A Modified Approach for the Segmentation of Unconstrained Cursive Modi Touching Characters Cluster

Manisha S. Deshmukh(✉) and Satish R. Kolhe

School of Computer Sciences,
Kavayitri Bahinabai Chaudhari North Maharashtra University,
Jalgaon, Maharashtra, India
manisha_d7@rediffmail.com, srkolhe2000@gmail.com

Abstract. In this paper, a robust character segmentation approach for cursive handwritten Modi script touching character cluster is presented. Prior to segmentation, the middle text region of the touching character cluster is separated by examining the location of Shirorekha and baseline. The middle text region is scrutinized for the estimation of ligature between two characters. Two different strategies are employed to find the location of the ligature. The selection of the strategy is based on the degree of connected component overlapratio. The foreground pixel intensity and vertical projection profile is scrutinized to segment the touching characters. The performance of the system is tested using the touching character clusters of the original archaic handwritten Modi documents. The proposed approach yields efficient touching characters cluster segmentation output and it is feasible to tackle most of the challenges in touching character cluster segmentation.

Keywords: Modi script · Touching character cluster segmentation · Overlap ratio

1 Introduction

The segmentation of the touching character is more challenging task if the script is handwritten, unconstrained, stylish and cursive. The Brahmi base ancient cursive Modi script is derived from Nagari family. This script is originated during 17th century and used in Maharashtra as administrative script up to 1950. It was used to write Marathi language as well as other languages as Hindi, Guajarati, Persian and so on. Mainly, this script was used for the fast writing. The text is written using Boru (Wooden stick/bird feather) and link over the Shirorekha without lifting the pen. Shirorekha is drawn before to write a text. Modi script text is not separated in sentences and words. Similarly, no punctuation mark is used to indicate the end of words or sentences. Thus, number of challenges are exhibits all over the segmentation of the characters as touching characters; nonuniform Shirorekha and base line; skewed or curved text lines; nonuniform

K. C. Santosh and B. Gawali (Eds.): RTIP2R 2020, CCIS 1380, pp. 431–444, 2021.
https://doi.org/10.1007/978-981-16-0507-9_36

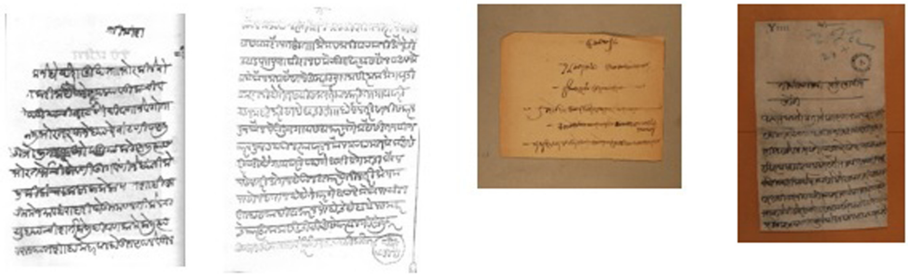

Fig. 1. Samples of original archaic handwritten degraded Modi Script document images.

text size and so on. Figure 1 shows the sample images of archaic original Modi script documents [15–19].

The number of segmentation challenges is under perception of the touching handwritten Modi character clusters as depicted in Fig. 2.

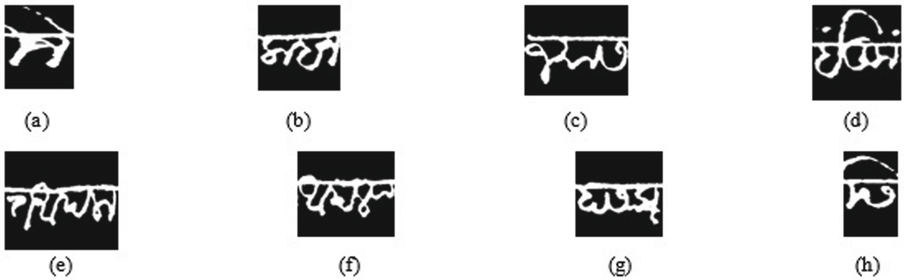

Fig. 2. Challenges in handwritten Modi touching character cluster segmentation. (a) Very dense touching character (b) Skewed Shirorekha (c) Entirely touching characters (d) Nonuniform base line (e) Nonuniform Shirorekha (f) Nonuniform size characters (g) Nonuniform touching characters (h) Looking like single character.

The accuracy of character recognition having one most important requisite aspect as the accuracy in character segmentation. However, the research work for touching Modi character cluster segmentation is not reported. Thus, there is essential to achieve the improvement in segmentation of touching handwritten Modi character cluster. The main objective of this research is to improve the accuracy of the freestyle, unconstrained, cursive and handwritten Modi touching character cluster segmentation. To improve the segmentation rate of the touching character cluster, a modified technique is proposed. The global and local zoning is employed with three types of analysis as: i) overlap ratio, ii) foreground pixel intensity and iii) vertical projection profile.

After giving the brief introduction, rest of the paper is clarified the proposed work as follows. Section 2 reviewed the related work. Section 3 discussed the framework of the handwritten Modi touching character cluster system. Details of the dataset which is used to check the performance of the proposed system is illustrated in Sect. 4. The experimental results are reported and discussed in Sect. 5. The conclusion is stated in the Sect. 6.

2 Literature Review

Generally, the character segmentation techniques are classified in two type explicit and implicit approach [2,5,6]. In the explicit approaches, segmentation point of the text part is finding out to separate the isolated characters. These approaches having the drawbacks given as follows [5].

- Over-segmentation or under-segmentation.
- Failed to segment the touching characters.
- Failed to segment overlapping characters.

In implicit approaches, to estimate the segmentation point intelligent system has been used. These techniques having two drawbacks given as follows [5].

- Failed to identify overlapping and touched text part.
- Requires huge training data and time which generates extensive overheads.

Handwritten character segmentation approaches found may vary significantly for the handwritten script like English, Chinese, Devanagari, Bengali, Guajarati etc. [1–9]. In the study of the handwritten character segmentation literature, it is found that the approaches are script specific and the number of issues are considered for character segmentation. It uses the prior knowledge about script's character structuring, writing style of script etc. [14]. The research presented in [13,16,18,19,21] proposes different types of strategies for the handwritten Devanagari script character segmentation. Graph distance theory-based approach for isolated, overlapping and touching Devanagari character segmentation is discussed in citech36ref10. This approach has two inefficiencies as: i) In locating accurate boundaries between characters through post-processing, and ii) Vertical cuts are gained for overlapped characters. The cursive handwritten touching English character segmentation techniques are presented in [10,12] and [23].

The research work presented in [13,16–21] proposes different types of strategies for the handwritten Devanagari script character segmentation. The work presented in [13,18] and [21] is the text-based approach used the structural properties to segments the touching Devanagari characters. This approach is script specific, parametric and having over-segmentation problem. It fails to segment the characters with broken left modifier and the vertical left modifier is too small. It also does not tackle with broken characters, touching characters. Graph distance theory-based approach for isolated, overlapping and touching Devanagari character segmentation is discussed in [16]. This approach has two inefficiencies as: i) In locating accurate boundaries between characters through post-processing, and ii) Vertical cuts are gained for overlapped characters. Morphological operation with minutiae detection algorithm is presented in [19] for the segmentation of Devanagari compound characters. A constrained based Devanagari touching character segmentation system is presented in [20] using analysis of bounding box with vertical bars technique. This system is failed to segment the broken and overlapping characters.

Kapoor S., & Verma V. [20] reported three opinions in the area of the character segmentation approaches as: i) The vertical projection profile-based character segmentation approaches are not suitable for free style, handwritten cursive, unconstrained overlapping characters; ii) The character segmentation approaches using Hidden Markov Model does not works well with overlapping characters; and iii) The water reservoir-based techniques are constrained based and are not able to segment the overlapping characters [22].

3 System Framework

A modified framework for Modi touching character segmentation system is illustrated in Fig. 3. The cluster of touching Modi characters is the input of this system. The proposed handwritten Modi touching character cluster segmentation procedure includes four stages, which are described as below.

3.1 Stage I

The initial stage of the character segmentation includes three processing sub steps. The input cluster images are in gray level. These images are converted into binary level by using Otsu's thresholding method [23]. The binarization of the images is desirable for the estimation of the location of the Shirorekha and text region. For the formation of the text region (Treg) the location of Shirorekha (Hline) and base line (Bline) is detected by calculating maximum and minimum peak of the horizontal projection profile using Eq. 1 and Eq. 2 respectively [2]. These locations are used for the separation of the text region of the Modi touching character cluster image as demonstrated in Fig. 4.

$$Hline = max_{(j=1:N)}\left(\sum_{i=1}^{M} C(i,j)\right) \tag{1}$$

$$Bline = min_{(j=1:N)}\left(\sum_{i=1}^{M} C(i,j)\right) \tag{2}$$

Here, C is the Modi touching characters cluster image of size M X N.

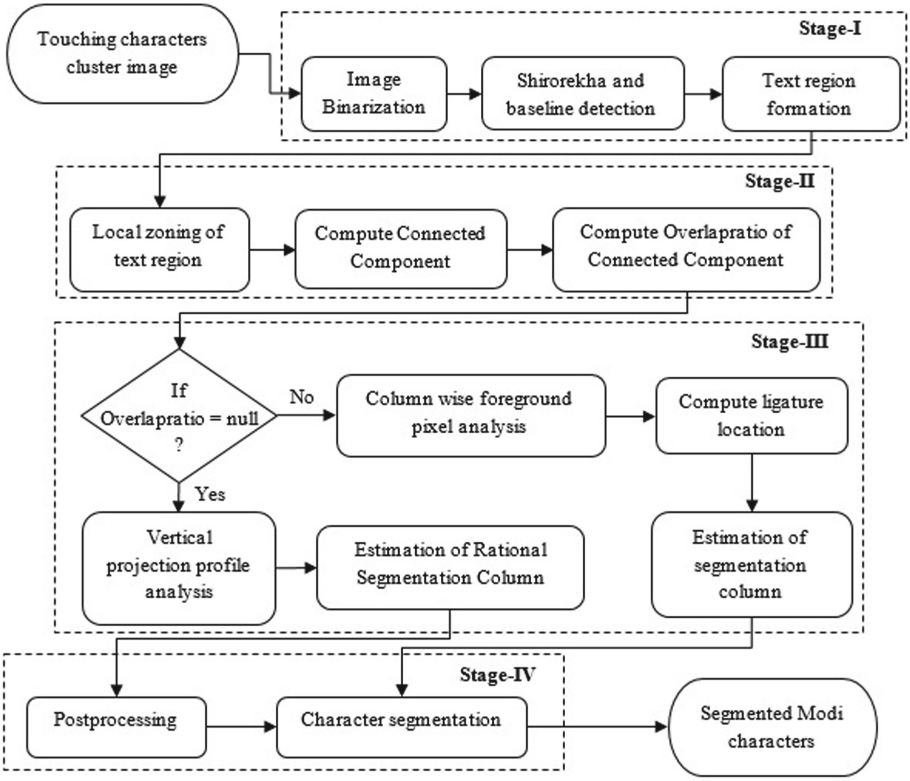

Fig. 3. Modified framework for the handwritten Modi touching character segmentation system.

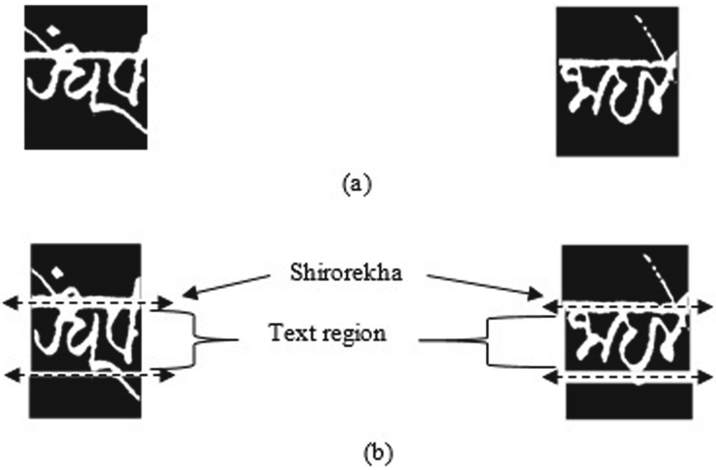

Fig. 4. The illustration of initial stage I of Modi character touching cluster segmentation (a) Input touching characters cluster (b) Shirorekha and base line detection and text region formation.

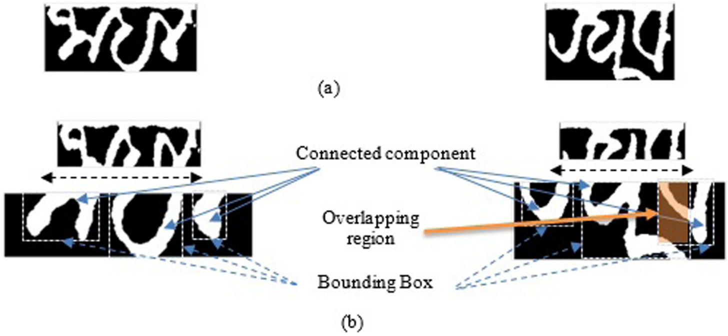

Fig. 5. Illustration of connected component and bounding box formation (a) Text region of the overlapping/touching characters cluster (b) Text region lower zone connected component and bounding box of the component.

3.2 Stage II

The next successive stage of the cluster segmentation includes local zoning of the text region (Treg). The text region is partitioned into two non-overlapping zones horizontally from the middle. The connected components (CC) from the lower region are intended as depicted in Fig. 5(b).

For each connected component the two elements are determined as its bounding box and area. The bounding box is rectangular region specified by a vector gives [X Y Width Height] of each component as shown in Fig. 5(b). Similarly, area (Areafig) is the actual number of pixels in the component computed which is calculated using Eq. 3.

$$Areafig(k) = \sum_{i=1}^{Height_k} \sum_{j=1}^{Width_k} CC_k(i,j) \qquad wherek = 1..Numberofconnectedcomponent \tag{3}$$

Further, the operative connected components (CCfig) are discriminated using the Eq. 4.

$$CCfig = \begin{cases} Areafig_k & Areafig_k \geq \sqrt{\mu(Areafig) + \sigma(Areafig)}/2 \\ none & \text{otherwise} \end{cases} \tag{4}$$

Where function μ and σ are mean and standard deviation of the area of the all connected components respectively. The overlap ratio between the bounding boxes of each operative connected component pair is computed using union ratio type as given in Eq. 5.

$$Oratio = \frac{CCfig_A \cap CCfig_B}{CCfig_A \cup CCfig_B} \tag{5}$$

Let $\mathrm{CCfig_A}$ and $\mathrm{CCfig_B}$ be the two connected components with $\mathrm{Box_A}$ and $\mathrm{Box_B}$ bounding box respectively. The overlap ratio (Oratio) between these two bounding boxes is an $\mathrm{M_oXN_o}$ matrix. Each element $(\mathrm{r_1}, \mathrm{r_2})$ of this matrix is corresponds to overlap ratio between row r_1 in Box_A and row r_2 in Box_B. The overlapratio is computed in the between 0 and 1 where 1 implies that a perfect overlap. This overlap ratio is exploited for the selection procedure of the touching cluster type. Clusters are classified in two classes as i) Partially touching clusters (Fig. 2 (b, d, e and g)) and iii) Entirely touching clustered (Fig. 2 (a, c and h)). The full process of this stage I and II is illustrated in Algorithm 1 from step1 to step8.

Algorithm1 SegTouch (Cfig)
Input: Cfig – Cluster of touching character's non-segmented text region
Output: Schar – Segmented characters

```
1: Cfig1 ⟵ img_Binarizarion (Cfig)                    // Gray to binary image conversion.
2: [H, W] ⟵ heightWidthOfImageOf ( Cfig1 )    // Find height and width of image
3: [Lb , Ls] ⟵ findSB (Cfig1, H, W)          //Find location of Shirorekha and baseline.
4: [Left, Right, I2] ⟵ findLeftRightEnd(Cfig1)    // Find left and right end location of the text
line.
5: [Upartfig, Mpartfig, Lpartfig] ⟵ separteUML(I2 , Lb, Ls)    // Find  text zone region
6: HalfMpartfig ⟵ extrLowerHalf(Mpartfig)                 // Extract lower half part of
Mpartfig
7: [CCfig, Areafig] ⟵ calAreaCC(HalfMpartfig)      // Calculate connected component and
area
                                          of each component in HalfMpartfig
8: Oratio ⟵  calOverlapRatio (CCfig)         // Compute the overlapping ratio of connected
                                       components which is operative
9: CntOratio ⟵ countOratio(Oratio)       // Count the region whose Oratio is greater than zero
10: if  isempty(CntOratio)              // Determine  the cluster is touching
       Function2 (Mpartfig, fig, h, w)
   else
       Function1 (Mpartfig, fig, h, w)
   endIF
11: return
```

3.3 Stage III

The overlapping ratio (CntOratio) is counted for the all the connected components. And this overlapping ratio is applied for the selection of the criteria of the touching character cluster segmentation process using Eq. 6. This is depicted in Algorithm 1 from step 9 to step10.

$$Criteria = \begin{cases} Function2() & CntOratio = null \\ Function1() & \text{otherwise} \end{cases} \tag{6}$$

A] Function1(): The partial touching character clusters are segmented using Function1 method. A window of size 3 X N_t is scanned through the text region. Related to each window first occurrence of the foreground pixels are searched from the bottom side. The ligature location (LocX) is finalized by scrutinized the location vector (Loc) with height of the text region M_t using Eq. 7.

$$LocX(s) = \begin{cases} Loc(i) & Loc(i) \leq M_t \wedge Loc \neq 0 \\ none & \text{otherwise} \end{cases} \tag{7}$$

Where i = 1...length(Loc). The ligature location vector is used to finalize the segmentation column. This character segmentation method is described in Algorithm 2.

```
Algorithm2 Function1 ( Mpartfig, fig, h, w)
Input: Mpartfig – The text region below to Shirorekha and upper to baseline of fig
        fig – Character cluster
        h – Height of Mpartfig
        w – Width of Mpartfig
Output: Schar – Segmented characters
1: for c ⟵ each three successive columns of Mpartfig
      lc ⟵ ( Mpartfig(c) )       // Extract the three successive columns region of Mpartfig
      for  j  ⟵  each row from bottom side
        if  isForegorundPixel( lc(j))              // Check for all foreground pixels
            Loc  ⟵ Loc U j+1
            break;
         endIf
   endFor
2: for l ⟵ each value of Loc                      // Estimate the ligature location
      if Loc(l) ≤ h/2 and Loc(l) ≠ 0
         LocX ⟵ LocX U Loc(l)
      endIf
   endFor
3: for l ⟵ each value of LocX -1                  // Calculate estimated character region locations
      if LocX(l+1) - LocX(l+1) ≠ 3
         SegLoc ⟵ SegLoc U LocX(l)
      endIf
   endFor
4: if size(SegLoc) =1                 / Check for the if character cluster is contain only two
characters
        SegLoc=1 U SegLoc;
   endIf
   if SegLoc(x) <= 2
       SegLoc= SegLoc U w;
   endIf
5: Schar ⟵ segCharacter(fig,Fec)
6: return
```

B] Function2(): The entirely touching characters clusters are segmented using vertical projection profile-based method. The vertical projection profile (Vpp) of the text region (Mpartfig) is calculated using Eq. 8.

$$Vpp = \sum_{i=1}^{M_t} Mpartfig(i,j) \qquad where\, 1 \leq j \geq N_t \tag{8}$$

The location vector (Loc) is formed by using the Eq. 9.

$$Loc(s) = \begin{cases} i & Vpp(i) \geq \mu(Vpp) + \sigma(Vpp) \\ none & \text{otherwise} \end{cases} \tag{9}$$

Where i = 1..Nt.Further, the estimated location vector (Loc) is analysed in the formation of the Rational segmentation column (SegLoc). The illustration of entirely touching characters cluster segmentation procedure is described in Algorithm 3.

```
Algorithm3 Function2 (Mpartfig, fig, h, w)
Input: Mpartfig – The text region below to Shirorekha and upper to baseline of fig
       fig – Character cluster
       h – Height of Mpartfig
       w – Width of Mpartfig
Output: Schar – Segmented characters
1: Vpp ⟵ VerticalPP( Mpartfig )   // Compute horizontal projection profile
2: for c ⟵ each column of Mpartfig
       if Vpp(c) ≥ mean(Vpp)+ std(Vpp)      // Check for ligature
          Loc ⟵ Loc U c
       endIf
   endFor
3: for l ⟵ each value of Loc                // Estimate Rational segmentation column
location
      if Loc(l) and Loc(l+1) are not consecutive
         SegLoc ⟵ SegLoc U Loc(l)
      endIf
4: Fec ⟵ postProcessing (SegLoc) //  Compute the Finalize segmentation column location
5: Schar ⟵ segCharacter(fig,Fec)         // Segment the character
6: return
```

3.4 Stage IV

The handwritten Modi script document are highly degraded due to number of reasons like aging, writing material and so on. Consequently, there may be under segmentation problem is occurred during the estimation of the Rational segmentation column. To reduce this problem, the Finalize Segmentation Columns (FSCs) is decided by analysing the Rational segmentation column (SegLoc) vector using the postprocessing method described in [2].

4 Dataset

The archaic original handwritten Modi script documents collected from the various archaeological places as Rajwade Sanshodhan Mandir, Dhule, Shri Samartha Vagdevta Mandir, Dhule etc., are used to test and evaluate the performance of the proposed system. The dataset contains 2540 document images. These document images already preprocessed means de-noised and globally de-skewed using the approaches presented in [25] and [1] respectively. The preprocessed document images are segmented in the text lines using the approach presented in [24]. The Modi text lines are segmented into the isolated Modi characters and touching characters clusters using the approach presented in [2]. These Modi touching characters clusters are the input of the proposed system.

5 Results and Discussions

The performance of the proposed handwritten Modi touching character cluster segmentation method is evaluated using a similar evaluation strategy that was illustrated in [2,28]. The evaluation method uses five evaluation factors as: Correct segmentation rate (CSR), Successful Segmentation Rate (SSR), precision, recall and F-measure illustrated in Eqs. 10–14 respectively. These aspects are calculated by counting the number of matches between the resultant segmented characters by the algorithm and ground truth characters in text line segments [2,28].

$$CSR = \frac{NC_R - (NC_c + NC_B)}{NC_G} \times 100 \tag{10}$$

$$SSR = \frac{NC_I}{NC_R} \times 100 \tag{11}$$

$$Precision = \frac{NC_I + NC_C}{NC_R} \tag{12}$$

$$Recall = \frac{NC_I + NC_C}{NC_G} \tag{13}$$

$$Precision = 2 \times \frac{Recall \times Precision}{Recall + Precision} \tag{14}$$

where,
NC_G: Number of ground truth characters.
NC_R: Number of resultant characters.
NC_I: Number of correctly segmented isolated characters.
NC_C: Number of overlapping or touching characters clusters.
NC_B: Number of incorrectly segmented characters.

Ground truth values are not available for the testing the performance of the handwritten Modi touching character cluster segmentation system. Thus, the numbers of 2249 Modi touching character clusters in 1072 text lines of ninety-two Modi documents are manually calculated and ground truth values are evaluated. Table 1 gives the result of touching Modi character cluster segmentation.

Table 1. The Modi touching character cluster segmentation result.

No. of clusters	NC_G	NC_R	NC_C	NC_B	NC_I	SSR(%)	CSR(%)	Precision	Recall	F-measure
1072	2249	2412	206	189	2017	89.68	83.62	0.9	0.9	0.9

The Successful Segmentation Rate (SSR) and Correct Segmentation Rate (CSR) are achieved using the proposed method for the Modi touching character cluster segmentation as 89.68% and 83.62% respectively. Partial touching Modi character clusters are segmented efficiently by proposed method. There are two issues are found about the bad segmentation. The first issue is with very tightly overlapping characters with touching characters exist in the same cluster. The second issue is the broken touching character as shown in Fig. 6(b). Still very tightly coupled character clusters and overlapping character clusters are not segmented properly. Figures 6, 7, 8 and 9 shows the examples of the segmented touching Modi character clusters.

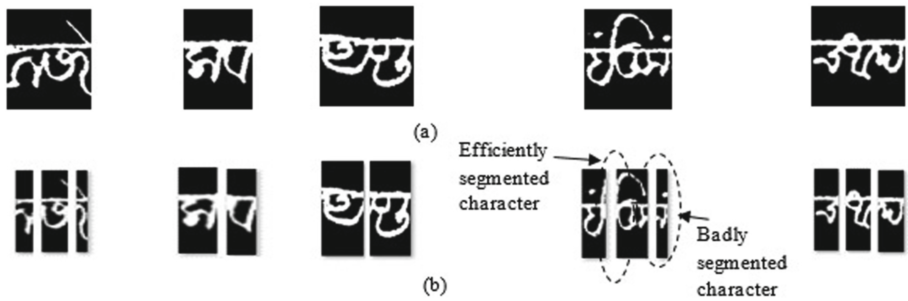

Fig. 6. Example of Modi script character segmentation of a Modi touching characters cluster (a) Modi touching character clusters (b) Segmented Modi touching character clusters.

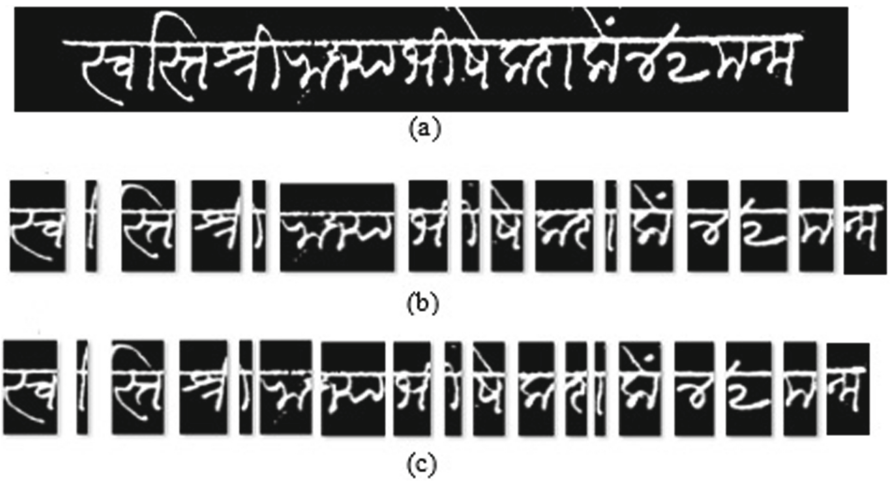

Fig. 7. Step wise example of Modi script character segmentation of a Modi document text line (a) Segmented text lines of a Modi Document (b) Segmented touching character clusters and isolated characters (c) Segmented characters of text lines.

Fig. 8. Example of Modi script character segmentation of a Modi document text line.

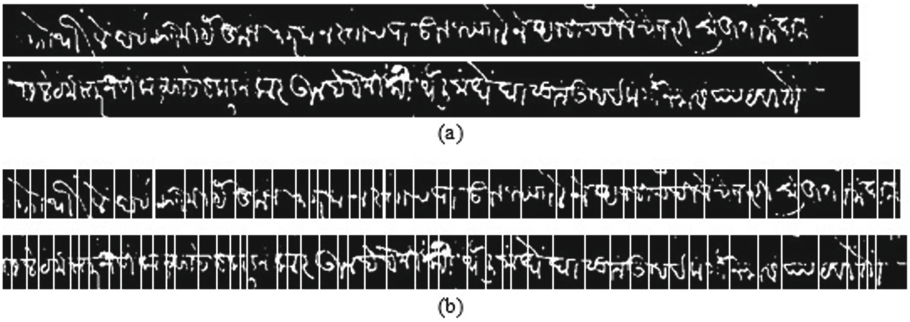

Fig. 9. Modi character segmentation (Broken characters and Shirorekha, non-uniform base line and Shirorekha) (a) Input Modi text lines (b) Segmented Modi characters.

6 Conclusions

The connected component overlapratio analysis effectively works on the different problems in touching character cluster segmentation. The proposed handwritten Modi touching character cluster segmentation approach is expertly dealing with the non-uniform Shirorekha and baseline. The results obtained proved that the presented character segmentation approach efficiently tackled with the character segmentation challenges like broken character, broken Shirorekha, variable size characters, variable distance between characters, skewed text lines and curved text lines. Highly degraded touching characters ligature is efficiently identified. The experimental results indicates that the proposed system is efficient to tackle the challenges in handwritten Modi touching character cluster segmentation. Hence, this system can be a part of the archaic handwritten freestyle and unconstrained Modi document segmentation system. Currently the work on the overlapping characters cluster segmentation is in progress.

References

1. Obaidullah, S.M., Halder, C., Santosh, K.C., Das, N., Roy, K.: PHDIndic_11: page-level handwritten document image dataset of 11 official Indic scripts for script identification. Multimed. Tools Appl. **77**(2), 1643–1678 (2017). https://doi.org/10.1007/s11042-017-4373-y
2. Obaidullah, S.M., Santosh, K.C., Halder, C., Das, N., Roy, K.: Automatic Indic script identification from handwritten documents: page, block, line and word-level approach. Int. J. Mach. Learn. Cybern. **10**(1), 87–106 (2017). https://doi.org/10.1007/s13042-017-0702-8
3. Obaidullah, S.M., Santosh, K.C., Das, N., Halder, C., Roy, K.: Handwritten Indic script identification in multi-script document images: a survey. Int. J. Pattern Recognit. Artif. Intell. **32**(10), 1856012 (2018)
4. Choudhary A., Rishi, R., Ahlawat, S: A new character segmentation approach for off-line cursive handwritten words. Proc. Comput. Sci. **17**, 88–95 (2013)

5. Kumar, M., Jindal, M.K., Sharma, R.K.: Segmentation of isolated and touching characters in offline handwritten Gurmukhi script recognition. Int. J. Inf. Technol. Comput. Sci. **6**(2), 58–63 (2014)
6. Kurniawan, F., Rahim, M.S.M., Daman, D., Rehman, A., Mohamad, D., Mariyam, S.: Region-based touched character segmentation in handwritten words. Int. J. Innov. Comput. Inf. Control **7**(6), 3107–3120 (2011)
7. Garg, N.K., Kaur, L., Jindal, M.K.: Segmentation of touching modifiers and consonants in middle region of handwritten Hindi text. Pattern Recognit. Image Anal. **25**(3), 413–417 (2015). https://doi.org/10.1134/S1054661815030050
8. Saba, T., Rehman, A., Elarbi-Boudihir, M.: Methods and strategies on off-line cursive touched characters segmentation: a directional review. Artif. Intell. Rev. **42**(4), 1047–1066 (2011). https://doi.org/10.1007/s10462-011-9271-5
9. Sharma, P., Sachan, M.K.: A review on character segmentation of touching and half character in handwritten Hindi text. Int. J. Adv. Res. Comput. Sci. **8**(3), 1078–1083 (2017)
10. Jindal, K., Kumar, R.: A novel shape-based character segmentation method for Devanagari script. Arabian J. Sci. Eng. **42**(8), 3221–3228 (2017). https://doi.org/10.1007/s13369-017-2420-7
11. Palakollu, S., Dhir, R., Rani, R.: Handwritten Hindi text segmentation techniques for lines and characters. In: Proceedings of the World Congress on Engineering and Computer Science, vol. 1, pp. 24–26 (2012)
12. Bag, S., Krishna, A.: Character segmentation of Hindi unconstrained handwritten words. In: Barneva, R.P., Bhattacharya, B.B., Brimkov, V.E. (eds.) IWCIA 2015. LNCS, vol. 9448, pp. 247–260. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-26145-4_18
13. Golait Snehal S., Malik L.: Handwritten Marathi compound character segmentation using minutiae detection algorithm. Proc. Comput. Sci. **87**, 18–24 (2016)
14. Kapoor S.,Verma V.: Fragmentation of handwritten touching characters in Devanagari script. Int. J. Inf. Technol. Model. Comput. (IJITMC) **2** 11–21 (2014)
15. Behera, S., Pradhan, A., Majhi, B.: A novel clustering based fuzzy approach for character segmentation in handwritten Odia scripts. In: 2017 Fourth International Conference on Image Information Processing (ICIIP), pp. 1–6. IEEE, December 2017
16. Kavitha, A.S., Shivakumara, P., Kumar, G.H., Lu, T.: A new watershed model based system for character segmentation in degraded text lines. AEU-Int. J. Electron. Commun. **71**, 45–52 (2017)
17. Otsu, N.: A threshold selection method from gray level histogram. IEEE Trans. Syst. Man Cybern. **19**(1), 62–66 (1979)
18. Deshmukh, M.S., Patil, M.P., Kolhe, S.R.: A hybrid text line segmentation approach for the ancient handwritten unconstrained freestyle Modi script documents. Imaging Sci. J. **66**(7), 433–442 (2018)
19. Deshmukh M.S., Kolhe, S.R.: A hybrid character segmentation approach for cursive unconstrained handwritten historical Modi script documents. In: International Conference on Sustainable Computing in Science, Technology & Management (SUSCOM-2019). SSRN Elsevier Digital Library (2019)
20. Deshmukh, M.S., Patil, M.P., Kolhe, S.R.: The divide-and-conquer based algorithm to detect and correct the skew angle in the old age historical handwritten Modi Lipi documents. Int. J. Comput. Sci. Appl. **14**(2), 47–63 (2017)

21. Deshmukh, M.S., Patil, M.P., Kolhe, S.R.: A dynamic statistical nonparametric cleaning and enhancement system for highly degraded ancient handwritten Modi Lipi documents. In: 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 1545–1551. IEEE, September 2017
22. Deshmukh, M.S., Manoj, P.P., Satish, R.K.: Offline handwritten Modi numerals recognition using chain code. In: Proceedings of the Third International Symposium on Women in Computing and Informatics. ACM (2015)
23. Kavallieratou E., Stamatatos E., Fakotakis N., Kokkinakis G.: Handwritten character segmentation using transformation-based learning. In: ICPR, p. 2634. IEEE, September 2000
24. Peng, G., Yu, P., Li, H., Li, H., Zhu, X.: A character segmentation algorithm for the palm leaf manuscripts. In: 2017 2nd IEEE International Conference on Computational Intelligence and Applications (ICCIA), pp. 354–358. IEEE, September 2017
25. Obaidullah, S.M., Santosh, K.C., Das, N., Halder, C., Roy, K.: Handwritten Indic script identification in multi-script document images: a survey. Int. J. Pattern Recognit. Artif. Intell. **32**(10), 1856012:1–1856012:26 (2018)
26. Halder, C., Obaidullah, S.M., Santosh, K.C., Roy, K.: Content independent writer identification on Bangla script: a document level approach. Int. J. Pattern Recognit. Artif. Intell. **32**(9), 1856011:1–1856011:24 (2018)
27. Mukherjee, H., Obaidullah, S.M., Santosh, K.C., Phadikar, S., Roy, K.: Line spectral frequency-based features and extreme learning machine for voice activity detection from audio signal. Int. J. Speech Technol. **21**(4), 753–760 (2018). https://doi.org/10.1007/s10772-018-9525-6
28. Santosh, K.C., Borra, S., Joshi, A., Dey, N.: Special section: advances in speech, music and audio signal processing (Articles 1–13). Int. J. Speech Technol. **22**(2), 293–294 (2019)