J. Intell. Syst. **20** (2011), 227–260
DOI 10.1515/JISYS.2011.013 

# Word Extraction and Character Segmentation from Text Lines of Unconstrained Handwritten Bangla Document Images

Ram Sarkar, Samir Malakar, Nibaran Das, Subhadip Basu, Mahantapas Kundu and Mita Nasipuri

**Abstract.** In this paper, a novel approach for word extraction and character segmentation from the handwritten Bangla document images is reported. At first, a modified Run Length Smoothing Algorithm (RLSA), called Spiral Run Length Smearing Algorithm (SRLSA), is applied for the extraction of words from the text lines of unconstrained handwritten Bangla document images. This technique has helped to overcome some of the drawbacks of standard horizontal and vertical RLSA techniques. SRLSA technique has been applied on the Bangla handwritten document image database CMATERdb1.1.1 and the success rate of the word extraction is found to be 86.01%. In the second part of the work, we have presented a useful solution to the problem on how best word images of handwritten Bangla script can be segmented into constituent characters. Moreover, the technique can segment the words having discontinuity in Matra, a prominent feature of Bangla script. It also optimizes the trade-off between under/over segmentation as Matra region and segmentation points are estimated more precisely. As a result, better word segmentation accuracy is achieved with minimal data loss. Here, a success rate of 92.48% is observed on a dataset of 750 handwritten Bangla words which is 3.35% higher than that of our earlier techniques.

**Keywords.** Word extraction, SRLSA, character segmentation, handwritten Bangla document image, CMATERdb.

**2010 Mathematics Subject Classification.** 68T10, 68U10.

## 1 Introduction

With the advent of digital computers and electronic storage mechanisms, conversion of paper documents into easily searchable electronic copies, which can be stored on any electronic storage device, has now become a reality. This conversion is of great importance as this helps in reducing the probability of their physical destruction as well as this facilitates in handling of those documents which in its paper form generally lie in the shelves gathering dust. Optical Character Recognition (OCR) is a special technique which helps in such conversion of text document images. It involves various stages ranging from scanning of the document to get

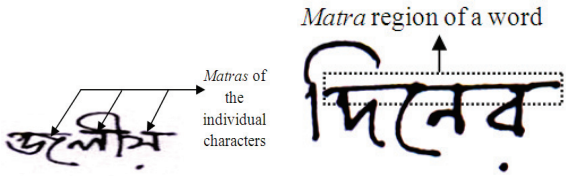
its digital image, text line extraction, word extraction, character segmentation to the character recognition. Each of these stages may be implemented by using various kinds of methods which have their own merits and demerits. In recent times, the methodologies for the OCR system have improved drastically but still there is not much advancement in the digitization of unconstrained handwritten document images.

It is already stated that word extraction is an essential part in document recognition system. Words in a text line must be identified correctly for their subsequent recognition by the computer. For printed documents, words are uniformly spaced and so the word identification becomes easy, but this is not true for handwritten documents. Thus for handwritten document images, the main challenge in the word extraction stage is to analyze the non-uniformity in inter-word and intra-word spacings in each text line. Wrong extraction of words during this stage will result in failure of most of the later stages in the handwritten document recognition system.

After word extraction, the next important step in OCR system is character segmentation. Character segmentation seeks to decompose word images into sub-images, each of which contains a constituent character. It is needed prior to machine recognition of individual character images that constitute the words. Higher segmentation accuracy ensures lower error rate during recognition. Knowledge of local features of consecutive characters is not always sufficient to identify proper segmentation points on character boundaries in a word image. In some cases, identification of proper segmentation points needs to be contextually consistent. That is, the word image recognized on the basis of various segmentation possibilities must select a match from the lexicon with the highest probability.

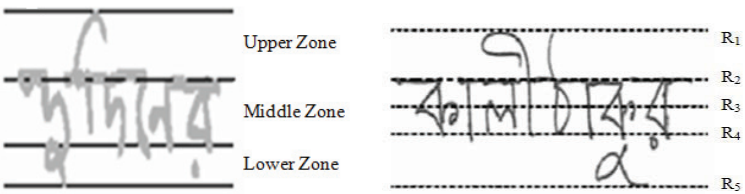
Word extraction and character segmentation techniques are mostly script dependent. It is not only because of variations of character shapes from one alphabet to other but there exists certain script specific features of text document. In this context, the present work considers the word extraction and character segmentation problems for *handwritten* text of Bangla script. Popularity wise, Bangla is the $5^{th}$ ranked in the world and $2^{nd}$ ranked in India as a script and language both. Bangla is the official language of Bangladesh. In India, Bangla is mostly used in West Bengal, Tripura and Assam. Moreover, Bangla script is also used for other two Indian languages, viz., Assamese and Manipuri.

In Bangla script, there is a prominent feature called the Matra. Typically, a Matra of Bangla character is a line of length at most the width of the character that passes horizontally touching the top of the character at some specified points. Joining the individual Matras of its constituent characters through a common line forms the Matra region of a word in Bangla script. Illustration of Matra in a Bangla word is shown in Figure 1(a) and (b).




(a) (b)

Figure 1. Illustration of Matra feature in a Bangla word.




(a) Different zones in a Bangla word.

(b) Different zone boundaries in a Bangla word.

Figure 2. Illustration of different zones and zone boundaries in a Bangla word image.

Before extracting the words or segmenting individual characters of each Bangla word in the text image, the word is horizontally partitioned into three adjacent zones as shown in Figure 2(a). The portion of each word on and above the Matra constitutes 'upper zone', the main body of the characters in a word lies within the 'middle zone', and the portion of the word, containing especially modified shapes and period like isolated character components, lies below the main body form the 'lower zone'. The technique of word segmentation, presented here, is based on detection of the Matra region.

## 1.1 Literature Review

### Word Extraction

Researchers have already tried to solve the problem of word extraction from handwritten documents but still the challenge exists. Most of them have used the conventional inter/intra word gap based techniques [1, 2], while some of them have also applied Artificial Neural Network (ANN) [3, 4] to solve this problem.

The technique which is lexicon-directed [5] have given good results for both printed and cursive handwriting. They posses tightly coupled segmentation and recognition components. They generate the best ways with which character images can be segmented; primitives of word can be assembled and matched to show a possible string in a lexicon. Here, the number of segmentation points is very high.

Contour detection method [6, 7] is also becoming popular in recent years due to their simplicity in implementation. In [6], initially the contour of the words present in a given text line are detected and then a threshold is chosen based on Median White-Run Length (MWR) and Average White Run-Length (AWR) present in the given text line. After that the word components are extracted from the text lines based on the contour and the previously chosen threshold value. At last, these words are represented in bounded boxes. Another method, reported in [8], analyzes the extent of Blobs in a scale space for the word extraction which gives a success rate of around 87 percent. Gaussian filters are used in this method. Recently, some researchers have used heuristics with ANN [9, 10] but these approaches are mainly meant for printed characters.

Run Length Smoothing Algorithm (RLSA) [11] is a very common technique used in the field of image processing. This method, mainly used for block segmentation and text discrimination, has developed for the document analysis system. Earlier, RLSA was used for the text line and word extraction purposes [12]. Here the smoothing algorithm is applied to the binarized image to find out the possible text line candidates. Later, the word extraction process is used to these text lines. The RLSA method has also been used for document page layout analysis [13] in printed documents. In this work, RLSA along with Neural Network Block Classified (NNBC) have been used to analyze the document page layout. NNBC consists of a principal component analyzer (PCA) and a self-organized feature map (SOFM). A modified version of this RLSA was first introduced in one of our earlier works [14] and was applied for extraction of word images from handwritten text lines. A detail description of the algorithm is also given in Section 4.3 of the current paper.

In another work [15], authors have discussed the image enhancement methods, text/graphics separation and identification and word language (English/Arabic) identification in printed document. Authors, here, have used RLSA technique horizontally to extract the text lines. Word extraction is also done on them using the same technique but with different set of threshold values. Khurshid et al. have proposed a system [16] for figure caption detection in printed document by employing a fusion of several information sources. In doing so, they have applied horizontal RLSA on the binarized image to extract the words.

### Character Segmentation

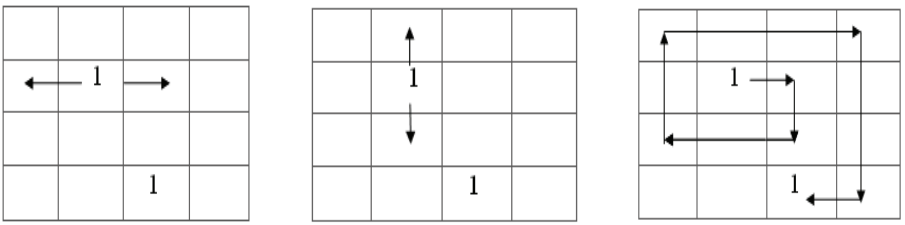
In the work [17–30], different character segmentation techniques have been introduced. All the segmentation works have been done on Matra-based scripts like Bangla [17–23], Devanagri [23–28] and Gurmukhi [29, 30]. In [17–23], several variations of algorithm for segmentation of Bangla word images have been proposed. Among them [17–22] have dealt with handwritten Bangla word images and [23] has dealt with printed Bangla word images as well as Devanagri words. In [17–20] different fuzzy function based algorithms, such as fuzzy triangular function [17] and bell-shaped function [18–20] have been reported for segmentation of handwritten Bangla word images. In one of our recent works [20], we have proposed further improvement of our existing techniques [17–19] by efficiently estimating potential segmentation points, resulting in minimal data loss.

In the work [21], recursive contour following technique has been applied for word segmentation from handwritten Bangla documents. Water reservoir principle [22] has been used in the past for the purpose of character extraction from handwritten Bangla word images. Reference involving touching character segmentation of printed Bangla/Devanagri words using fuzzy multifactorial analysis [23] is also available in the literature.

## 2 Motivation

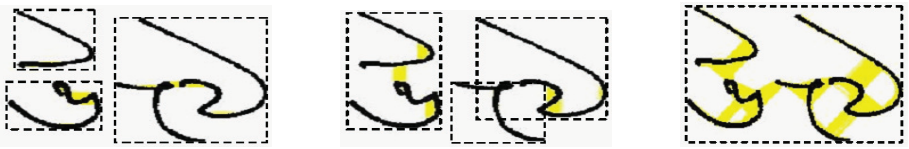
Word extraction is more difficult in unconstrained handwritten document images than the printed ones. In a printed document, all the characters have definite shapes and sizes, also intra-word and inter-word spacings are fixed as they are typed through the keyboard or typewriter. This makes the process of word extraction much easier. But, the problems related to unconstrained handwritten document images are much more complicated due to the wide varieties in handwriting patterns of the individuals.

In this paper, we have modified the standard RLSA technique and have proposed a new category called Spiral Run Length Smearing Algorithm (SRLSA). The SRLSA overcomes some of the shortcomings of the other two RLSA versions, namely Horizontal Run Length Smoothing Algorithm (HRLSA) and Vertical Run Length Smoothing Algorithm (VRLSA) and helps to obtain a better result in the word extraction stage. The HRLSA and VRLSA fail when the neighboring data pixel does not lie in a straight horizontal or vertical line. Though these pixels may lie very close spatially, but they are not smeared by these traditional methods. Figure 3 depicts such situations for both HRLSA (Figure 3(a)) and VRLSA (Figure 3(b)). In this type of case, SRLSA (Figure 3(c)) would be more useful.




(a) HRLSA (b) VRLSA (c) SRLSA

Figure 3. Comparison of different smearing strategies.



(a) HRLSA (b) VRLSA (c) SRLSA

Figure 4. Smearing of Bangla script word image using different versions of RLSAs.

Figure 4 shows the outcome of different versions of RLSAs on a word image written in Bangla script.

For any skew corrected printed document image, HRLSA works well for detection of Matra and this simplifies word extraction process from the text lines. However, if there is slightest skew, then this method fails. In handwritten Bangla script, the Matra is not at all prominent and often appears discontinuous as well as curvy. Hence the SRLSA technique works more efficiently for handwritten Bangla script in comparison with HRLSA or VRLSA.

From the above discussion related to character segmentation, it is clear that accurate estimation of Matra region of Bangla script plays a significant role towards segmentation of the word images. In most of our earlier works [17–19], detection of starting row of middle zone (i.e. $R_2$ as shown in Figure 2(b)) in the Bangla word fails in some cases as shown in Figure 5. The work proposed in [18] is basically an improvement of our earlier technique reported in [17]. In the work [18], a two-stage approach for Bangla character segmentation technique is reported. In this work, all the components after Connected Component Labeling (CCL) [31, 32] are sent to the Segmentation Decision step without doing any pre-processing. As a result, it is observed that Multi Layer Perceptron (MLP) classifier's success rate, in deciding whether to segment a connected component of a word, was not very sat-

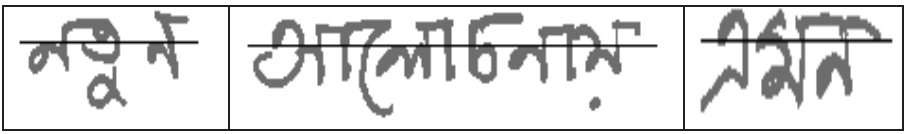
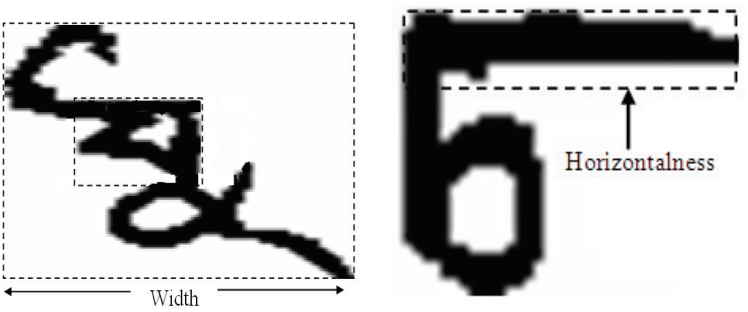


Figure 5. Wrong estimation of $R_2$ using the technique described in [18].




(a) (b)

Figure 6. Word components with high values of (a) width and (b) horizontalness feature.

isfactory. A possible reason for this is the vagueness of some feature descriptors, selected for the classification of the connected components. For example, horizontalness feature for identifying Matra region or width of the component sometime misleads the result. High values of these two features always do not mean that the component will be segmented vertically as shown in Figure 6(a) and (b). Further, application of the technique for determining segmentation points, [17–19, 24–26] also leads to under-segmentation, as shown in Figure 7(a), (b) and (c). In the present work, we have modified the technique of character segmentation, reported in [18], to overcome the problems discussed above.

## 3 Present Work

The current work may be viewed as an amalgamation of two of our recent works [14, 20]. Firstly, we have used the word extraction methodology, proposed in [14], that extracts the words from the text lines of handwritten Bangla document images. Secondly, we have applied the character segmentation technique, described in [20], on the extracted words to isolate the constituent characters.

SRLSA technique is applied to segment the Bangla handwritten text lines into constituent words. The input of the present work is the document images which

(a) (b) (c)

Figure 7. Different cases of under-segmentation, obtained by our earlier technique [18], are shown (darker regions represent potential segmentation points). Some of the constituent characters remain connected erroneously along the top of the segmentation points, highlighted by dotted oval shapes.

have already been segmented into text lines. We have binarized these text line images first, and then it is followed by finding the connected components present in the individual text line by applying CCL algorithm [31,32]. After that, we have removed the noises from the binarized image. The technique then applies SRLSA to smear the neighboring data pixels of the connected components in a segmented text line in order to get the word boundaries. In doing so, the neighboring components get merged and it eventually helps to form the word components in that particular text line. The bounding boxes of the components are found out and then two phases of refinements are done on the identified components. In the first phase we have merged the components, which actually belong to a single word, with respect to their spatial distance. In the second phase, lengthy horizontal components, which may be formed due to concatenation of more than one word, are broken into smaller components based on the spacing within the successive components. After refining the components, we have finally identified the word boundaries in each text line in the document image.

After extracting the words from the text line, we have developed a novel technique for identification of potential segmentation points on the Matra region to isolate constituent characters from the word image of Bangla script. In the first stage, CCL algorithm [31, 32] is applied to identify connected sub-parts of the word images. The second stage involves an approach for classification of the connected sub-parts into either of the Segment Further (SF) or Do Not Segment (DNS) classes using a rule based prior selection methodology and well-known MLP based classifier with a set of features extracted from these components. Finally, fuzzy features are used to identify potential segmentation points in an effective way on

the detected Matra region for subsequent extraction of constituted characters in the SF components. The basic steps of operations involved in the present work are illustrated in Figure 8.

## 4 Word Extraction Technique

### 4.1 Preprocessing

In this step, the documents are binarized by a simple adaptive *thresholding* technique, where the threshold is chosen as the *mean* of the *maximum* and *minimum* gray level values in each document image. All the binarized images are archived in DAT format, where the foreground and background pixels are represented as '1' and '0' respectively.

### 4.2 Component Finding

Here individual text lines are considered sequentially for the processing. All the connected components present in the text lines and their respective bounding boxes are obtained. A *component* or *segment* is defined as a connected region of black pixels in any document page. To identify such components, we have implemented an 8-way CCL algorithm [31,32] in the current work. An 8-way connected component may be defined in a way that every pixel can be reached from any other pixels through a combination of moves in only eight directions (i.e., up, down, left, right, or any of the four diagonal directions). After this, the components which are having very small size (less than a threshold) are discarded and their corresponding values in the binarized image are set to '0'.

### 4.3 Spiral Run Length Smearing Algorithm (SRLSA)

This is the main feature of our word extraction technique. In SRLSA, when a data pixel is found in the binarized image then the neighboring pixels are scanned for another data pixel in a spiral way. The scanning continues until either another data pixel is observed or it crosses a *threshold* distance from the starting data pixel. In the first case, the values of all the background pixels, which lie in the line joining the newly detected pixel and the original data pixel, are changed to '1' in the binarized image.

In the present work, SRLSA is applied separately in two directions as shown in Figure 9. Figure 9(a) shows that the smearing direction is started in eastward direction and Figure 9(b) shows that the same is started in northward direction. In

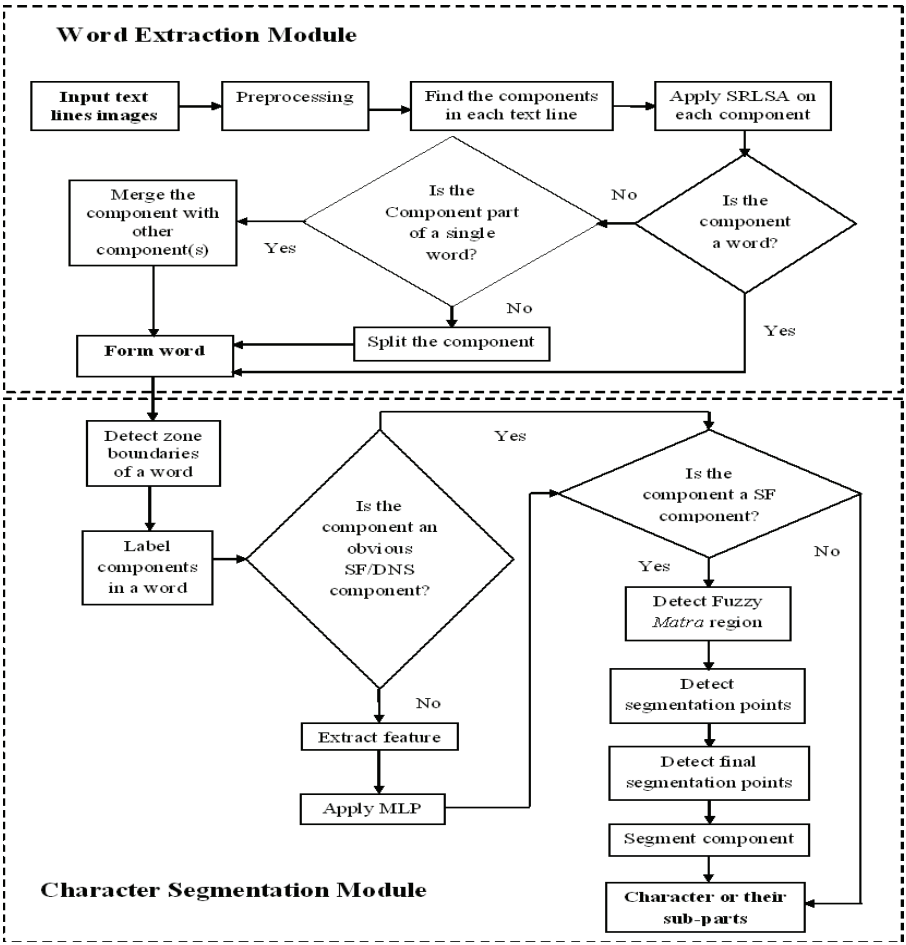
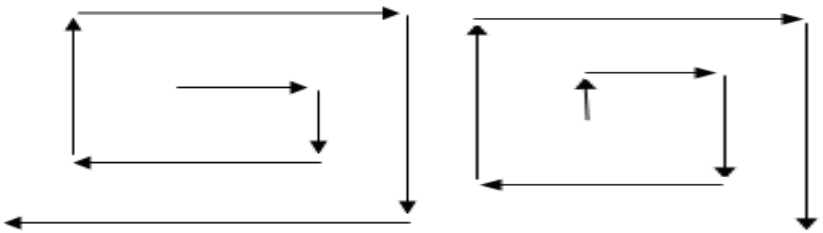



Figure 8. Basic steps involved in the present work.

printed Bangla script, the constituent characters in a word are generally connected through the Matra whereas in handwritten Bangla words, as mentioned earlier, Matras are not prominent and often discontinuous due to lifting up of pens. Thus in handwritten documents, number of connected components is much more than that of printed documents. Keeping this fact in mind, we have applied the SRLSA in the above two directions to incorporate all the nearby components which might be a part of a single word. Finally, their respective resultant binarized image is OR-ed to get the result.



(a) Smearing in East direction.

(b) Smearing in North direction.

Figure 9. Illustration of two variations of SRLSAs.

### 4.4 Word Formation

In this step, every pair of smeared components which overlap each other completely or partially and belong to the same text line is considered one by one. A threshold value is chosen and if the overlapping length for these components is *greater than* this threshold value then this pair of component is merged to form a single component.

### 4.5 Component Merging

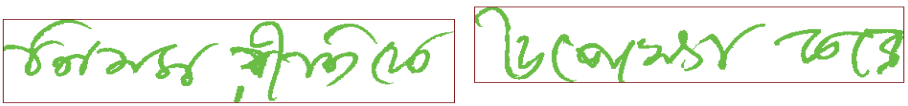
We have already mentioned that in handwritten documents, often components of a single word remain separated unusually due to the writing style of individuals. Our SRLSA technique sometime fails to connect these components, which requires to be merged as they belong to the same word image. At the start of this step, the average spacing between every two successive smeared components (scanning from left to right) are calculated. If their spacing is less than a certain percentage of the average spacing between the consecutive components in the entire document image, then said components are merged to form a single word component. In Figure 10(a) the pair of components, marked by dotted circular region, lying within a certain percentage of the average spacing between successive components of the entire text document image and they have been merged, which is illustrated in Figure 10(b).

### 4.6 Component Splitting

In handwritten documents, sometimes consecutive words get merged due to not lifting up of the pen at the proper place or non-uniform spacings among the words.

(a) Components before merging. (b) Components after merging.

Figure 10. An example of component merging case.



(a) Component without splitting.

(b) Component after splitting.

Figure 11. An example of component splitting case.

These components need to be segmented in order to get the correct word boundaries. At this point, the components having very wide horizontal length compared to the average horizontal length of the components present in the entire document image, are considered. These exceptionally large components are scanned from left to right and if there is any space within the component**,** which is greater than a certain percentage of the average spacing between the components in the entire document page, then these are broken down into two components. Figure 11(a) shows a sample text line segment where several words are identified as single word. Figure 11(b) shows successfully extracted word images of the text line segment, as shown in Figure 11(a), after the refinement.

After this, we obtain the final output of the word extraction process which would be useful for character segmentation technique.

## 5 Character Segmentation Technique

### 5.1 Detection of Zone Boundaries in a Word Image

Constituent characters or their sub-parts of words often extend above the common Matra or appear below the main character body. In the current work, we have

identified three adjacent horizontally partitioned zones (viz., upper, middle and lower) from each word image as shown in Figure 2(a). More specifically, the top row of the upper zone ($R_1$), the top row of the middle zone ($R_2$), the middle row of the middle zone ($R_3$), the bottom row of the middle zone ($R_4$) and the bottom row of the lower zone ($R_5$) are identified from the word image as shown in Figure 2(b).

A horizontal pixel scan of the word image from top towards bottom identifies the first row, with any black pixel, as the top row of the upper zone i.e., $R_1$. Similarly, a horizontal scan from bottom towards top identifies the first row, with any black pixel, as the bottom row of the lower zone i.e., $R_5$. Identification of the top and bottom row boundaries of the middle zone (a key decision for subsequent features extraction) is a challenging task in handwritten Bangla word.

In our earlier work [18], we scanned the whole image to calculate sum of all the lengths of horizontal runs of black pixels for each row and then estimated $R_2$ using those values. But sometimes this may give us misleading information. It may so happen because there are cases related to handwriting style of individual where the sum of maximum longest run length may appear anywhere in the word image and due to which $R_2$ is not estimated correctly as shown in Figure 5. Therefore, we have modified the technique for determination of $R_2$ as mentioned in [18].

We know that generally Matra of handwritten word images do not appear in the lower half of the image. So in the present work, to identify the common Matra of the word, horizontalness of each row is computed from the top to half of the word images i.e. from $R_1$ to $R_1 + (R_5 - R_1)/2$. Each black pixel of the word image in the said region is replaced by the length of the longest run of black pixels in horizontal direction by itself. Sum of the horizontal longest run values of all the pixels in a row is computed for each row of the word image. The row with the highest sum represents the row with maximum horizontalness [18]. This row signifies the possible upper boundary of the middle zone and we have called it as 1st approximation of the upper boundary of the middle zone ($R_{21}$). Then from the verticalness feature [18], we have estimated the 2nd approximation of $R_2$ and have called it $R_{22}$. Techniques involving the estimates of $R_{21}$ and $R_{22}$ are discussed in [18].

But even after estimating $R_{21}$ and $R_{22}$ we have observed that in few cases, the Matra regions are not estimated accurately (as shown in Figure 5). To address this issue, we have estimated another value of $R_2$ as the row containing the longest single run of black pixels and have called it as $R_{23}$. Finally, we have taken the average of the three $R_2$ approximations and have called it as $R_2^{\text{final}}$, such that $R_2^{\text{final}} = (R_{21} + R_{22} + R_{23})/3$. This new estimation of $R_2$ (involving three approximations) is observed to be more accurate in comparison to our prior work [18] involving two such approximations. We have taken $R_2^{\text{final}}$ as our final upper boundary ($R_2$) of middle zone for a handwritten word image.

Also we know that generally bottom row of the middle zone (i.e. $R_4$) of handwritten word images do not appear in the upper zone. So in our current work, to identify the $R_4$ of the word images, horizontal transition points between text and background pixels are computed from the middle to bottom of the word images i.e. from $(R_1 + (R_5 - R_1)/2)$ to $R_5$. In each row, starting from the middle row to the bottom row of word image, the sum of transition points between text pixel to background pixel and vice versa are computed. The average number of transition points in the lower half of the image is computed as eta ($\eta$). Now the 1st row from bottom row of lower zone to half of the word image, which has greater transition points than $\eta$, is identified as the bottom row of the middle zone (say $R_{41}$). Again, as in case of $R_2$, we have estimated $R_4$ from the verticalness feature [18] and have called it $R_{42}$. Then we have taken the average of $R_{41}$ and $R_{42}$ as the final $R_4$ i.e. $R_4^{\text{final}} = (R_{41} + R_{42})/2$. We have taken $R_4^{\text{final}}$ as bottom row of middle zone i.e. $R_4$. Finally, the middle row of the middle zone is taken as $R_3$ i.e. $R_3 = (R_2 + R_4)/2$.

### 5.2 CCL of Word Images

We have implemented an 8-way CCL algorithm [31, 32] in the current work for identifying the connected components within the word image. Identification of connected word components requires detection of the connected pixels therein and marking them with identical labels. CCL algorithm [31, 32] scans the word image pixel by pixel from left to right and from top to bottom. During scanning, it considers all 8 neighbors of each pixel. For each of the connected components, all its member pixels appearing in the sub-image are replaced by a single distinct symbol. This is done to complete labeling of the connected pixels in the image and to generate uniquely coded connected components. Figure 12 shows a word image with its three connected components. Each of such connected components is subsequently extracted for analysis.

### 5.3 Selection of SF and DNS Components

Among all the digitized word sub-parts generated after CCL algorithm, a decision is often required to identify only the components that need further segmentation because of the presence of many inherently segmented characters or their subparts in word images. Thus, all word components may not require further segmentation at all. These components are often classified into SF and DNS classes as shown in Figure 12. Segmentation of DNS components is an overhead as it causes over segmentation of word components. Therefore, selection of SF and DNS components not only minimizes the character isolation overhead but also the over segmentation probability. For this, we have developed here a two stage selection for SF and DNS class components. These stages are described in Subsections 5.3.1 and 5.3.2.

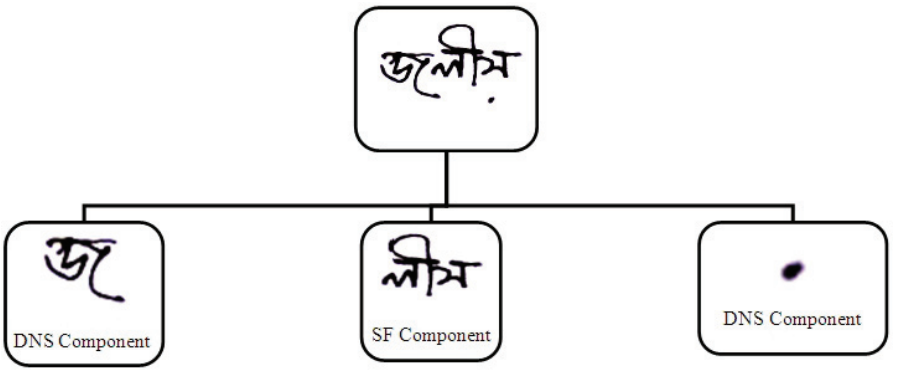



Figure 12. A sample word image and its three connected components.

**Initial Selection of Obvious SF and DNS Class Components**

In the work [18], MLP based classifier were used for such a classification problem. However, consideration of all word sub-parts, obtained by applying CCL, in the said classification algorithm not only increases computational overhead, but also leads to ambiguities in the selection leading to erroneous classification. To solve this problem, a pre-selection step is introduced in the present work which uses two scale-invariant threshold values that identify obvious SF and DNS class components, prior to MLP-based classification scheme.

In the current approach, all the word components are divided hypothetically into pieces by using a separating line (horizontal) along $R_3$ i.e. the middle line of the middle zone. The row, along which this separating line lies, is selected experimentally. After this hypothetical separation, the number of connected sub-components or pieces generated as a result of this division is counted. We have applied this number as the decision maker i.e., based on the number of generated sub-components the original component is categorized into one of the two types of classes, viz., DNS or SF. If the number of sub-components in a component is less than a threshold value $T_1$, then we have considered the component as a member of DNS class. On the other hand, if the same is greater than another threshold value $T_2$ then the component is considered as belonging to the SF class. Some of the sample components classified successfully using this thresholding technique are shown in Figure 13.

The components with number of sub-components ($n$) between $T_1$ and $T_2$, i.e. $T_1 < n < T_2$, are sent to a previously trained MLP classifier to accurately classify the components. This is done so, as decision-making on these components is not possible by using either $T_1$ or $T_2$. Experimentally, we have observed the values of

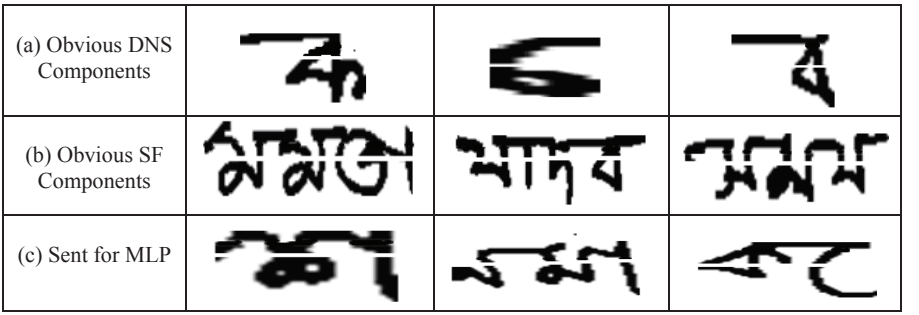



Figure 13. Sample connected component images of Bangla script words which are pre-classified as (a) Obvious DNS components, (b) Obvious SF components and (c) sent for MLP based classifier for subsequent SF/DNS class identification.

$T_1$ and $T_2$ as 2 and 5 respectively.

From the images of Figure 13, it is evident that this choice of $T_1$ is suitable for classification of the component containing a single character or its subpart as DNS components. Also, the choice of $T_2$ is done in such a way that, multiple touching characters or their sub-parts generate more number of components than $T_2$. These components are classified as SF components. In all remaining cases, ambiguities may exist and thus need sophisticated techniques such as MLP based classifier and associated feature vector.

**Classification of SF/DNS Components Using MLP Classifier**

In the present work, an MLP based classifier is used for classification of connected word components, which are not classified in the pre-processing stage, into either of the two classes to decide whether the given component needs to be further segmented or not, using the feature set mentioned in Table 1. The MLP based classifier designed for this work is trained with the Back Propagation (BP) algorithm. It minimizes the sum of the squared errors for the training samples by conducting a gradient descent search in the weight space. The number of neurons in a hidden layer in the same is also adjusted during its training. In the current methodology we have designed a new feature set containing 11 statistical features, as described in Table 1. The following discussions justify the choices of respective feature descriptors.

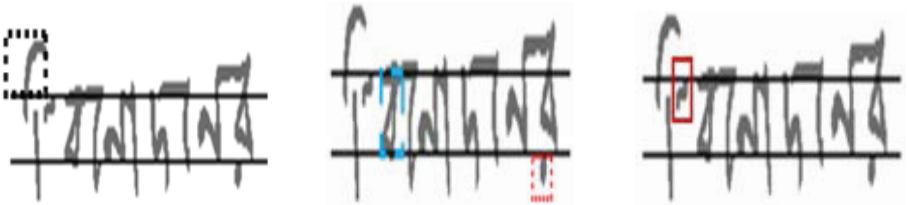
The higher value of feature F1 signifies that the component may belong to DNS class, as this component may have some part(s) in the upper zone of the word as shown in Figure 14(a). A similar explanation is applicable for the features F2 and F4 for the components in middle zone and the lower zone respectively and is

| Feature ID | Feature Description |
|---|---|
| F1 | Percentage height (w.r.t. the overall word height) of the component that appears upper zone of the word image |
| F2 | Percentage height (w.r.t. the overall word height) of the component that appears middle zone of the word image |
| F3 | Percentage height (w.r.t. the overall word height) of the component that appears lower half of the middle zone of the word image |
| F4 | Percentage height (w.r.t. the overall word height) of the component that appears lower zone of the word image |
| F5 | Maximum horizontalness of the component within the region $R_2$ to $R_4$ |
| F6 | Area of the component within the region $R_2$ to $R_4$ |
| F7 | Number of data pixel of the component within the region $R_2$ to $R_4$ |
| F8 | Number of data pixel of the component on $R_2$ |
| F9 | Width of the component along $R_2$ |
| F10 | Maximum width of the component within the region $R_2$ to $R_4$ |
| F11 | Number of segmentation-point clusters on the Matra region of the component |

Table 1. Feature vector and their description.

illustrated in Figure 14(b). The feature F3 is used to classify the noise segment (i.e. broken part(s) of Matra). These noise segments almost certainly appear partially in upper and/or middle zone as shown in Figure 14(c). Therefore, F3 values for these components will be zero.

Feature F5, i.e. maximum horizontalness feature, has been used in the work [18]. However, due to writing styles of individuals this feature value may be higher in the upper, lower or lower half of the middle zone if the ascendant (character sub-parts in the upper-zone of the word image) or descendant (character sub-parts in the lower-zone of the word image) is extended unnecessarily as shown in the Figure 15(a). Because of this, in the present work we have additionally used feature F10, i.e. maximum width of the component within the region $R_2$ to $R_4$ as shown in Figure 15(b). Lesser value of this feature implies the component may be categorized as DNS class component. In feature F6, as used in work [18], the

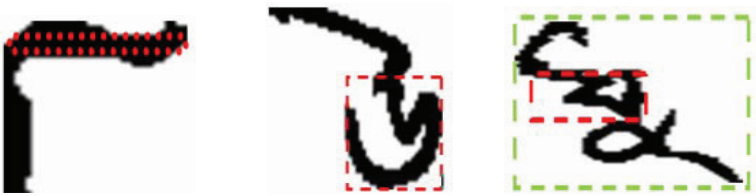


(a) Word component in the upper zone inside color box.

(b) Word components in the middle and lower zone inside color boxes.

(c) Noise component inside color box.

Figure 14. Illustration of features F1, F2, F3 and F4.



(a) Maximum horizontalness in the upper zone due to extended ascendant.

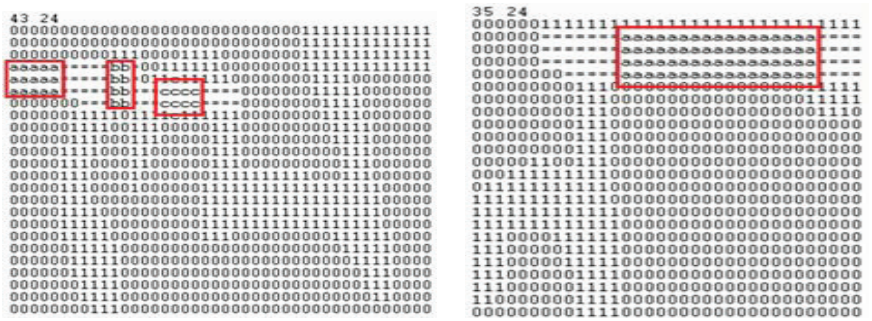
(b) Maximum width within the region $R_2$ to $R_4$.

(c) Word component with extended ascendant and descendant.

Figure 15. Illustration of features F5 and F6.

whole component was considered for area calculation. But this feature value may be higher for the component of DNS class due to extended ascendant and/or descendant as shown in Figure 15(c). For this reason, we have modified the feature value of F6 by considering only the area of interest, i.e. the area within the region $R_2$ to $R_4$ only. Due to the same reason, the feature value of F7 i.e. number of data pixels is also calculated only within the region $R_2$ to $R_4$. Higher the value of feature F7 more is the possibility of the component belonging to the class SF. Similarly, high value feature F8 implies more prominent and continuity of the Matra i.e. component will be a member of the SF class.

Again, due to cursive handwriting or discontinuity of Matra, the value of feature F9 may be lower for the components that need to be segmented further. That is why we have also taken feature F10 that gives the width of the component in the middle zone, i.e. vertical projection of the components within $R_2$ to $R_4$ along $R_2$



(a) (b)

Figure 16. Illustration of feature F11 (rectangular region indicates a cluster of segmentation points).

(central Matra row).

Feature F11 is the number of segmentation-point cluster. Often a component gets segmented to generate multiple, close segmentation points on the Matra region (Selection of Matra and segmentation point is discussed in Sections 5.4 and 5.5). Using 8-way connectivity, we have identified cluster of such segmentation points and the number such clusters is considered as a feature value. More is the number of clusters, higher is chance of the component classified as SF class. In the previous work [18], the number of segmentation-points was considered as feature. But more number of segmentation points may not always imply that the component needs to be segmented further. This is illustrated in Figure 16(a) and (b). Though Figure 16(b) contains more number of segmentation points than Figure 16(a) but the component in Figure 16(b) does not require further segmentation. To compute feature F11, potential segmentation points in the region $R_1$ to $R_3$ of connected components are to be determined first.

### 5.4 Determination of Matra Pixels Using a Fuzzy Membership Function and Horizontalness Feature for SF Components

The boundary between the sets of Matra pixels and non-Matra pixels in the region $R_1$ to $R_3$ is not distinct in practice. The black pixels lying over the line $R_2$ have got strongest membership to the set of Matra pixels. As they are away on both sides of the line $R_2$, their degree of belongingness to the set diminishes, as shown in Figure 17(a), through a membership function $\mu_{\text{MATRA}}(x)$. The exact expression

of $\mu_{\text{MATRA}}(x)$ is shown below:

$$\mu_{\text{MATRA}}(x) = \frac{1}{1 + \left|\frac{x-c}{a}\right|^{2b}} \tag{1}$$

where $c$ $(= R_2)$ denotes the center of the function, shown in Figure 17(a), and '$a$' and '$b$' are parameters of the equation. The value of 'a' is chosen as $|R_1 - R_2|/2$ for upper side of $R_2$ and for lower side of $R_2$ it is chosen as $|R_2 - R_3|/2$. The value of '$b$' is chosen as 1.

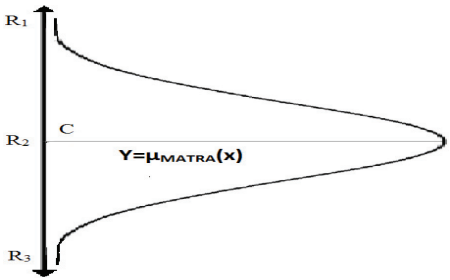
To finally determine whether a black pixel in region $R_1$ to $R_3$ is a Matra pixel, the product of the horizontalness [18] of the black line segments, on which the pixel lies, and its $\mu_{\text{MATRA}}$ value is computed. If the product exceeds the average value of all such products for all black pixels in the region $R_1$ to $R_3$ then the pixel is finally considered as Matra pixel here. All such Matra pixels constitute the Matra region.

### 5.5 Determination of Potential Segmentation Points Using Two Fuzzy Membership Functions for SF Components

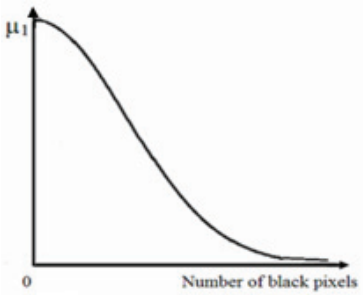
Potential segmentation points are basically Matra pixels, across which the component is to be fragmented vertically if it falls in the SF category. They are basically candidates for segmentation points until classification of the segment in SF class is completed. Potential segmentation points usually lie on the column positions along which the values of black pixel count are less. The less is the value of the black pixel count along the column position of a Matra pixel the higher is the degree of belongingness of the pixel to the class of potential segmentation points. To simulate this, a membership, $\mu_1$, as shown in Figure 17(b), is introduced here. The equation to this function is

$$\mu_1 = \frac{1}{1 + \left|\frac{x-c}{a}\right|^{2b}} \quad \text{for } x \geq 0. \tag{2}$$

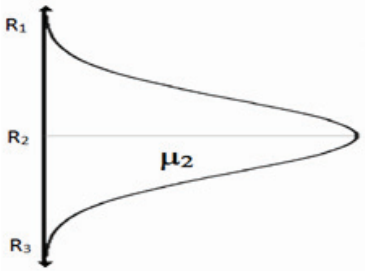
The values of parameters $a, b, c$ are chosen as follows: $c = 0, b = 1, a = \text{WM}$, where WM is the maximum vertical width of the Matra region, defined in Section 5.4. To ascertain a Matra pixel as a potential segmentation point, its distance from the line $R_2$ is considered here. The less is the distance the higher is its degree of belongingness to the set of potential segmentation points. Ideally, it would be on $R_2$. On this basis, another membership function $\mu_2$, is introduced here. The function is shown in Figure 17(c). The values of the parameters of $\mu_2$ are same as that of $\mu_{\text{MATRA}}$.




(a) The membership function for the set of *Matra* pixels.




(b) The membership function $\mu_1$.




(c) The membership function $\mu_2$.

Figure 17. The membership functions for determination of potential segmentation points.

To decide about whether a Matra pixel in region $R_1$ to $R_3$ would be a potential segmentation point, the average of $\mu_1$ and $\mu_2$ values are computed. If the average exceeds the mean of averages for all the Matra pixels in the region $R_1$ to $R_3$ then the said pixel is to be considered as a potential segmentation point. Feature F11, mentioned in Section 5.3, represents the total number clusters of all such pixels which are identified as potential segmentation points.

### 5.6 Identification of Actual Segmentation Points in the SF Components

For determination of actual segmentation points for SF components, there is always a trade-off between under/over segmentation of word images. In the current work, we have attempted to optimize between the two, with minimum loss of information. The issue of segmentation also becomes difficult in the presence of ascendants in the upper zone of the word component. For this reason, we have

further designed algorithm-steps to identify a single column for segmentation on the Matra region.

As already mentioned that the use of the fuzzy features often generate multiple, neighboring segmentation points along the Matra region. We have identified the cluster of such segmentation points using 8-neighbors CCL algorithm as illustrated in feature F11 selection. It may be observed from Figure 16(a) that actual segmentation points should not involve all the potential segmentation points in the cluster, rather focus on only the pixels that optimally separate the connected parts into different characters (or their sub-parts) of the word image.

Selection of points which accurately segment the word components (sub-parts) into their constituent characters or sub-parts is a challenging issue. In case of poor selection of such points, over-segmentation may occur during the segmentation process. As a result of these, characters of their sub-parts may be internally broken/segmented, leading to loss of information.

In the light of the above facts, we have selected the actual (more accurate) segmentation points from each cluster of segmentation points in the current work. There are two primary decisions to be taken for this purpose, firstly, selection of the row-boundaries for segmentation along specific columns on the Matra region and secondly, identification of the segmentation columns in each segmentation-cluster.

The algorithmic steps involved in this process are given below:

1. Check whether there is any ascendant in the word component under consideration. Estimation of the height of upper zone of the component does the checking. If the height of the said zone is exceeding some adaptively tuned threshold value $(0.2 * (R_4 - R_2))$, then it can be said that component has an ascendant part in the upper zone.

2. In either of the cases, the generated potential segmentation points are clustered and uniquely labeled using 8-way CCL algorithm. For each cluster, the following technique is applied to determine the segmentation column along which we can segment the word component under consideration:

   A) If there is no ascendant in the word component under consideration; calculate the sum of number of data pixels, Matra pixels and segmentation-point pixels for each column in the region from $R_1$ to $(R_3 - R_2)/2$. Otherwise, calculate the same for each column in the region from $(R_2 - R_1)/2$ to $(R_2 + (R_3 - R_2)/2)$.

   B) Consider the column for segmentation within the estimated region (row boundaries), which has the minimum sum, as calculated in step A.

Once the word components are segmented into constituent character or their sub-parts, again 8-way CCL algorithm is applied to separate each such word component. Finally, such segmented components will be considered for recognition as meaningful characters.

## 6 Experimental Results

### 6.1 Word Extraction

The data set of the Bangla script handwritten document images, which is used in our word extraction process, is available freely at `http://code.google.com/p/cmaterdb`. CMATERdb [33] is a pattern recognition database repository created at the "Center for Microprocessor Application for Training Education and Research" (CMATER) laboratory, Jadavpur University, India. CMATERdb1.1.1 is the Bangla script handwritten document database which contains 100 pages in its first version. In the ground truth data of the same, called CMATERgt1.1.1, the successive text lines are colored differently by applying an already developed text line extraction technique [34]. The possible errors that might have been generated in this process are corrected by a software tool called "GT Gen 1.1" which has also been developed at CMATER laboratory and freely available at `http://code.google.com/p/cmaterdb`. The description of the said database and the performance of the present word extraction methodology on it are shown in Table 2.

For manual evaluation of the Success Rate (SR) of word extraction technique, we have considered two types of errors, viz. under-segmented word and over-segmented word. If a single word component is erroneously broken down into two/more words, as shown in Figure 18, it is considered as an *over-segmentation*

| | |
|---|---|
| Number of document images (Page) | 100 |
| Total number of words present (T) | 15 730 |
| Average word width (in pixel) | 179.13 |
| Average word height (in pixel) | 68.06 |
| Total number of words extracted correctly | 13 530 |
| Under-segmented words (U) | 1256 |
| Over-segmented words (O) | 944 |
| **Average SR** | **86.01%** |

Table 2. Detail description of the experimental results on CMATERdb1.1.1 database.

Figure 18. An illustration of over-segmented word image: error is highlighted with dotted circular region.

error. Similarly, if two/more words are recognized as a single word, as shown in Figure 19, then it is considered as an *under-segmentation* error. In both the cases, such extracted words are also treated as wrongly extracted words. The total number of under-segmented and over-segmented words is considered in the estimation of the SR of the word extraction technique. More specifically,

$$\mathrm{SR} = ((T - (O + U)) * 100)/T, \tag{3}$$

where

$O$ =number of over-segmented words,
$U$ =number of under-segmented words,
$T$ =number of actual words present in the document page.

**Analysis of Erroneous/Successful Cases**

Out of the 100 document images processed, we have extracted 90 percent or more words successfully in 55 document images. In some of the images, SR is very poor due to the handwriting style. In these images either the average spacing between words of the text lines is very small or the writing style is cursive which has led to under-segmentation of the words.

Careful observation shows that the under-segmentation errors in the document in Figure 20, marked by dotted circular regions, are occurred as the consecutive words are connected or inter-word spacing is very less (see the first text line segment in Figure 20). In Figure 21, handwriting style is cursive with long strokes which makes inter-word spacings inconsistent. This results in under-segmentation

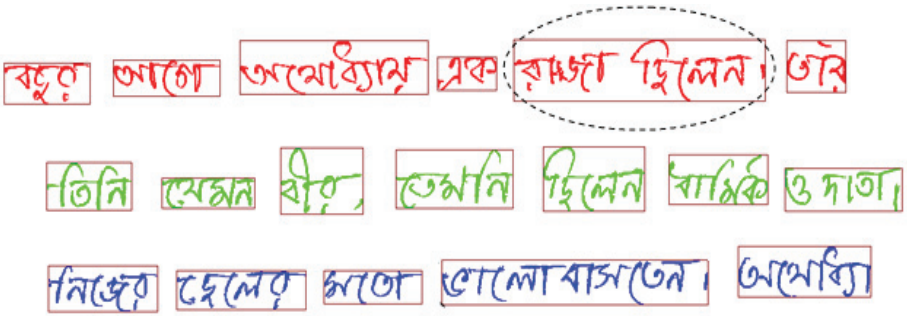


Figure 19. An illustration of under-segmented word image: error is highlighted with dotted circular region.

Figure 20. Example of error case: errors are highlighted with dotted circular regions.

of word images in the text lines. Figure 22 shows successfully extracted word images, by the present technique, on a sample document image. SR achieved by the present word extraction technique, using equation (3), on the said database is 86.01%.

### 6.2 Character Segmentation

The dataset, used for the experimentation of the character segmentation technique, consists of 750 handwritten Bangla word images. We have randomly selected this dataset from the CMATERdb1.1.1. The different improvements, in character segmentation, obtained by the present technique over our earlier techniques [17, 18] are discussed below.

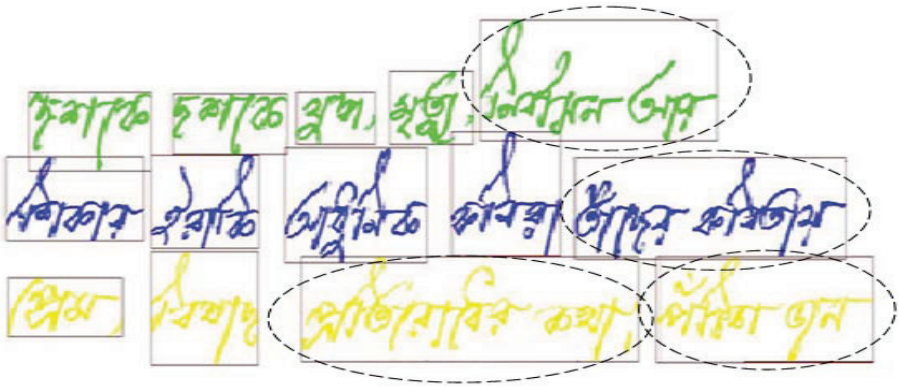


Figure 21. Example of error case: errors are highlighted with dotted circular regions.

### Improvement in Zone Boundary Selection

In the determination of upper boundary of the middle zone i.e. $R_2$, our present technique produces better result, which is demonstrated with few Bangla word images as shown in Figure 23. Figure 23(a) and (b) show the results obtained for the same with techniques discussed in [18] and the current one respectively.

### Improvement in the Classification of the Word Components into SF and DNS Components

In the first stage, to classify the connected segments into one of the two classes, namely, SF and DNS, a pre-selection procedure of obvious SF and DNS components is followed and rest of the components are sent to an MLP based classifier designed with *Back Propagation* (BP) learning algorithm. For preparation of the training and the test sets, a collection of 9000 of such connected segments of Bangla word images is formed. For 3-fold cross validation of results, the original dataset is divided into three equal mutually disjoint parts. For each fold of the test set, the corresponding training set is formed with the rest of the dataset. Thus three pairs of the test and the training sets are formed for three fold cross validation of results. In each of these pairs, the training and the test sets are of sizes 6000 samples and 3000 samples respectively.

For the present work, a single layer MLP, i.e., an MLP with one hidden layer is chosen. To design an MLP for classification of word components, several runs of BP algorithm with learning rate $(\eta) = 0.8$ and momentum term $(\alpha) = 0.7$ are executed with different number of neurons in its hidden layer.

Figure 22. Successfully extracted word images by the present technique on a sample document image.

The recognition performances of our present classification schema for SF and DNS for the 3-folds are 93.4%, 92% and 92.7%. Finally, the average success rate of these three sets of experiments is computed as **92.7%**, whereas result for classification of these components described in [18] gives the average success rate as 87.5%. Table 3 shows the comparative results for the classification of SF and DNS components. The entries in the table represents success rate in percentage. The quantity in brackets represents the number of hidden layer neurons.

Figures 24(a)–(e) illustrates some sample images classified by our SF/DNS classification schema. Figure 24(a) shows an example of DNS class component which is classified properly. Similarly, Figure 24(b) and Figure 24(c) show two sample images of SF class components which are classified correctly. Figure 24(d) and Figure 24(e) show two sample images of DNS class components which are misclassified as SF class components by our SF/DNS classification technique.

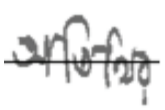
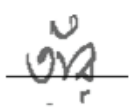
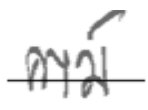
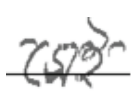
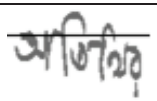
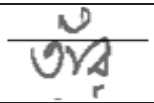
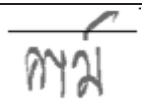
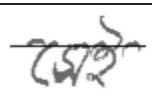
| | | | | |
|---|---|---|---|---|
| (a) $R_2$ using technique described in [18] |  |  |  |  |
| (b) $R_2$ using current technique |  |  |  |  |

Figure 23. Comparison of the techniques for the detection of the starting row of middle zone i.e. $R_2$.

| Test Set | Train Set | By the technique described in [18] | By the present technique |
|---|---|---|---|
| Fold 1 | Fold 2 + Fold 3 | 88.53% (18) | 93.40% (16) |
| Fold 2 | Fold 1 + Fold 3 | 87.07% (10) | 92.00% (14) |
| Fold 3 | Fold 1 + Fold 2 | 86.90% (12) | 92.70% (14) |

Table 3. Comparison chart for classification of SF and DNS components.

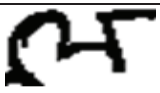
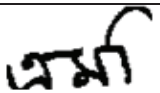
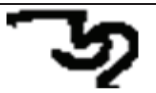
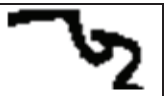
| | | | | |
|---|---|---|---|---|
|  |  |  |  |  |
| (a) | (b) | (c) | (d) | (e) |

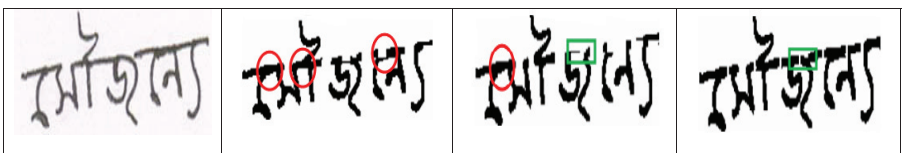
Figure 24. Some of the correctly classified and misclassified test samples: (a) successfully classified into DNS class; (b) and (c) successfully classified into SF class; (d) and (e) DNS components misclassified into SF class.

**Improvement in the Segmentation of Word Images**

The segmentation technique is applied to generate actual segmentation points of the SF class components. Figure 25(a) and Figure 26(a) show two sample word images considered for our experiment. Figure 25(b) and Figure 26(b) show corresponding segmentation results using the technique reported in [17] where SF and DNS classification algorithms are not applied. Similarly, Figure 25(c) and Figure 26(c) show the segmentation results after applying SF and DNS classification scheme, described in [18]. Finally, Figure 25(d) and Figure 26(d) show the results after applying the present work. From Figure 25(d) it is evident that the present character segmentation technique segments the word correctly, with minimal loss

(a) Original word image | (b) Segmentation result using technique reported in [17] | (c) Segmentation result using technique described in [18] | (d) Segmentation result using present technique

Figure 25. Segmentation results on a sample word image is shown, after applying segmentation techniques described in [17,18] and the current one (circular regions indicate under-segmentation errors and rectangular regions indicate over-segmentation error), showing the superiority of the current technique over the previous works.



(a) Original word image | (b) Segmentation result using technique reported in [17] | (c) Segmentation result using technique described in [18] | (d) Segmentation result using present technique

Figure 26. An under-segmentation error of the current work is shown in (d), despite improvements with respect to earlier works [17, 18] (circular regions indicate under-segmentation errors and rectangular region indicates over-segmentation error).

| Number of words | Number of actual components present | Work Reference | Number of components produced | Success Rate (%) |
|---|---|---|---|---|
| 750 | 3819 | [17] | 3249 | 85.07 |
| | | [18] | 3404 | 89.13 |
| | | **Current Work** | **3532** | **92.48** |

Table 4. A comparative description of the present character segmentation technique with our earlier works.

of information, in comparison to our earlier techniques [17, 18]. Figure 26(d) shows a failure case (over segmentation) of the present technique, which improves but not completely eliminates segmentation errors generated by our previous techniques.

The overall character segmentation accuracy on the 750 handwritten Bangla word images, is evaluated as **92.48%**, where as the techniques defined in [17] and [18] give success rate of 85.07% and 89.13% respectively on the same dataset. The comparison chart has been depicted in Table 4.

## 7 Conclusion

OCR of any handwritten/printed documents involves various stages ranging from scanning of the document to get its digital image, text line extraction, word extraction, character segmentation to the character recognition. In this paper, two important stages of OCR system viz., word extraction and character segmentation from the handwritten Bangla document images are reported.

At first, a modified RLSA technique, called SRLSA, is applied for the extraction of words of the unconstrained handwritten Bangla text document. We have successfully applied our SRLSA technique on the Bangla handwritten document image database CMATERdb1.1.1. This technique has helped us to overcome some of the drawbacks of standard horizontal and vertical RLSA techniques in word extraction procedure. In future, we would like to refine our word extraction technique by including a few more characteristics of the handwritten word images to get better results from the cursive handwritten document images.

In the second part of the work, we have presented a practical solution to the problem on how best word images of handwritten Bangla script can be segmented into constituent characters. Moreover, the technique can segment the words having discontinuity in Matra. It also optimizes the trade-off between under/over segmentation as Matra region and segmentation point clusters are estimated correctly. As a result, better word segmentation accuracy is achieved with minimal data loss. This character segmentation methodology may be also applied on the other Matra-based scripts, viz., Devanagri, Gurmukhi etc. However, there are further scopes of improvements of the present technique. An iterative implementation of the present technique, along with the existing segmentation algorithm, or designing more precious feature set for MLP may further improve the overall segmentation performance of handwritten Bangla word images in future. By varying the classifier or combining the results of the different classifiers, the improvement of the present technique is also possible.

We are also working on the incorporation the other Matra based scripts in the databases in which our techniques could be applied successfully. The work as a whole can be considered as a significant contribution towards the development of a computer OCR system for handwritten Bangla text document.

**Acknowledgments.** Authors are thankful to the “Center for Microprocessor Application for Training Education and Research” (CMATER), “Project on Storage Retrieval and Understanding of Video for Multimedia” (SRUVM) of Computer Science & Engineering Department, Jadavpur University, India, for providing infrastructure facilities during progress of the work. The work reported here, has

been partially funded by DST, Govt. of India, PURSE (Promotion of University Research and Scientific Excellence) Program.

## Bibliography

[1] Liwicki, M., Scherz, M., Bunke, H.: Word Extraction from On-Line Handwritten Text Lines. In: 18th International Conference on Pattern Recognition, Vol. 2, pp. 929–933, Hong Kong (2006).

[2] Bozinovic, R. M., Srihari, S. N.: Off-Line Cursive Script Word Recognition. In: IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 11 (1989), pp. 68–83.

[3] Blumenstein, M., Verma, B.: A New Segmentation Algorithm for Handwritten Word Recognition. In: International Joint Conference on Neural Networks, Vol. 4, pp. 2893–2898, Washington, DC , USA (1999).

[4] Chiang, J.-H.: Hybrid Neural Network Model in Handwritten Word Recognition. In: Neural Networks, Vol. 11(2) (1998), pp. 337–346.

[5] Gader, P., Whalen, M. , Ganzberger, M., Hepp, D.: Handprinted Word Recognition on a NIST Data Set. In: Machine Vision Applications, Vol. 8(1) (1995), pp. 31–40.

[6] Kurniawan, F., Khan, A. R., Mohamad, D.: Contour vs Non-Contour based Word Segmentation from Handwritten Text Lines: an experimental analysis. In: International Journal of Digital Content Technology and its Applications Vol. 3(2) (2009), pp. 127–131.

[7] Kurniawan, F., Rehman, A., Mohamad, D.: From contours to characters segmentation of cursive handwritten words with neural assistance. In: International Conference on Instrumentation, Communications, Information Technology, and Biomedical Engineering, pp. 1–4, Bandung (2009).

[8] Manmatha, R., Srimal, N.: Scale Space Technique for Word Segmentation in Handwritten Documents. In: Scale-Space Theories in Computer Vision, Lecture Notes in Computer Science, Vol. 1682/1999 (1999), pp. 22–33.

[9] Martin, G. L., Rashid, M., Pittman, J. A.: Integrated Segmentation and Recognition through Exhaustive Scans or Learned Saccadic Jumps. In: International Journal of Pattern Recognition and Artificial Intelligence, Vol. 7 (1993), pp. 831–847.

[10] Eastwood, B., Jennings, A., Harvey, A.: A Feature Based Neural Network Segmenter for Handwritten Words. In: International Conference on Computational Intelligence and Multimedia Applications, pp. 286–290, Gold Coast, Australia (1997).

[11] Wong, K. Y., Casey, R. G., Wahl, F. M.: Document analysis system. In: IBM Journal of Research and Development, Vol. 26 (1982), pp. 647–656.

[12] Priyanka, N., Pal, S., Mandal, R.: Line and Word Segmentation Approach for Printed Documents. In: IJCA Special Issue on “Recent Trends in Image Processing and Pattern Recognition”, RTIPPR (1), pp. 30–36, (2010).

[13] Strouthopoulos, C., Papamarkos, N., Chamzas, C.: PLA using RLSA and a neural network. In: Engineering Applications of Artificial Intelligence, Vol. 12(2) (1999), pp. 119–138.

[14] Sarkar, R., Moulik, S., Das, N., Basu, S., Nasipuri, M., Basu, D. K.: Word extraction from unconstrained handwritten Bangla document images using Spiral Run Length Smearing Algorithm. Accepted for publication in 5th Indian International Conference on Artificial Intelligence (IICAI-11) to be held in Tumkur, India, December 14–16 (2011)

[15] Abbass, Y. M., Fakher, W., Rashwan, M.: Arabic/English Identification in a hybrid complex documents images. In: ICGST International Conference on Graphics, Vision and Image Processing (GVIP), pp. 189–194, CICC, Cairo, Egypt (2005).

[16] Khurshid, K., Faure, C., Vincent, N.: Fusion of Word Spotting and Spatial Information for Figure Caption Retrieval in Historical Document Images. In: 10th International Conference on Document Analysis and Recognition, Barcelona, Spain, pp. 266–270, (2009).

[17] Basu, S., Sarkar, R., Das, N., Kundu, M., Nasipuri, M., Basu, D,.K.: A Fuzzy Technique for Segmentation of Handwritten Bangla Word Images. In: International Conference on Computing: Theory and Applications (ICCTA), pp. 427–432, Mar, Kolkata (2007).

[18] Sarkar, R., Das, N., Basu, S., Kundu, M., Nasipuri, M., Basu, D. K.: A two-stage approach for Segmentation of Handwritten Bangla word Images In: International Conference on Frontiers in Handwritten Recognition (ICFHR – 08), pp. 403–408, Aug, Canada (2008).

[19] Sarkar, R., Das, N., Basu, S., Kundu, M.: An improved fuzzy technique for segmentation of handwritten Bangla word images In: 2nd National Conference on Recent Trends in Information Systems (ReTIS-08), pp. 24–29, Feb, Kolkata (2008).

[20] Malakar, S., Ghosh, P., Sarkar, R., Das, N., Basu, S., Nasipuri, M.: An Improved Offline Handwritten Character Segmentation Algorithm for Bangla Script. Accepted for publication in 5th Indian International Conference on Artificial Intelligence (IICAI-11) to be held in Tumkur, India, December 14–16, (2011).

[21] Vishnu, A., Chaudhuri, B. B.: Segmentation of Bangla Handwritten Text into Characters by Recursive Contour Following. In: International Conference on Document Analysis and Recognition, pp. 402–405, (1999).

[22] Pal, U., Datta, S., Segmentation of Bangla Unconstrained Handwritten text In: International Conference on Document Analysis and Recognition, pp. 1128–1132, (2003).

[23] Garain, U., Chaudhuri, B. B.: Segmentation of touching characters in printed Devnagri and Bangla scripts using fuzzy multifactorial analysis. In: International Journal IEEE Trans. On Systems, Man and Cybernetics: Applications and Reviews, 22 (Part C), pp. 449–459 (2002).

[24] Sarkar, R., Sen, B., Das, N., Basu, S.: Fuzzification to the Technique of Unconstrained Handwritten Devanagari Script Segmentation. In: International Conference on Cognition and Recognition (ICCR-08), pp. 93–98, Apr, Mandya, Karnataka (2008).

[25] Sarkar, R., Sen, B., Das, N., Basu, S.: Handwritten Devanagari Script Segmentation: A non-linear Fuzzy Approach. In: IEEE Conference on AI Tools and Engineering (CD), Mar, Pune (2008).

[26] Sarkar, R., Agarwal, R., Maity, D., Bhattacharjee, J., Roy, D., Jain, J.: An Improved Fuzzy Approach to Segmentation for Handwritten Devanagari Scripts. In: IEEE WieNSET-07 (CD), Jun, Kolkata (2007).

[27] Sinha, R. M. K., Bansal, V.: On Devanagari Document Processing. In: IEEE International Conference on Systems, Man and Cybernetics, Vancouver, Canada (1995).

[28] Bansal, V., Sinha, R. M. K.: Segmentation of touching and fused Devanagari characters. In: International Journal Pattern Recognition, 35(4) (2002), pp. 875–893.

[29] Sharma, R. K., Singh, A.: Segmentation of Handwritten Text in Gurmukhi Script. In: International Journal Computer Science and Security, Vol. 2(3) (2008).

[30] Sharma, D. V., Lehal, G. S.: An Iterative Algorithm for Segmentation of Isolated Handwritten Words in Gurmukhi Script. In: International Conference on Pattern Recognition, pp. 1022–125, (2006).

[31] Yapa, R. D., Harada, K.: Connected component labeling algorithms for gray-scale images and evaluation of performance using digital mammograms. In: International Journal of Computer Science and Network Security, Vol. 8(6) (2008), pp. 33–41.

[32] Park, J., Carl, G. L., Chen, H.: Fast Connected Component Labeling Algorithm Using A Divide And Conquer Technique. In: International Conference on Computers and their Applications, pp. 373–376, New Orleans, Louisiana, USA (2000).

[33] Sarkar, R., Das, N., Basu, S., Kundu, M., Nasipuri, M., Basu, D. K.: CMATERdb1: a database of unconstrained handwritten Bangla and Bangla–English mixed script document image. In: International Journal on Document Analysis and Recognition (in press); DOI: 10.1007/s10032-011-0148-6.

[34] Khandelwal, A., Choudhury, P., Sarkar, R., Basu, S., Nasipuri, M., Das, N.: Text Line Segmentation for Unconstrained Handwritten Document Images Using Neighborhood Connected Component Analysis. In: International Conference on Pattern Recognition and Machine Intelligence, pp. 369–374, (2009).

Received June 25, 2011.

**Author information**

Ram Sarkar, Department of Computer Science and Engineering, Jadavpur University, Kolkata, India.
E-mail: raamsarkar@gmail.com

Samir Malakar, Department of Computer Application, MCKV Institute of Engineering, Liluah, Howrah, India.
E-mail: malakarsamir@gmail.com

Nibaran Das, Department of Computer Science and Engineering, Jadavpur University, Kolkata, India.
E-mail: nibaran@gmail.com

Subhadip Basu, Department of Computer Science and Engineering, Jadavpur University, Kolkata, India.
E-mail: bsubhadip@gmail.com

Mahantapas Kundu, Department of Computer Science and Engineering, Jadavpur University, Kolkata, India.
E-mail: mahantapas@gmail.com

Mita Nasipuri, Department of Computer Science and Engineering, Jadavpur University, Kolkata, India.
E-mail: mitanasipuri@gmail.com