Indonesian Journal of Electrical Engineering and Computer Science
Vol. 31, No. 1, July 2023, pp. 521~530
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v31.i1.pp521-530

Segmentation approach for offline handwritten Kannada scripts

Krupashankari S Sandyal[1], Kiran Y Chandrappa[2]
[1]Department of Information Science and Engineering, Dayananda Sagar College of Engineering, Visvesvaraya Technological University, Bengaluru, India
[2]Department of Information Science and Engineering, Global Academy of Technology, Visvesvaraya Technological University, Bengaluru, India

Article Info

Article history:

Received Nov 16, 2022
Revised Mar 26, 2023
Accepted Mar 27, 2023

Keywords:

Feature extraction
Hough transform
Pre-processing
Recognition system
Segmentation

ABSTRACT

India has more than 1,600 official languages, making it a multilingual country. Kannada, one of the major languages, originated in the state of Karnataka and is currently ranked 33rd among the accents that are most often spoken throughout the world. However, the survey shows that much more effort is needed to create a complete handwritten identification system. Segmentation is one of the crucial steps in a handwriting identification system that extracts significant objects from an image. The feature extraction and classification phases of handwritten text recognition will be more successful if the segmentation approaches selected are efficient. In the proposed system, segmentation was accomplished using bounding box and contour tracing methods. The result got is delivered to the next step of handwritten identification system. An average accuracy of 92.6% is worked out for line segmentation and word segmentation.

Corresponding Author:

Krupashankari S Sandyal
Department of Information Science and Engineering, Dayananda Sagar College of Engineering
Visvesvaraya Technological University
Bengaluru, Karnataka, 560078 India
Email: krupasandyal@gmail.com

1. INTRODUCTION

A computer system that can automatically decipher text from images is what text recognition in images aims to achieve. The abundance of beneficial automatic indexing or information retrieval applications for text detection and recognition in general, such as document indexing, content-based image retrieval, and licence plate recognition, further increases the potential for more sophisticated handwritten recognition systems. Handwritten text recognition system eliminates the need for manual retyping of critical documents when putting them into electronic databases. The processes of image pre-processing, segmentation, feature extraction, and character recognition are prevalent in all handwritten recognition systems. The effectiveness of the preceding stages has a significant adverse effect on each of these stages' outcomes. The different writing styles and text distortion in handwritten papers make it challenging to segment and filter digitized documents based on a query. Because of the extensive database and structural complicatedness, the growth of text identification system for few of the Indian languages like Kannada and Telugu is anticipated a laborious process [1]. These challenges are further amplified by the possibility of character overlap in some instances. Despite repeated attempts, creating a higher precision recognition framework for all Indian languages is incredibly hard. The paper's main body is organised as follows: the previous research is briefly summarized in part 2, and details on the suggested technique are provided in section 3. The tests and results are detailed in section 4 and section 5 offers the conclusion and futuristic recommendations.

2. LITERATURE SURVEY

Literature survey concentrates on the specified subject, critical analysis and relationship between various works done by different authors. Table 1 outlines the research that has been done so far in the area of handwritten text detection by numerous researchers. There have also been illustrations of the research's preliminary findings. The literature overview provides a comprehensive analysis of the numerous text line segmentation-related challenges, which aids researchers in comprehending and advancing their work in this area.

Table 1. Illustrations of the research's preliminary findings

Authors	Techniques/algorithm used for feature extraction/segmenation/classification	Accuracy	Publication year
Fernandes *et al.* [2]	Tesseract tool, convolution neural network (CNN).	Tesseract too l-86%, CNN-87%.	2019
Sushma *et al.* [3]	Pre-processing methods-height normalization, bounding box extraction and binarization, Feature extraction methods-SIFT and SURF.	32.5% to 75% range for every word.	2016
Gowda and Kanchana [4]	Feature extraction-curviness of the characters using CNN, Edge-based segmentation algorithm, Categorization-SVM, KNN and Random Forest algorithms.	Accuracy for random forest-95%, SVM-96% and KNN-92%.	2022
Kohli and Kumar [5]	To segregate touching components, have used the segmentation facilitation feature to pinpoint the junction path.	Segmentation accuracy 89.9%.	2021
Kaur *et al.* [6]	The techniques of detecting header lines, base lines, and contours.	Accuracy depends upon the segmentation technique.	2010
Choudhary *et al.* [7]	Vertical segmentation algorithm.	Segmentation accuracy 83.5%.	2013
Mello and Lacerda [8]	Segmentation via feature point selection, skeletonization and clustering using self-organizing maps.	Recognition rate 65.79%.	2013
Mahto *et al.* [9]	Scheme for merging horizontal and vertical projection feature selection.	Recognition accuracy 98.06%.	2015
Ramappa and Krishnamurthy [10]	Bounding box technique, Hough transform and contour detection.	An average segmentation rate of 91% and 70% for lines and words is obtained.	2012
Thungamani and Kumar [11]	Horizontal projection profile, vertical projection profile.	Accuracy depends upon the segmentation algorithm.	2012
Vishwanath *et al.* [12]	Composite feature vector retrieval utilizing gradient-based feature descriptors, edge density filter and adaptive projection profiling for segmentation; Classification-SVM.	The positive predictive value for the texts of Bengali, Telugu, and Kannda were 74.7240%, 76.9728%, and 79.9518%, respectively.	2023
Obaidullah *et al.* [13]	Bounding box, chain-code direction histogram, radon transform, categorization using MLP and random forest.	For word-level accuracy for Tri script recognition, the statistics are 96.76%, 95.83%, 99.03%, and 96.60%.	2019
Rao *et al.* [14]	Extended nonlinear kernel residual network.	Test accuracy 97.72%.	2018
Kavitha and Srimathi [15]	Max pooling, Softmax classifier, CNN model.	Training accuracy of 95.16%.	2022
Ghosh *et al.* [16]	Max-voting, probabilistic voting used for feature extraction, wavelet transformation, CNN for training.	Accuracy of 95.04%.	2019
Kumari and Babu [17]	Segmentation is based on mathematical morphology, CNN.	Accuracy of 98.7%.	2021
Muppalaneni [18]	CNN are incorporated into methodologies for deep learning and frameworks for machine learning.	As high as 79.61% for testing precision and 96.13% for precision.	2020
Thakral and Kumar [19]	Cluster detection technique, horizontal projection profile.	Contiguous characters segmented with 88% accuracy and touching, conjunct characters with 95% accuracy.	2014

As a result of the drawbacks described in the works, there is room for additional study and advancement in the pre-processing and segmentation of handwritten documents. This inspired us to create a system that effectively segments and recognizes handwritten Kannada documents.

2.1. Kannada script

Kannada is the primary language in Karnataka. From the Brahmi-descended Kadamba and Chalukya scripts came the development of the aksharas used in Kannada. Figure 1 illustrates the fundamental alphabet of the Kannada language, which consists of 16 vowels and 34 consonants [20], [21]. There is a vowel sign (modifier) for each vowel and a fundamental form for each consonant (primitive). A basic consonant can be

combined with a vowel sign to create a set of 16 composite consonant-vowel (CV) symbols known as the "gunithakshara." All 34 consonants in Kannada have a short/half version known as "Vatthus," sometimes known as half consonants or subscripts. A conjunct-consonant letter can be created by placing any half-consonant as a subscript character another consonant or a CV character. This work examines the pre-processing and segmentation of documents written by hand in Kannada.

Figure 1. Kannada vowel and consonants samples

3. PROPOSED METHOD

In this paper a better method for pre-processing and segmenting handwritten Kannada documents has been suggested. Traditionally, the feature extraction and recognition stage in all character recognition systems utilizes the result of the segmentation procedure as its input. When a sample is improperly segmented, the recognition system is unable to identify it. However, the segmentation phase is not informed of this information. We have made an effort to minimize this gap between the processes of segmentation and recognition in our suggested strategy. The diagram shown in Figure 2 explores the stages used in the recognition system. To generate effective feature template from the input image at training stage consists of pre-processing, segmentation, relevant feature extraction and storing feature template in the knowledge base for classification.

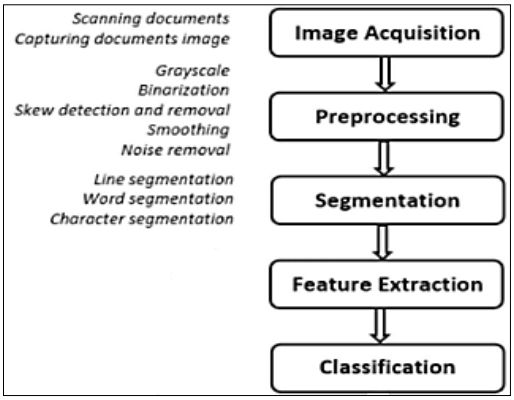

Figure 2. Flow chart

3.1. Data collection

Kannada handwritten dataset has been created by different age groups of people, it includes elements like diverse alignments, writing styles, and character lengths, as well as aspects like pen quality, paper quality, ink colour and others that make the handwritten Kannada text more complicated. Every image was scanned at a resolution of 300 dpi. 200 of these documents have been taken into consideration for the experiment. As a result, handwritten text faces a greater hurdle than printed text.

3.2. Pre-processing

We used the adaptive binarization strategy to finish the binarization stage of the pre-processing procedure as shown in Figure 3. Binarization is a process that turns a multi-tonal image into a bi-tonal one. It is customary to map the text pixels in the foreground of document images to black and the background of the image to white. To detect and fix skew, we used the hough transform, which converts an image from polar to Cartesian coordinates.

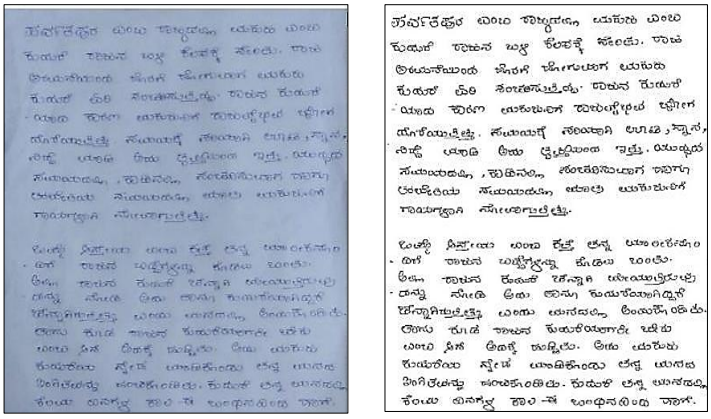

Figure 3. Results for adaptive thresholding

3.2.1. Skew detection and correction: Hough transform

The hough transform [22], initially developed to find patterns in an image, has indeed been refined to find curves in both two dimensions and three dimensions. By employing the subsequent processes, an image is converted into the hough space for line detection:

– Step 1. The expression for a line in Cartesian form is:

$$y = mx + b$$

where, m is the line's gradient or slope (rise/run) and b is the y-intercept.

We are looking for as many lines in image space that connect a set of edge points or a binary image signifying edge. Suppose we have two edge locations (x1, y1) and (x2, y2). We figure out the corresponding b values for every edge point at different gradient intensities (m=-0.5, 1.0, and 1.5). Figures 4 and 5 illustrate the image and parameter space in detail. This shared point (m, b) symbolises the line in the image space. Unfortunately, when the line is vertical, the slope, m, is unquantifiable. We employ the hough space, a different parameter space, to get around this.

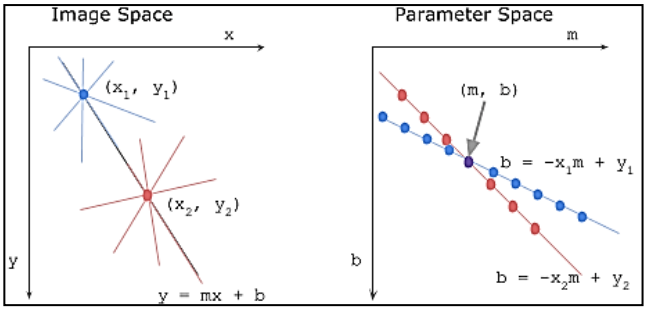

Figure 4. Image space and parameter space

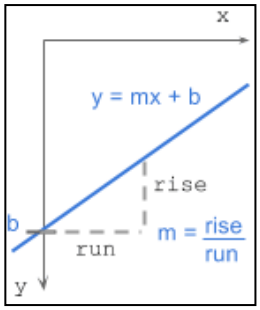

Figure 5. Straight line

– Step 2. Angle-distance parameter space in the polar coordinate system is:

$$\rho = x \cos\theta + y \sin\theta$$

where, the spacing between the source and the line is expressed as rho. [-max_dist to max_dist]. The image's diagonal length is max_dist. θ is the angle between the origin and the line [-90° to 90°].

– Step 3. Hough transform space

By computing ρ with a point at each angle between -90° and 90°, the image is turned into the hough space. Peaks (ρ, θ) in the hough transform space are where the curves produced by collinear regions in the image space intersect. In image space, a line will earn more "votes" if more curves overlap at a given point. Figure 6 shows the results for hough lines detected. The characteristics of the input image's most significant lines are shown by local maxima in the accumulator. The easiest way to find peaks is to use a threshold or a relative threshold. Figure 7 shows the results for skew corrected image after applying hough method.

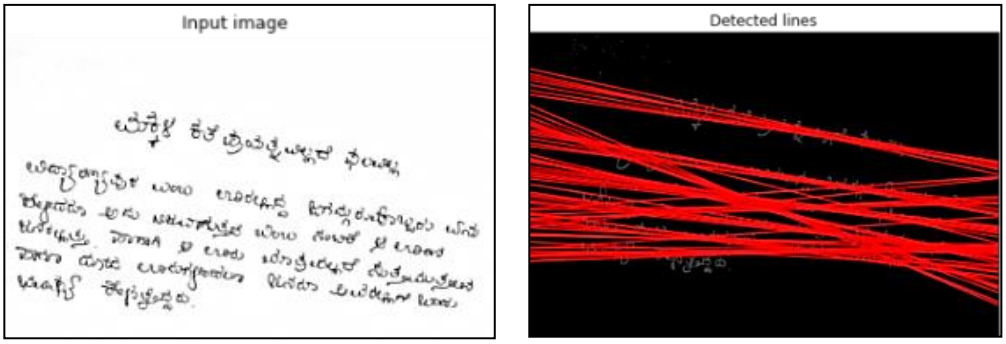

Figure 6. Hough lines detected

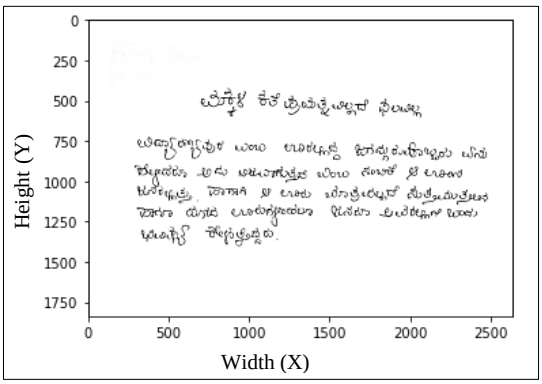

Figure 7. Skew corrected image using hough transform

3.2.2. Segmentation: method

Segmentation consists of three steps: dilatation, contour tracing and bounding rectangles.

– Dilation: dilation enlarges the boundaries of objects in an image by pixels. The number of times pixels are added to or deleted from the image's objects are determined by the dimension and shape of the structuring element used to process it.

– Contour tracing: the boundary of an object in a picture is known as a contour. To identify or classify things, contours are represented in a variety of ways.

Bounding rectangles: It refers to the border's coordinates that enclose an image. The x and y coordinates of the rectangle's upper-left corner and those of its lower-right corner, which serve as a point of reference for word detection, are used by the bounding box algorithm to generate a fictitious rectangle. The algorithm 1 for contour tracing is used to identify word boundary pixels. The contour is the line that connects all the identically intense points along an image's edge. It is carried out on a digital image of a word to extract details about its shape that will be utilized as characteristics in categorization. The under-segmentation issue brought on by characters overlapping can be resolved by contour-based approaches since they give a clear description of the shape of the characters. Additionally, since the baselines don't need to be adjusted numerous times, it lowers the errors made when extracting baselines.

To facilitate image processing, boundary detection is performed to locate the edges in an image. There are many edge identification techniques available, here we have used sobel edge detection method. To find the midway where we can make a threshold and extract the peaks regions, we can map horizontal projections profile (HPP) of an image. The array of the total number of rows in a two-dimensional (2D) frame is called the HPP. Figure 8 shows HPP for sobel edge. Sobel edge detection is where it creates an image emphasising edges as shown in Figure 9.

Algorithm 1. Contour tracing

```
Step 1.  Read image as grayscale
Step 2.  convert to grayscale
Step 3.  threshold=THRESH_BINARY
Step 4.  set kernel size
             apply morphology
             MORPH_CLOSE and MORPH_ERODE
Step 5.  get largest contour
             contours=find Contours
             area_threshold=0
             for c in contours:
            area=contourArea(c)
            if area>area_threshold:
          area_threshold=area
          big_contour=c
Step 6.  get bounding box
             x, y, w, h=boundingRect(big_contour)
Step 7.  draw filled contour on black background
             mask=zeros_like(gray)
             merge mask
            drawcontours
Step 8.  apply mask to input
             result1=img.copy()
             result1=bitwise_and(result1, mask)
Step 9.  crop result
             result=result1[y:y+h, x:x+w]
Step 10. view result
```

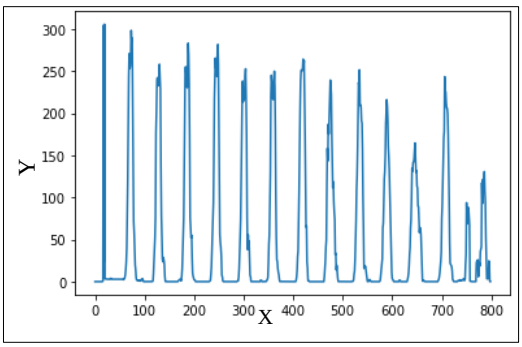

Figure 8. HPP of sobel image in 2D

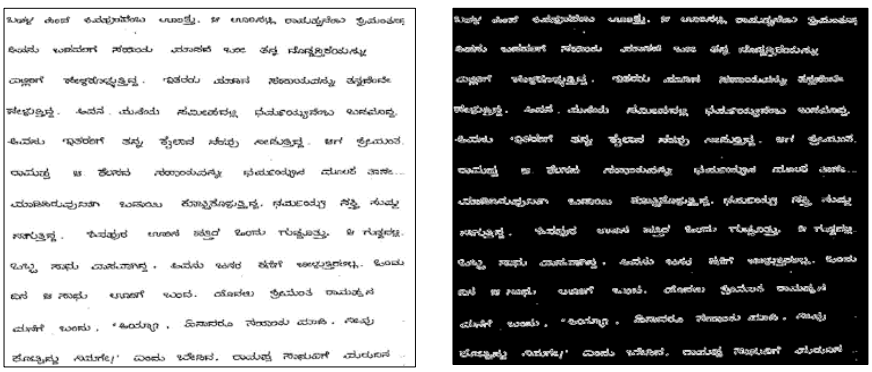

Figure 9. Sobel edge detection

In word segmentation, a bounding box is used to help the computer identify the word's location inside the image. The bounding box encloses the word's picture and indicates where it is in space. The bounding box strategy is used whenever a character is spotted in order to segregate the character by applying a bounding box. The final version of character segmentation is a clearly differentiated character. The segmentation algorithm 2 displayed presents in detail an enhanced character segmentation that improves the detection efficiency of recognition system.

Algorithm 2. Segmentation

```
Input: Kannada Handwritten scanned image of size mXn
Output: Set of segregated characters
step 1. Pre-processing
            Convert an RGB-formatted image to a monochrome version using an adaptive
            thresholding technique to get enhanced image.
step 2. Skew Detection and Correction
           Detect the skew angle and correct using Hough transform technique.
step 3. Segmentation
           Apply morphological operation on image.
step 4. Find Contours on dilated image
step 5. Draw lines on the boundary using values obtained from the contours method.
           Bounding Rectangle is used for drawing bounding box
step 6. Repeat step 5 for each line in the page
step 7. Change the kernel width for word segmentation
            and repeat the same for character segmentation.
```

4. EXPERIMENTAL RESULTS AND DISCUSSION

Handwritten character recognition is a contentious issue in optical character recognition applications and pattern categorization. Figure 10 depicts a decent number of hough lines connecting our words. The hough line method also gives us the angle made by the line with the origin as shown in Figure 11.

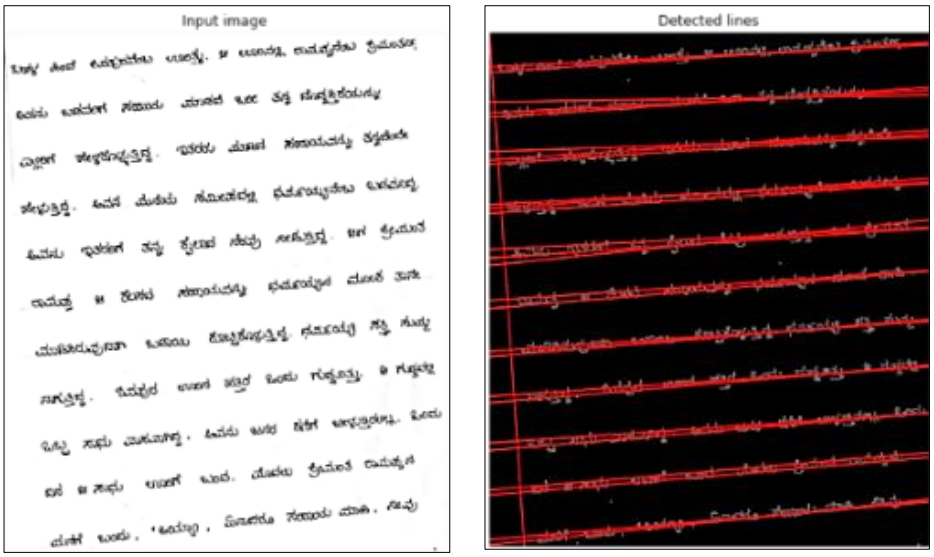

Figure 10. Hough lines drawn on skewed image

Result achieved for line segmentation phase is displayed in Figure 12. Figures 13-15 exhibit instances of line segmented images, word segmented images and character fragmented images respectively. The proposed method yields an average segmentation accuracy of 92.6%. However, because there are more segmented characters than the total number of characters, accuracy cannot be calculated at the character level. This is due to the fact that the Kannada script consists of consonant modifiers that are joined with one of the characters to create compound characters, which are used very frequently in this language. Compound characters are challenging to segment and should be approached from a different angle.

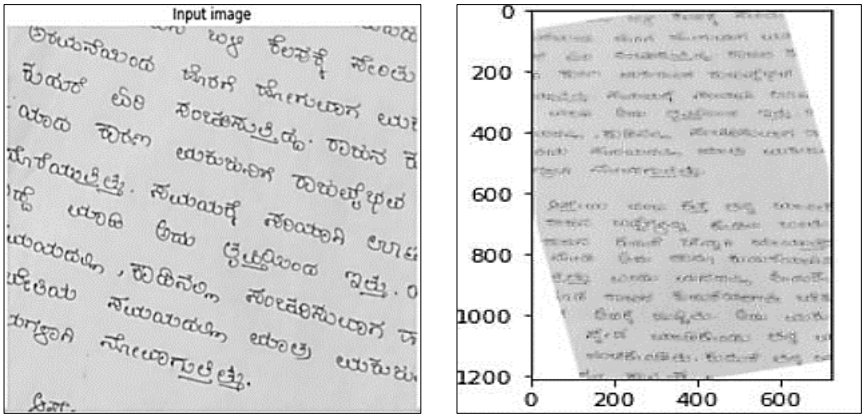

Figure 11. Results for skew angle detection

Figure 12. Result achieved for line segmentation phase

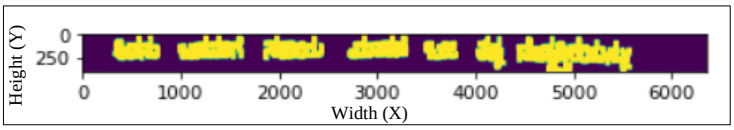

Figure 13. Accuracy achieved in line segmentation

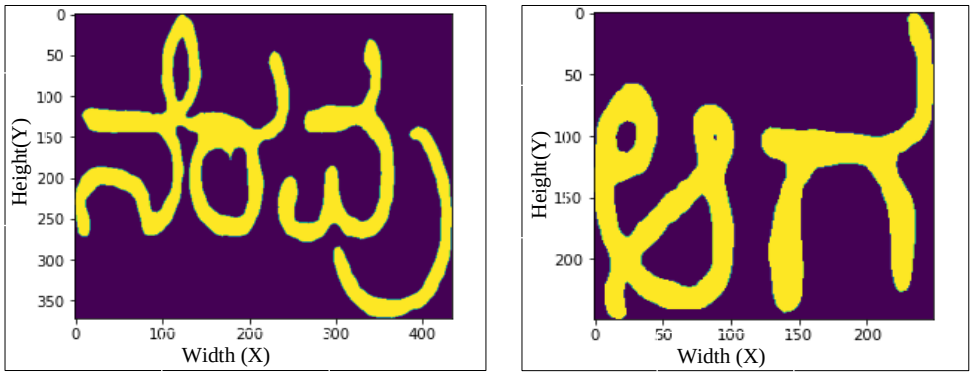

Figure 14. Word segmentation results

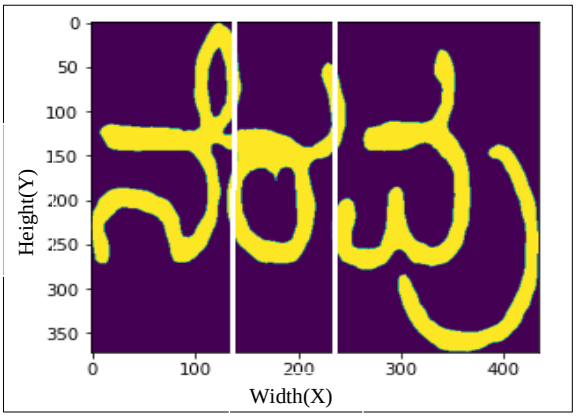

Figure 15. Character segmentation

The proposed method is contrasted with the existing methods in Table 2. For a dataset of 50, in method 1 three level segmentation is taken into account. Only the line and word segmentation are the focus of method 2 and 3. In developing the proposed approach, we took into account a considerably larger data set of 200 and created a three-level segmentation that includes line, word, and character segmentation.

Table 2. Assessment of the proposed method with the existing strategies

Author	Segmentation method	Size of dataset	Segmentation rate
Saleem Pasha, *et. al.*, [23] method 1	Modified projection profile, connected component.	50	97.5%
Alireza Alaei, *et. al.*, [24] method 2	Potential piece-wise separation line approach	204	94.98%
Mamatha *et. al.*, [25] method 3	Morphological operations, projection profile	100	94.5%
Proposed	Contour tracing and bounding box	200	92.6%

5. CONCLUSION

One of the crucial stages of a manually written recognition system is segmentation. Several segments of the pre-processed image are created. This study tries to partition text into three levels: lines, words, and characters, using appropriate pre-processing approaches. Dilatation, contour tracing and bounding rectangles are applied in segmentation to identify well-separated and overlapping lines. Word and line segmentation both obtain an average accuracy of 92.6%. Due to the frequent occurrence of compound characters, complete precision cannot be obtained at the character level. Such complex characters can be difficult to segment, thus it is important to approach the problem from a different perspective.

REFERENCES

[1] K. S. S. Kumar, A. M. Namboodiri, and C. V. Jawahar, "Learning segmentation of documents with complex scripts," in *Computer Vision, Graphics and Image Processing: 5th Indian Conference, ICVGIP 2006*, 2006, pp. 749–760, doi: 10.1007/11949619_67.

[2] R. Fernandes and A. P. Rodrigues, "Kannada handwritten script recognition using machine learning techniques," in *2019 IEEE International Conference on Distributed Computing, VLSI, Electrical Circuits and Robotics, DISCOVER 2019 - Proceedings*, Aug. 2019, pp. 1–6, doi: 10.1109/DISCOVER47552.2019.9008097.

[3] A. Sushma and G. S. Veena, "Kannada handwritten word conversion to electronic textual format using HMM model," in *2016 International Conference on Computation System and Information Technology for Sustainable Solutions, CSITSS 2016*, Oct. 2016, pp. 330–335, doi: 10.1109/CSITSS.2016.7779380.

[4] D. K. Gowda and V. Kanchana, "Kannada handwritten character recognition and classification through OCR using hybrid machine learning techniques," in *IEEE International Conference on Data Science and Information System, ICDSIS 2022*, Jul. 2022, pp. 1–6, doi: 10.1109/ICDSIS55133.2022.9915906.

[5] M. Kohli and S. Kumar, "Segmentation of handwritten words into characters," *Multimedia Tools and Applications*, vol. 80, no. 14, pp. 22121–22133, Jun. 2021, doi: 10.1007/s11042-021-10638-0.

[6] N. K. Garg, L. Kaur, and M. K. Jindal, "A new method for line segmentation of handwritten Hindi text," in *ITNG2010 - 7th International Conference on Information Technology: New Generations*, 2010, pp. 392–397, doi: 10.1109/ITNG.2010.89.

[7] A. Choudhary, R. Rishi, and S. Ahlawat, "New character segmentation approach for off-line cursive handwritten words," *Procedia Computer Science*, vol. 17, pp. 88–95, 2013, doi: 10.1016/j.procs.2013.05.013.

[8] E. B. Lacerda and C. A. B. Mello, "Segmentation of connected handwritten digits using self-organizing maps," *Expert Systems with Applications*, vol. 40, no. 15, pp. 5867–5877, Nov. 2013, doi: 10.1016/j.eswa.2013.05.006.

[9] M. K. Mahto, K. Bhatia, and R. K. Sharma, "Combined horizontal and vertical projection feature extraction technique for Gurmukhi handwritten character recognition," in *Conference Proceeding - 2015 International Conference on Advances in Computer Engineering and Applications, ICACEA 2015*, Mar. 2015, pp. 59–65, doi: 10.1109/ICACEA.2015.7164646.

[10] M. H. Ramappa and S. Krishnamurthy, "Skew detection, correction and segmentation of Handwritten Kannada document," *International Journal of Advanced Science and Technology*, vol. 48, pp. 71–88, 2012.
[11] R. G. Casey and E. Lecolinet, "A survey of methods and strategies in character segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 7, pp. 690–706, 1996, doi: 10.1109/34.506792.
[12] N. V. Vishwanath, K. Manjunathachari, and K. S. Prasad, "Multi-lingual character segmentation and recognition based on adaptive projection profiles and composite feature vectors," *Multimedia Tools and Applications*, 2023, doi: 10.1007/s11042-023-14523-w.
[13] S. M. Obaidullah, K. C. Santosh, C. Halder, N. Das, and K. Roy, "Automatic Indic script identification from handwritten documents: page, block, line and word-level approach," *International Journal of Machine Learning and Cybernetics*, vol. 10, no. 1, pp. 87–106, Jan. 2019, doi: 10.1007/s13042-017-0702-8.
[14] Z. Rao *et al.*, "Research on a handwritten character recognition algorithm based on an extended nonlinear kernel residual network," *KSII Transactions on Internet and Information Systems*, vol. 12, no. 1, pp. 413–435, Jan. 2018, doi: 10.3837/tiis.2018.01.020.
[15] B. R. Kavitha and C. Srimathi, "Benchmarking on offline Handwritten Tamil Character Recognition using convolutional neural networks," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 4, pp. 1183–1190, Apr. 2022, doi: 10.1016/j.jksuci.2019.06.004.
[16] S. Ukil, S. Ghosh, S. M. Obaidullah, K. C. Santosh, K. Roy, and N. Das, "Improved word-level handwritten Indic script identification by integrating small convolutional neural networks," *Neural Computing and Applications*, vol. 32, no. 7, pp. 2829–2844, Apr. 2020, doi: 10.1007/s00521-019-04111-1.
[17] T. M. Kumari and A. V. Babu, "Recognition of offline hand written telugu script using deep learning," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, pp. 2456–3307, 2021, doi: 10.32628/IJSRCSEIT.
[18] N. B. Muppalaneni, "Handwritten telugu compound character prediction using convolutional neural network," in *International Conference on Emerging Trends in Information Technology and Engineering, ic-ETITE 2020*, Feb. 2020, pp. 1–4, doi: 10.1109/ic-ETITE47903.2020.349.
[19] B. Thakral and M. Kumar, "Devanagari handwritten text segmentation for overlapping and conjunct characters- A proficient technique," in *Proceedings - 2014 3rd International Conference on Reliability, Infocom Technologies and Optimization: Trends and Future Directions, ICRITO 2014*, Oct. 2015, pp. 1–4, doi: 10.1109/ICRITO.2014.7014746.
[20] R. S. Kunte and R. D. S. Samuel, "A simple and efficient optical character recognition system for basic symbols in printed Kannada text," *Sadhana - Academy Proceedings in Engineering Sciences*, vol. 32, no. 5, pp. 521–533, Oct. 2007, doi: 10.1007/s12046-007-0039-1.
[21] K. Indira and S. S. Selvi, "Kannada character recognition system a review," *2010, arXiv: 1001.5352*.
[22] R. C.Gonzalez, R. E.Woods, and S. L.Eddins, "Understanding digital image processing using MATLAB," *Indian Edition*, 2004.
[23] S. Pasha and M. C. Padma, "Segmentation of handwritten documents containing Kannada script," *International Journal of Computer Applications*, vol. 144, no. 12, pp. 1–6, Jun. 2016, doi: 10.5120/ijca2016910485.
[24] A. Alaei, P. Nagabhushan, and U. Pal, "A benchmark Kannada handwritten document dataset and its segmentation," in *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, Sep. 2011, pp. 141–145, doi: 10.1109/ICDAR.2011.37.
[25] H. R. Mamatha and K. Srikantamurthy, "Morphological operations and projection profiles based segmentation of handwritten Kannada document," *International Journal of Applied Information Systems*, vol. 4, no. 5, pp. 13–19, Oct. 2012, doi: 10.5120/ijais12-450704.

BIOGRAPHIES OF AUTHORS

Krupashankari S Sandyal assistant professor at Dayananda Sagar College of Engineering, Bengaluru, Karnataka, India. She holds a postgraduate degree in computer science and engineering from Dayananda Sagar College of Engineering, Bengaluru under Visvesvaraya Technological University. She has more than 8 years of academic experience which includes teaching and research experience. Her research areas of interest are image processing, machine learning, medical image analysis, and pattern recognition. She can be contacted at email: krupasandyal@gmail.com.

Dr. Kiran Y Chandrappa professor and head, in information science and engineering, Global Academy of Technology, Bengaluru, Karnataka, India. He obtained his post graduate degree in software engineering, from SJCE Mysore in the year 2003 and was awarded Ph.D. from Jain University, Bengaluru in the year 2015 in image processing and pattern recognition. He has more than 25 years of academic experience which includes teaching and research experience. He has published more than 50 research articles in refereed journals and also he is one of technical consultants for software company Aspire Business solutions, Bengaluru. He serves on the editorial board of reputed journals i.e., International Journal of Decision Support System Technology (IGI Global USA), CRC Press-A Taylor and amp, Francis Company. He worked as a reviewer for IEEE Transactions. His research areas of interest include computer vision, pattern recognition, medical image analysis and data analytics. He can be contacted at email: kiranchandrappa@gmail.com.