International Journal of Advanced Computer Technology (IJACT)

ISSN:2319-7900

SEGMENTATION OF HANDWRITTEN KANNADA TEXT DOCUMENT THROUGH COMPUTATION OF STANDARD ERROR AND WEIGHTED BUCKET ALGORITHM

A. Sunanda Dixit, Assistant Professor, ISE Department, DSCE, Bangalore.
B. S.Ranjitha, Final BE(ECE), Bangalore Institute of Technology, Bangalore.
C. Dr .H. N. Suresh, & Research coordinator, Dept. of IT, Bangalore Institute of Technology, Bangalore.

Abstract

Recognition and Segmentation of handwritten and scanned documentation images into text lines form a crucial base for Handwritten Character Recognition (HCR) / Optical Character Recognition (OCR) systems. The magnitude of the complexity increases as we move across text line segmentation process from printed text images towards a handwritten text images. Further implications of South Indian languages such as Kannada, the technology for segmentation is compounded for its complexity owing to the nature of the language that poses challenges due to its skewedness in lines, style of writing, split characters, modifiers as well as the gaps found corresponding to inter and intra word. The paper presents and defines a novel approach for unconstrained text line segmentation of document image scripts that are handwritten in Kannada language. The proposed approach computes the standard error of the sampling distribution for the components in the image and uses this as a reference to formulate a character seed or syllable. The character seed so formed are linked using weighted bucketing algorithm to form a line segment. This concept is implemented and the results obtained from this approach are analyzed. Experiments conducted evidences that the proposed algorithm achieves high level of accuracy for detecting unconstrained text line of handwritten script document images for Kannada language.

Keywords- Centroid, component syllables, Handwritten Text Documents, Weighted bucketing, Sstandard deviation.

Introduction

Last few decades has seen a tremendous progress in the field of Character Recognition (CR) to a level, adequate to generate applications that are driven specific to technology. But however technology for Indian languages remains an open challenge. Huge archives for Kannada handwritten documents persists and are still to be exploited by OCR systems. Digital document analysis is gaining popularity for Indian scripts and rapid increase in computational technique enables a significant role in methodologies for extracting information. Text lines in most of the handwritten documents are typically skewed and compact. Because of the overlapping and touching characters the complexity of these documents is high and, automatic text line segmentation technique remains an open research field.

Unconstrained text line segmentation has received tremendous focus as much research works are being carried out for handwritten document images. While talking about handwritten scripts there are variations in skew angles, inconsistent lines and dissimilar fonts. Many works has been done to resolve these issues based on the logic of splitting the document vertically and identifying the textual zones[1], algorithm based on the computation of an information content level [2], spatial sub-domains and average character height estimation [3] and technique based on a generalized adaptive local connectivity map (ALCM) using a steerable directional filter[4]. When dealing with Indian languages the intrinsic assumptions made by such algorithms seldom holds good. Indian languages that are classified as Devanagari script are complicated in terms of structure and computation [5] [6]. For segmentation of Indian script documentation Images several methods classified as dissection have been proposed [7]. Other methods are also focused on projection profile, that are inclined towards white space analysis [8] and connected component based [9] [10]. Horizontal segmentation of lines in Chinese handwritten texts is proposed in[12]. Internal slant is used for projecting the blocks. Separated blocks are painted using painting algorithm which leads to integration between the blocks. Edge detection and border detection is performed. The text lines are connected using Fuzzy triangles. Detecting the skew angle is proposed using fan filter to decompose the image to its components. It is designed in different levels and size to decompose the image to its components. The coefficient which holds the highest energy will be achieved to detect the skew angle of a text image [13].

This research work presents a new methodology for Kannada language which considers the interaction between the text lines and covers the regional property of each character representing the line. The solution algorithm defines a novel technique to precisely identify and formulate the composite character syllable. The combinational or grouping technique to form the syllable shall minimize the ambiguities preceded during line segmentation process. Thus text-lines are extracted from a scanned Kannada document image by integrating the character syllable so formed to represent a line using weighted bucketing algorithm. The paper is categorically structured into sections. The core aspect of Kannada script is captured in Section II. The methodology focused is provided in section III. Section IV depicts the core technique for Kannada text line segmentation. The dataset used for experimenting as well as Result analysis is presented in Section V. The conclusion and summarization for work is described in Section VI.

Concept of Kannada Script

Kannada is an ancient language spoken in Southern parts of India. The language defines 47 characters (13 vowels and 34 consonants). The scripts are composed of these characters and in conjunction with consonant modifiers (vaththus) to form a meaningful word. These vaththus or glyphs that are combined to represent syllables are quiet frequently used in the language. Segmentation of such syllables are difficult and needs more logic in comparison with Latin based scripts like English. Some of the complex syllables are listed below Figure 1

ವ್ಯ ತ್ವ ಜ್ಞಾ ಜ್ಯೊ

Figure 1. Complex Syllables of Kannada Langauge

The complexity of Kannada script is further increased due to the variance in the spatial features of the characters especially as they are handwritten. Immense differences can be found in terms of the density of the strokes, the length of the strokes as well as the number of stokes employed to define a feature of the character. There is a fine difference the way characters are represented in a printed text versus handwritten.

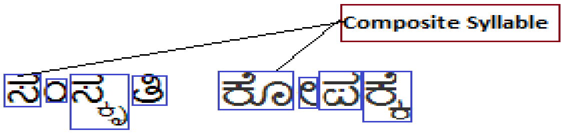


Figure 2. Kannada Printed Script

Here we consider couple of words from Kannada script to understand the nature of the problem. The Figure 2 depicts Kannada words in printed text format. As can be observed, each character is continuous and complete. The spacing is very systematic and precise, thus the componentization process yields a meaningful syllable.

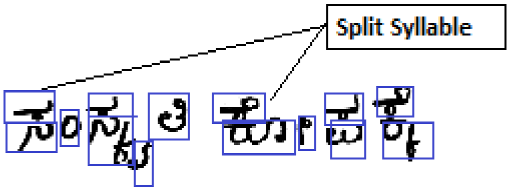


Figure 3. Kannada Handwritten Script

In comparison, we have the same words that are handwritten, shown in Figure 3. It is evident that the characters are disjointed. The spacing is not even and as such when the image is componentized, we get more blocks. These disjointed blocks adds to complexity resulting in incorrect lines while segmentation.

Methodology

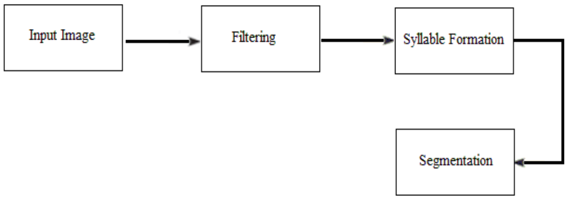


Figure 4. Block Diagram of the Proposed Solution

Figure 4 gives the overview of the proposed solution. A handwritten text document undergoes a two stage filtration process whose output is given to the syllable formation. In this stage all the noise is removed and a meaningful syllable component is formed. Then the joining of the components in a bottom-up approach is carried out to represent a line. The text line thus obtained is accurate and free from skew for image representing Kannada scripts.

Our method is designed to address the specific complexity of Kannada script. The broken units in the script need to be identified and then combined to form a meaningful syllable. The logic of merging involves computing the standard error of the mean (SE) that represents the standard deviation of the sample-mean's estimate of the components population mean for the script image. SE is usually estimated by taking the sample estimate of the components standard deviation divided by the square root of the number of components.

Here we are assuming statistical independence of the values in the sample.

$$SE = \sigma/\sqrt{N} \qquad (1)$$

Where
$\sigma =$ *standard deviation of the component size*
$N =$ *number of components in the image*

The formula for standard error (SE) is derived from the term about the variance of a sum of independent random variable [11].
Let C1, C2….CN are N independent characteristic values of the component defining the population of the script image. The image has a mean value µ and standard deviation represented by σ, then The Total Variance

$$T = (C1 + C2\ldots+ CN) = N * \sigma 2 \qquad (2)$$

The variance for every component represented as

$$C_v = \frac{1}{\sqrt{N^2}}\left(N - \sigma^2\right) \qquad (3)$$

Thus the standard error or in other words standard deviation for the

$$\text{Variance=} \quad \frac{\sigma}{\sqrt{N}} \qquad (4)$$

The standard error gives the variance from the standard deviation of the sampling distribution of the components in the script image. We take the complement of this value to identify the components that are split and can be combined

$$ST = (1 - SE) \qquad (5)$$

Where

ST = Sample Threshold
SE = Standard Error for the sample

Using the above Threshold value we compare to identify the list of split components. These split components are recursively processed to identify its nearest neighboring component and a decision is taken if the syllables can be merged. The outcome is list of syllable that form a meaningful characteristic set that can be distributed and combined to formulate a line.

Implementation

In this section an innovative method for line segmentation of Kannada handwritten document has been proposed. If the spacing for a given syllable is static and the spacing is known, it is possible to normalize the fitting error. The challenge as seen in Figure 6, is just not that the line spacing value varies but also the spacing between the syllables (vaththus) vary even in the single document. Therefore, line segmentation is estimated locally by using the method that accounts line spacing as well as the syllable spacing. The method is composed of two vital stages. In the first stage, component refining technique is used for formulating a base unit of Kannada specific component syllable. In the second stage, weighted bucketing algorithm is proposed for the segmentation of the document image text into lines.

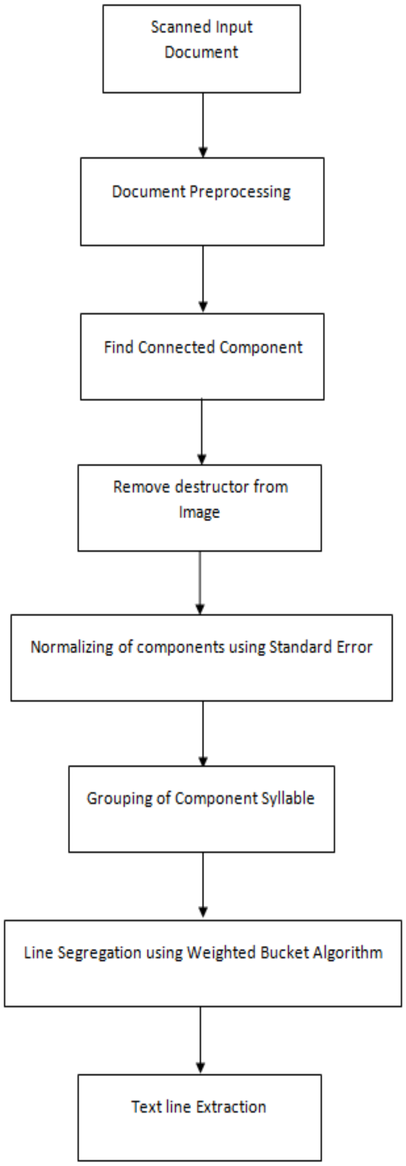
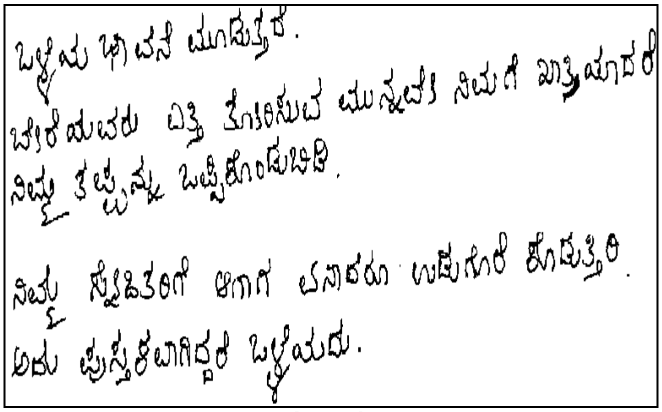

Scanned Input Document
Document Preprocessing
Find Connected Component
Remove destructor from Image
Normalizing of components using Standard Error
Grouping of Component Syllable
Line Segregation using Weighted Bucket Algorithm
Text line Extraction

Figure 5: Flow Diagram of the Implementation Logic

The functional block diagram of the proposed text-line segmentation algorithm is shown in Figure 5, which depicts different steps spread across the two stages clearly. The steps in Stage 1 are: (a) Document Pre-processing, (b) Removal of Noise from the Image, (c) Find Connected Components, (d) Estimation of Merge Syllable using Standard Error and (e) Grouping of Component Syllable. Once the lists of Syllable components are obtained the step (f) and (g) for Line Identification process are carried-out to segment the images into the corresponding lines. Each step is described in detail in the following sub-sections.



ಒಳ್ಳೆಯ ಭಾವನೆ ಮೂಡುತ್ತದೆ.
ಬೇರೆಯವರು ಎತ್ತಿ ತೋರಿಸುವ ಮುನ್ನವೇ ನಿಮಗೆ ಖಾತ್ರಿಯಾದರೆ
ನಿಮ್ಮ ತಪ್ಪನ್ನು ಒಪ್ಪಿಕೊಂಡುಬಿಡಿ.
ನಿಮ್ಮ ಸ್ನೇಹಿತರಿಗೆ ಆಗಾಗ ಏನಾದರೂ ಉಡುಗೊರೆ ಕೊಡುತ್ತಿರಿ.
ಅದು ಪುಸ್ತಕವಾಗಿದ್ದರೆ ಒಳ್ಳೆಯದು.

Figure 6: Kannada Handwritten Document

The input is typically a scanned image of the text document. In the pre-processing step the document image is converted into its binary equivalent using Otsu thresholding method. The binary image so obtained is then corrected for skewedness' that may have resulted as a consequence of the document having been improperly placed during scanning through application of some well-known techniques of Histogram analysis. The resultant document forms the first base for segmentation process.

The basic logic of text line formation technique starts with the segmentation of a document image into simple fragments. Thus in the next step the document image is bifurcated into its component. A two pass algorithm is applied to identify the connected components. For each of the components so formed, bounding box is defined and labelled. The list of the components is now given to the next step for refining.

Expunction of the components has to be done to remove any distracters in the image. These distracter are unwanted pixels or group of pixels that may have arisen due to some issues such as soiling, scanner limitations or may represent very small value such as "full stop". These do not add much value to the line identification logic, but may prove error prone to the algorithm. The elimination of distracters is also an important sequence. Expunction threshold can be formulated by considering each component as a weighted cluster. If P is the total weightage of the pixels in the image then it is distributed into several clusters, equivalent to a process that finds

$$\mathcal{W} = \{w_1, w_2, \dots, w_n\} \tag{6}$$

Where,

$$w_1 \cup w_2 \cup \dots \cup w_n = P \tag{7}$$

and $w_i \cap w_j = \emptyset (1 \le i < j \le n)$. Based on this observation, expunction threshold is formulated as 10% of the mean weight of the pixels

$$E_t = \left[\sum_{i=1-n} (w_i) \quad / \, n) \right] \quad * 10/100 \tag{8}$$

The connected components having values less than E_t are eliminated resulting in a refined list of standard components.

The core of the algorithm relents to normalization of components. It is evident from Figure 3 that the component shall also represent split characteristics or a part of the characters. These do not confine to a complete syllable. Thus a need arises for combining the components to form a meaningful syllable. To identify the potential components those are to be combined the measure of standard error (5) is adopted. As is evident, height is taken as a factor as the split characters are nearly half the average height of the component box.

$$MeanH_t = \left[\sum_{i=1-n} (h_i) \quad / \, n) \right] \tag{9}$$

Where
$h_i = Height\ of\ the\ component\ box$
$n = number\ of\ component\ box$

The equation to define the threshold (CH_t) is formulated as

$$CH_t = [(1 - SE_t) * MeanH_t] \tag{10}$$

Where

$SE_t = Standard\ Error\ forHeight\ of\ the\ component\ box$
$MeanH_t = average\ for\ Height\ of\ component\ box$

The standard components list is thus bifurcated to form:

1. New list of normalize components that are associated to be a potential candidature for merging.
2. List of residual components that do not need merging.

The algorithm for grouping of the components to form a single meaningful syllable component is a recursive activity as depicted in Figure 8. The normalized list is sorted in the ascending order of the centroid x-axis value. The component is chosen one by one and analyzed for the nearest component to be merged by iterating through the reminder of the list.

Criteria for merging the two components can be defined as follows

1. Two components to be merged must have sufficient horizontal overlap.
2. The difference between the vertical co-ordinates between the two component seed must be small.

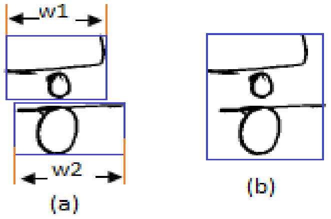


Figure 7: Block Diagram indicating Horizontal Overlap

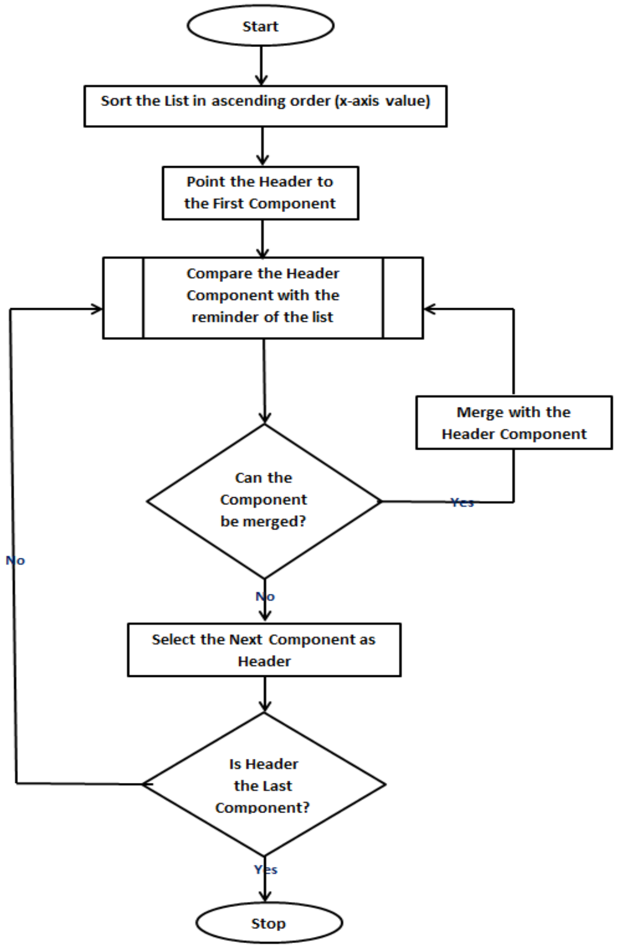

Start
Sort the List in ascending order (x-axis value)
Point the Header to the First Component
Compare the Header Component with the reminder of the list
Merge with the Header Component
Can the Component be merged?
Yes
No
No
Select the Next Component as Header
Is Header the Last Component?
Yes
Stop

Figure 8: Flow for Component Grouping

Here we define the Horizontal overlapping are as

$$\text{Overlapping Area} = \frac{\text{width}(w_1 \cap w_2)}{\min\{w_1, w_2\}} \qquad (11)$$

Where

w_1 *and* w_2 *are width of the two successive component box as shown in Figure.7 (a)*

Now the bounding boxes with height close to half the mean height are combined into one composite box as shown in Figure 7(b). Their widths are refined. The centroids of all boundary boxes are correspondingly redefined.

The reconstituted list is integrated with the residual component list to result in one single composite list ready for line segmentation.

Weighted bucketing algorithm is applied to segment the component syllables into different lines. The idea of segregation using bucketing logic is to divide the interval into n subintervals, iterating across the syllables to distribute the elements into buckets and then sort the elements in the bucket. Thus each bucket shall represent a line. Bucket algorithm works as follows:

1. Initially create an array of empty "bucket".
2. Select an empty bucket.
3. Select the component syllable with highest weight and link to the bucket.
4. Go over the original list of component syllables, putting each syllable in the bucket that falls within the range.
5. Repeat steps 2-3 till all the component syllables are segregated into buckets
6. Delete any empty buckets left.
7. Sort each bucket to arrange the syllable from left to right.

To derive the number of buckets we compute the mean of the heights of all the syllable components and divide the image height by the mean value. The equation is formulated as

$$Number\ of\ Bucket = \left(Ht_{image} / \ MeanH_t \right) \qquad (12)$$

Where

$Ht_{image} = Height\ of\ the\ Document\ Image$

$MeanH_t = \ average\ Height\ of\ component\ box$

Aggregation of all the components to form a line is based on a simple principle of the immediate neighboring component boxes having sufficient overlap distance along y-axis.

Now consider the Kannada handwritten text document having derived the connecting components as shown in Figure 9 (a). The goal is to segregate and arrange the boxes to form a line of text as shown in Figure 9 (b). To illustrate the line segregation procedure consider the Box representation [a, b….l] of the components as indicated in Figure 10.

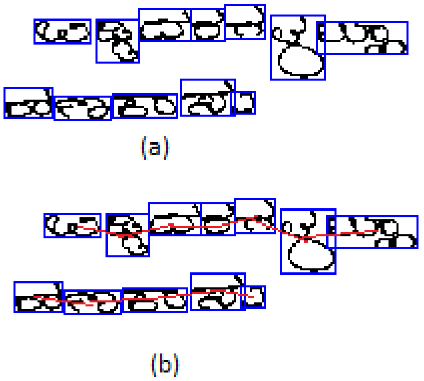


Figure 9: Kannada Handwritten script (a)Connected Components (b) after line segment formation

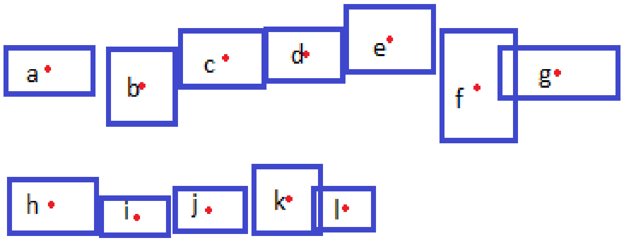


Figure 10: Box illustration of text connecting components

Line segregation algorithm works as follows

1. Select the Box with the highest weighted value and mark it as root. *[e]*
2. Split the list into left component and right component corresponding to the boxes that appear to the left or right of the root.
   Right Component = [a, b, c, d, h, i, j, k] and Left Component = [f, g, l]
3. Select the Right Component for Source List.
4. Choose the most recent box or root as the comparator.
5. Traverse the Source List to identify the immediate neighbor using the criteria of max overlap on Y-axis and minimum horizontal distance and add to the new list.
   List [d]
6. Repeat step 3 and 4 till no new neighbor can be found.
   List [d,c,b,a]
7. Select the Left Component for Source List and root as comparator.
8. Repeat step 4 to 6
   List [d,c,b,a,f,g]
9. Add the root box to the List
   List [d,c,b,a,f,g,e]

The residual list will have the boxes that form a new line. So repeat the algorithm from step 1 to 9 to get a segregated list of all the lines. Figure 11 (a) indicate the buckets.

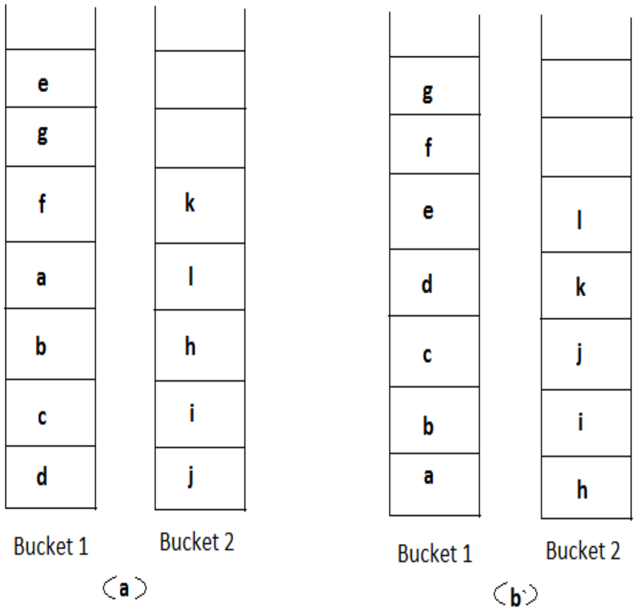


Figure 11: Bucket illustration of text connecting components (a) components segregated into buckets forming line (b)Components sorted to form a meaningful line.

To finalize the line segmentation process the components in the bucket are sorted in ascending order for their x-axis value. The outcome is a meaningful line as depicted in Figure 11 (b). To have a better view, the centroid of all the components can be joined as represented in Figure 12

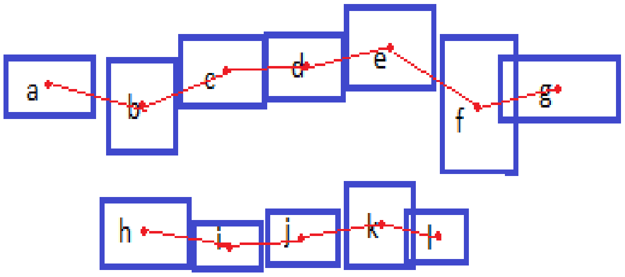


Figure 12: Illustration of text line segmentation

Sporadic lines might have been generated as shown in Figure 13 (a). These lines result as two or more modifier syllables (vaththus) can combine to project a separate line. Such lines are characterized by having less number of components and are usually placed at quiet a huge distance from each other. The fine tuning step is to identify such lines and merge the components with the predecessor line as depicted in Figure 13 (b). This completes an accurate line segmentation of Kannada Handwritten Text Documents.

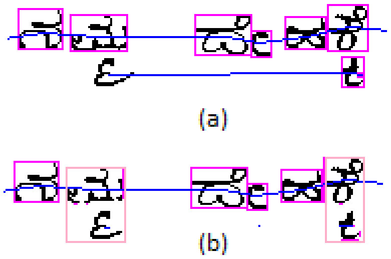


Figure 13: Fine Tuning process of line segmentation: (a) sporadic line as a consequence of vatthus, (b) final result of line segmentation

Result and Discussion

In order to assess this method, experiment is conducted on scanned handwritten Kannada text documents. Figure 14 shows the input document that has been subjected to the pre-processing step of binarization. Bifurcating into connected components being the next basic step is shown in Figure 15. The next core process is identifying the components that are need to be merged to form a syllable as shown in Figure 16.These are represented in green box. Then syllable is framed by merging the characters as shown Figure 17. The text lines are detected and segmented by application of weighted bucketing algorithm and each detected line is indicated by a line joining the centroid values, as illustrated in Figure 18. Further, the refinement of segmented output is as shown in Figure 19.

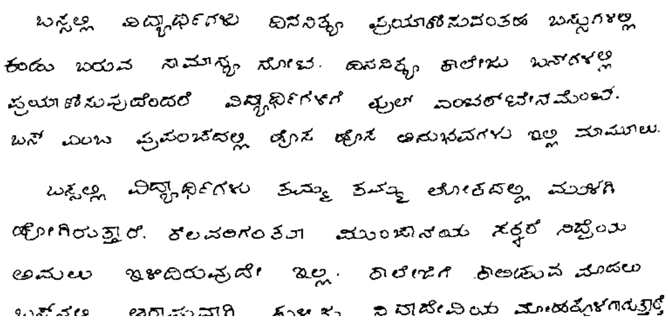


Figure 14: Kannada Handwritten Document after pre-processing

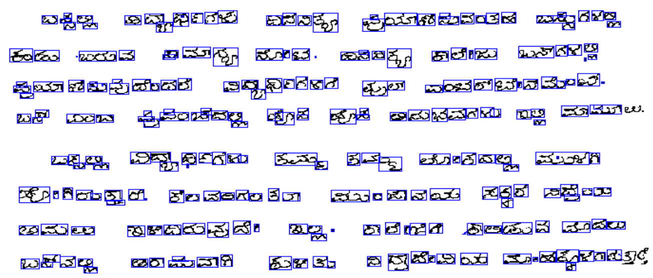


Figure 15: Componentization of document image

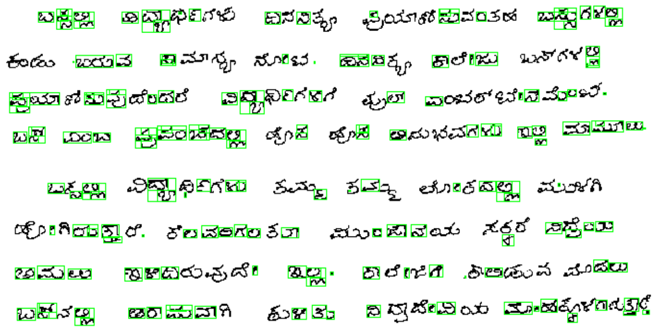


Figure 16: Potential Syllable identified for Mergining

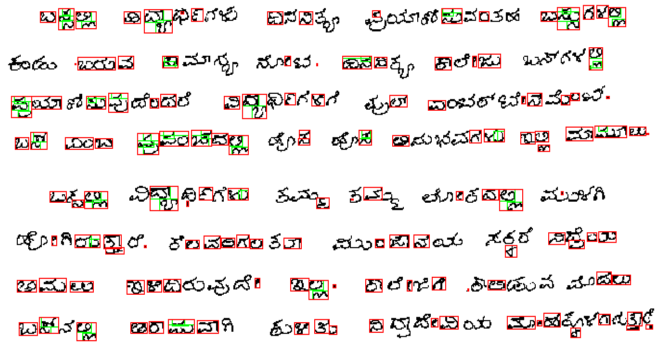


Figure 17: Merging of Broken Syllable Components

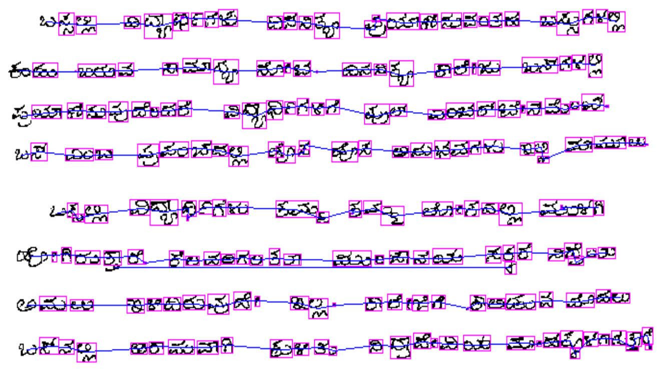


Figure 18: Line Segmentation using Weighted Bucket algorithm

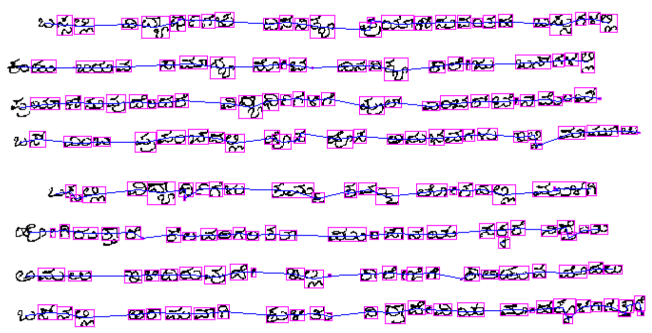


Figure 19: Fime tuning of Line Segmentation

To summarize the performance of the proposed algorithm, experiments have been conducted on 80 Kannada handwritten scripts. The result form the experimentation is as follows:

TABLE I. PERFORMANCE EVALUATION OF THE PROPOSED SYSTEM

Number of Documents	80
Average Number of Lines in Document	32
Total number of Lines Analysed	2560
Number of Lines Segmented	2512
Accuracy of Segmentation	98.12%

Conclusion

The scope of the work presented here reflects a robust schema for segmenting of text lines for a Kannada handwritten document. Analyzing the existing methods of line segmentation, it has been noted that the accuracy and performance is muted due to the split character nature of Kannada syllables. The key strength of our novel solution for text line segmentation for handwritten documents is:

a) Eliminate line segmentation errors as a consequence of broken character syllable.

b) Handle special characters (vaththus) components without any fuss.

c) Implicitly the complexity related to skewness of the line is also addressed

The logic considers both the standard error to determine the factor for fitting the component of each text line and the distances between text lines. This method is proved to be very efficient compared to the traditional methods.

References

[1] Strand, L, Malmberg, F and Svensson, S.K. " Minimal Cost-Path for Path-Based Distances", Image and Signal Processing and Analysis, pp. 379-384, 2007.

[2] Xiaojun Du, Wumo Pan and Tien. D. Bui, "Text Line Segmentation in Handwritten Documents Using Mumford-Shah Mode"l, Pattern Recognition, Vol.42(12), pp.3136-3145, 2009.

[3] Louloudis. G., Gatos. B. & Halatsis. C., "Text line detection in handwritten documents", (2008)

[4] Shi, Z.; Seltur, S. & Govindaraju, V. (2009). A Steerable Directional Local Profile Technique.

[5] U. Pal, B.B. Chaudhuri. (2004): Indian script character recognition: a survey, Pattern Recognition, 37,1887 – 1899.

[6] B. Anuradhaand, Arun Agarwal and C. Raghavendra Rao, "An Overview of OCR Research in Indian Scripts", International Journal of Computer Science and Engineering System,(IJCSES), Vol.2, No.2, pp. 141-153, 2008.

[7] Richard G. Cassey and Eric Lecolinet, " A Survey of methods and strategies in character segmentation", IEEE Transactions on Pattern analysis and Machine Intelligence, Vol 18, No.7, 1996.

[8] C. V Lakshmi and C. Patvardhan, "An optical character recognition system for printed Telugu text, Pattern Analysis & Applications, Vol.7, pp.190-204, 2004.

[9] Agarwal and David Doermann, "A Dynamic Page Segmentation approach based on Voronoi and Docstrum features",10th International Conference, ICDAR, 2009.

[10] B.M. Sagar, G. Shoba and P. Ramakanth Kumar, "Character Segmentation algorithms for kannada optical character Recognition", Proceedings of the International Conference on Wavelet Analysis and Pattern Recognition, 2008.

[11] T.P. Hutchinson, "Essentials of statistical methods in 41 pages", Rumsby Scientific Publishing., 1993.

[12] Hossein Kardan Moghaddam "The Horizontal Segmentation of Lines in Chinese Handwritten Texts Based on the Intervals (Distances) in Fuzzy Triangles ", Journal of Basic and Applied Scientific Research, Vol 3,pp 165-172, 2013.

[13] Khalil Ibrahim Alsaif and Montaha Tariq Alsarraj, " New Technique For Skew Angle Detection Of Text In Image Document" , International Journal of Information Technology and Business Management , Vol.16 No.1, pp.102-110, 2013.

Biographies

Sunanda Dixit obtained her Bachelor's degree in Electrical and Electronics Engineering from Visvesvaraya Technological University and M.Tech degree in Computer Science and Engineering from VTU, India. Since 2010 she has been a Ph.D. student at Visvesvaraya Technological University, India. Currently she is working as an Assistant Professor in Information Science and Engineering department, Dayananda Sagar College of Engineering, Bangalore India. She has published more than 8 research papers in the refereed international journals and presented contributed research papers in refereed international and national conferences. Her research interest includes document image processing , pattern recognition and pattern classification. She is a life member of Indian Society for Technical Education.

Dr. H.N. Suresh received his BE (E&C) from P.E.S College of Engineering, Mysore University, Karnataka, India, in the year 1989 and completed his M.Tech (Bio Medical Instrumentation) from SJCE Mysore affiliated to University of Mysore., in the year of 1996 and since then he is actively involved in teaching and research and has Twenty six years of experience in teaching. He obtained his PhD (ECE) from Anna university of Technology. He worked at various capacities in affiliated University Engineering Colleges. For Visveswaraya Technical University and Bangalore University he worked as a Chairman for Board of Examiners, and member of Board of Studies etc. At present he is working as Professor in Bangalore Institute of Technology, Bangalore Affiliated to Visveswaraya Technical University. He has good exposure in the field of signal processing, Wavelet Transforms, Neural Networks, Pattern recognition, Bio Medical Signal Processing, Netwoking and Adaptive Neural network systems. He has published more than 30 research papers in the refereed international journals and presented contributed research papers in refereed international and national conferences. He is a member of IEEE, Bio Medical Society of India, ISTE,IMAPS & Fellow member of IETE.