol{y}$ = (gap_max_1, gap_max_2) has only 2 entries with gap_max_1 having a small value, the baseline is missed (note that gap_max_2 does not appear in the feature vector). On the other hand, in Fig. 6 (*b*) there are several green points and hence there are a larger number of gap_max_i values. Consequently, $N/2$ is larger so that gap_max_2 (which is larger than gap_max_1) appears in the feature vector.

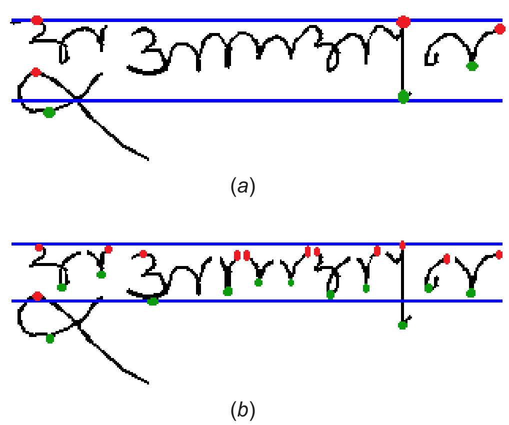
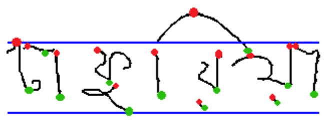


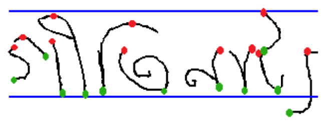
Figure 6. (a) A sample having only 3 strokes. Though headline is properly detected, baseline is not properly obtained. (b) Same sample after its segmentation into sub-strokes. Both headline and baseline are properly detected.

However, we have trained both the MLP networks described above on the basis of the same training set of samples using features extracted only from the strokes (instead of sub-strokes) and have tested on the same test set. The accuracy of the headline placement in the desired position obtained for the test set here is 90.4%. The same for the baseline is 88.1%.

Two word samples are shown in Fig. 7 for which the proposed core region detection approach based on sub-strokes, fail to detect the core region properly. For the sample in Fig. 7(*a*), the baseline has not been correctly detected while the headline is misplaced in Fig. 7(*b*).



(*a*)



(*b*)

Figure 7. Two samples where core region detection is not done properly. (a) Headline has been correctly detected, but not the baseline. (b) Headline has not been correctly detected. Conclusions

In the present study, we proposed a novel approach of detection of core regions of handwritten word samples. The proposed algorithm is based on a machine learning based approach. Although we simulated the approach for online handwritten cursive word for Bangla, the same may as well be studied for detection of similar region of online /offline word samples of different Indian scripts. In future, we shall study such possible extensions of the proposed approach to other Indian scripts.

REFERENCES

[1] A. Papandreou, B. Gatos, Word slant estimation using non-horizontal character parts and core-region information, *Proc. of the 10th Int. Workshop on Document Analysis Systems*, pp. 307-311, 2012.

[2] A. Rehman, D. Mohammad, G. Sulong, T. Saba, Simple and effective techniques for core-region detection and slant correction in offline script recognition, *Proc. of Int. Conf. on Signal and Image Proc. Appl.*, pp.15-20, 2009.

[3] A. Vinciarelli, J. Juergen, A new normalization technique for cursive handwritten words, *Pattern Recognition Letters*, vol. 22 (9), pp.1043-1050, 2001.

[4] M. Cote, E. Lecolinet, M. Cheriet, C. Suen, Automatic reading of cursive scripts using a reading model and perceptual concepts, *Int. Journ. Doc. Anal. Recog.*, vol.1(1), pp.3-17, 1998.

[5] R. M. Bozinovic, S. N. Srihari, Off-line cursive script word recognition, *IEEE Trans. Pattern Anal. & Machine Intell.* vol.11(1), pp.69-82, 1989.

[6] A. Papandreou, B. Gatos, "Slant estimation and core-region detection for handwritten Latin words", *Pattern Recognition Letters*, vol. 35, pp. 16-22, 2014.

[7] U. Bhattacharya, R. Banerjee, S. Baral, R. De and S. K. Parui, A semi-automatic annotation scheme for Bangla online mixed cursive handwriting samples, *Proc. of Int. Conf. on Frontiers in Handwriting Recognition*, pp.676-681, 2012.

[8] Sk. Mohiuddin, U. Bhattacharya and S. K. Parui, Unconstrained Bangla online handwriting recognition based on MLP and SVM, *Proc. of MOCR/AND '11*, Beijing, ACM New York, NY, USA, 2011.

[9] U. Bhattacharya, A. Nigam, Y. S. Rawat and S. K. Parui, An Analytic Scheme for Online Handwritten Bangla Cursive Word Recognition, *Proc. of the ICFHR*, pp.320 - 325, 2008.

[10] S. Basu, R. Sarkar, N. Das, M. Kundu, M. Nasipuri, D. K. Basu, "A Fuzzy Technique for Segmentation of Handwritten Bangla Word Images", *Proc. of the Int. Conf. on Comput.: Theory and Appl.,* pp 427-433, 2007.

[11] A. Roy, T. K. Bhowmik, S. K. Parui and U. Roy, "A Novel Approach to Skew Detection and Character Segmentation for Handwritten Bangla Words", *Proc. of the Digital Imaging Computing: Tech. and Appl.*, pp.203–210, 2005.

[12] U. Pal, S. Datta, "Segmentation of Bangla Unconstrained Handwritten text," *Proc. of the 7th ICDAR*, pp.1128-1132, 2003.

[13] A. Bishnu, B. B. Chaudhuri, "Segmentation of Bangla Handwritten Text into Characters by Recursive Contour Following," *Proc. of the 5th ICDAR*, pp.402-405, 1999.

[14] B. B. Chaudhuri and S. Ghosh, "A statistical study of Bangla corpus", *Proceedings of ICCLSDP '98*, Calcutta, India, pp.C32 - C37, 1998.