Improved recognition of aged Kannada documents by effective segmentation of merged characters

Madhavaraj A, A G Ramakrishnan, Shiva Kumar H R
MILE Laboratory, Dept. of Electrical Engineering
Indian Institute of Science
Bangalore, India
madhavarajaa, agrkrish, shivahr@gmail.com

Nagaraj Bhat
Department of Electronics and Communication
Birla Institute of Technology
Mesra, Ranchi
nagbhat25@gmail.com

Abstract—In optical character recognition of very old books, the recognition accuracy drops mainly due to the merging or breaking of characters. In this paper, we propose the first algorithm to segment merged Kannada characters by using a hypothesis to select the positions to be cut. This method searches for the best possible positions to segment, by taking into account the support vector machine classifier's recognition score and the validity of the aspect ratio (width to height ratio) of the segments between every pair of cut positions. The hypothesis to select the cut position is based on the fact that a concave surface exists above and below the touching portion. These concave surfaces are noted down by tracing the valleys in the top contour of the image and similarly doing it for the image rotated upside-down. The cut positions are then derived as closely matching valleys of the original and the rotated images. Our proposed segmentation algorithm works well for different font styles, shapes and sizes better than the existing vertical projection profile based segmentation. The proposed algorithm has been tested on 1125 different word images, each containing multiple merged characters, from an old Kannada book and 89.6% correct segmentation is achieved and the character recognition accuracy of merged words is 91.2%. A few points of merge are still missed due to the absence of a matched valley due to the specific shapes of the particular characters meeting at the merges.

Keywords— optical character recognition; aspect ratio; merged character segmentation; recognition based segmentation; support vector machine; recognition score; OCR; Kannada; matched valleys; segmentation path; vertical projection profile.

I. INTRODUCTION

Optical character recognizers (OCRs) with good performance on aged printed documents are still not available for many Indian scripts [1]. The Medical Intelligence and Language Engineering (MILE) Laboratory at the Department of Electrical Engineering has been working on Tamil [2-4] and Kannada [5-6] OCRs for some time and in the recent past, about 200 Tamil books have been converted to Braille books using the Tamil OCR developed here, with a good graphics user interface known as PrintToBraille tool [7], which also supports Telugu now. Bilingual OCRs are also being developed to handle Devanagari or Roman script, along with Tamil [8-10] or Kannada [11] script, using script recognition at the word level using Gabor features [12-13].

Incorrect segmentation of merged characters is a major cause of poor performance of OCRs on old documents. There exist three types of merges as defined by [14], namely linear, nonlinear and overlapped. Linear merge is one in which the two characters could be separated by a linear function. A non-linear type is one in which the merged symbols can only be separated by a non-linear function. If the merged components cannot be separated either way, then it is called an overlapped merge. A wide range of techniques have been proposed to segment and recognize the merged characters in Chinese and Roman scripts, and can be basically classified as recognition-free and recognition-based approaches. In the former approach, a set of rules are used to segment the characters without using a recognizer, whereas in the latter approach, the likelihood scores for the segmented characters given by the recognizer are used to validate the segmentation.

There are many recognition based approaches to segment merged Roman characters from printed documents. Bayer et al. proposed a statistical cut classifier and a search algorithm for selecting the merge position [15]. Although this method works for different font styles, the search space can be very large and extremely time consuming if the width of the merged image is large. Zhang et al. [16] proposed a method to segment merged symbols in mathematical expressions by extracting concave points in the image by contour analysis, and then using them to construct segmentation paths which are then verified by a recognizer. Neural network has also been used to segment merged characters by using the shortest path algorithm seeking minimal penalty cuts [17]. Vertical projection profile (VPP) of the merged image has been used to find optimal cut positions along with the classification score for Chinese characters [18]. Similar technique by using VPP has been tried in [19] for segmenting merged Roman characters. In [20], the authors segment the merged Gurmukhi characters by first vertically cutting exactly in the middle of the merged pair and then finding the final cut position by choosing a column within an optimizing window and validating the cut position with the information from the recognizer and the aspect ratios of the split components. In [21-22], attention-feedback segmentation has been proposed for segmenting both merged and split characters from online handwritten Tamil words.

There are also many recognition-free segmentation approaches in the literature, most of which are designed for handwritten characters rather than for the printed ones.

We record our immense thanks to Technology Development for Indian Languages (TDIL), Department of Information Technology, Government of India for funding this work under the OCR consortium project.

Chang et al. [23] proposed a technique to construct a convex hull around the merged character and to use the typographic attributes derived from the concave residual in the hull along with the shortest path algorithm to properly segment printed merged characters without employing a classifier. In [24], the skeleton of the image was used along with self-organizing map networks to decide the optimal segmentation path for merged handwritten digits. Dropfall algorithm was proposed by Congedo et al. [25] to segment merges in handwritten numerals. This algorithm simulates the flow of a hypothetical drop from top to bottom over the contour of the merged image and to seep through the merged portion to split the characters.

In this paper, we report a maiden attempt to segment merged Kannada characters in printed documents. This recognition based method picks the lines joining matching valleys from the top and bottom contours of the image as the likely segmentation paths. The aspect ratio of the segmented characters and likelihood scores of support vector machine (SVM) classifier are used to choose the best possible cut paths. Our method can handle images with multiple merges and fonts with different sizes and styles (bold/italic). Our algorithm has proved to be more efficient in segmenting the merged characters when evaluated against the VPP based initial choice of segmentation paths.

The paper is organized as follows: Section II gives a brief overview about the Kannada script and the baseline recognizer used in our experimentation. Section III describes the matching valleys based algorithm to generate the candidate segmentation paths. In section IV, we describe the complete recognition based segmentation procedure to select the best set of segmentation paths among all possible candidates. Experimental results are presented and discussed in section V. Conclusion and proposed future work form section VI.

II. BASELINE KANNADA CHARACTER RECOGNIZER

The Kannada script has 34 consonants and 14 vowels and 10 numerals and they combine with each other to form consonant-consonant and consonant-vowel characters forming a total of 309 unique classes for the entire script. In consonant clusters, special graphemes called ottu occur below the baseline. For more details on Kannada graphemes, see [26-27].

A. Properties of Kannada script

The template is used to format your paper and style the text. All margins, column widths, line spaces, and text fonts are prescribed; please do not alter them. You may note peculiarities. For example, the head margin in this template measures proportionately more than is customary. This measurement and others are deliberate, using specifications that anticipate your paper as one part of the entire proceedings, and not as an independent document. Please do not revise any of the current designations.

B. Character Recognizer

The character recognizer assigns a particular label to each segmented component. This step is the heart of any OCR system and involves two principal stages, namely (i) feature extraction and (ii) classification. During feature extraction, the relevant discriminative information, named the feature vector, is extracted from the raw character image. In the classification stage, the extracted feature vector is given to a classifier which outputs the label to which the vector belongs, based on a pre-trained model. Common discriminative classifiers used in OCR are the k-nearest neighbor, artificial neural network (ANN) and SVM. The accuracy of the OCR system mainly depends on the efficacy of the recognizer. Further, since the proposed method for segmenting merged characters is recognition based, character recognition is a key process and must be accurate and faster. Since the characters in Kannada are complex and contain many similar-looking pairs, choosing the type of discriminative feature vector is critically important.

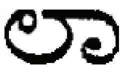
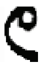
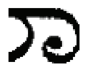
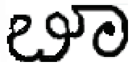
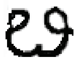
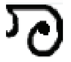
1) Feature extraction: We have used a combination of correlation and discrete wavelet transform feature. First, the character image is resized to 32x32. Then, we compute the one sided auto-correlation for each row of the image and similarly for each column of the image. Then, we compute the DWT of the image using Haar wavelets. Finally, elements of the row-wise correlation matrix, column-wise correlation matrix and the wavelet transform matrix (each of size 32x32) are concatenated to get the feature vector with a total dimension of 3072 (3x32x32 = 3072). In our experiments, the above set of features have proved to be more discriminative than the features obtained from template, discrete cosine transform (DCT), Karhunen-Loeve transform projections and block based DCT.

2) Classification: In addition to choosing appropriate features, it is important to choose a suitable classifier to realize a reliable character recognizer. We have used SVM with linear kernel as the classifier to train and test the feature vectors in our experiments. Separate SVM models are trained with feature vectors obtained from 64,811 samples (approx. 210 samples per class) for Kannada character classes. Also, statistics about the aspect ratios (width to height ratio) of the samples of each of the classes are calculated and are used for the purpose of verification during segmentation.

III. ALGORITHM TO GENERATE SEGMENTATION PATHS

The merge segmentation algorithms available in the literature were primarily designed for Roman characters and numerals, where the structure and shape of the characters are simple. However, in Kannada, the symbols have complex structures and the number of classes are also much higher that these algorithms cannot be applied, since almost all of these algorithms sometimes cut the valid Kannada characters at a position, where the components to the right and left of the cut might also be valid symbols but they are not actually merged. This phenomenon is illustrated in Table I. However, by exploiting the nature of the shape of the characters of Kannada (as almost every character has curved outer surface), we have derived an algorithm to efficiently segment the merged characters, which is described in the following sections.

TABLE I. EXAMPLES ILLUSTRATING HOW EXISTING SEGMENTATION TECHNIQUES IN THE LITERATURE MAY INCORRECTLY SPLIT SOME LEGITIMATE CHARACTERS TO RESULT IN OTHERWISE VALID KANNADA COMPONENTS.

Legitimate Symbol	Incorrect Split but valid sub-components	
 laa	 vowel_sign_ii	 vowel_sign_aa
 bau	 bi	 vowel_sign_aa

A. *Detection of merged characters:*

The first and foremost step is to decide whether an initially segmented component contains any merge or not. To do so, we have used the well known aspect ratio check. Once a segmented component is recognized, we decide it is merged if the aspect ratio of the character image is greater than the maximum aspect ratio of the recognized class. If and only if a segmented component is detected as merged, it goes further to the merge-segmentation algorithm.

B. *Method 1: Generation of segmentation paths using valleys:*

This is our proposed algorithm to construct possible linear segmentation paths in the merged image by matching the valleys obtained from the top and bottom contours of the image. The flow of the algorithm is discussed below.

1) Step 1: From the merged image, a list of valleys is obtained (Top valleys) using the algorithm given in Table II.

2) Step 2: The merged image is rotated by 180o and the list of valleys (bottom valleys) is once again obtained for this rotated image, using the same algorithm in Table II.

3) Step 3: This is the step that actually generates the segmentation paths. Each candidate segmentation path is the line joining a pair of top and bottom valleys. A pair is formed by taking one distinct point each from the list of top and bottom valleys. Further, priority is given for the pair of points which are minimally separated horizontally, and any pair whose horizontal distance is more than 5 pixels is not considered. Figure 1 describes the generation of candidate segmentation paths for a sample Kannada image. The top and bottom valleys are shown in red and black colors, respectively, and the segmentation paths are shown in yellow.

C. *Method 2: Generation of segmentation paths using vertical projection profile:*

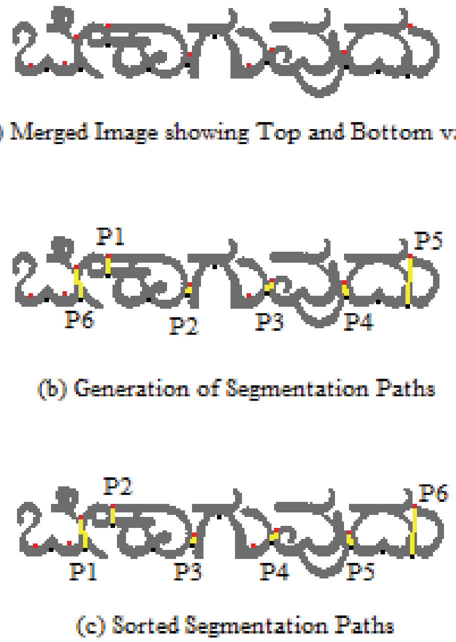
This method of generating the segmentation path is similar to the one described in [19]. Here, vertical segmentation paths are derived based on the vertical projection profile (VPP) of the merged image as discussed below.

TABLE II. ALGORITHM TO FIND VALLEYS

1. Initialize n=1 and create an empty list for storing the valley points.
2. Search for the first ON pixel in the nth column of the image, vertically from the top.
3. Note down the point. The pixel immediately above that point is called the *Hit Point* for that column.
4. If a valid hit point is not found, increment 'n' and go to Step 2; else from the hit point, traverse through the surface of the character and note down the lowest point as the *valley point* for the nth column.
5. If either the horizontal or the vertical distance between the hit point and the valley point is less than 2 pixels and if that point already exists in the list, then ignore that valley as a noisy local valley; else store the location of the point in the list of valley points. Increment 'n' and if 'n' exceeds the number of columns in the image, go to Step 6; else go to Step 2.
6. Return the final unique list of the valley points in the ascending order of their column numbers.

1) Step 1: Obtain the VPP of the given image and its first order derivative (VPP') and the second derivative (VPP").

2) Step 2: Note down the columns corresponding to minima in VPP, where VPP' and VPP" values at that column are zero and positive, respectively. Now, the selected column numbers are stored in a list, while ensuring that the horizontal distance between any two chosen columns is greater than 10 pixels (since no character has a width less than 10 pixels in Kannada for normal font sizes at 300 dpi scanning resolution).



(a) Merged Image showing Top and Bottom valleys

(b) Generation of Segmentation Paths

(c) Sorted Segmentation Paths

Figure 1. Valleys and candidate segmentation paths for a sample Kannada merged word image. The top and bottom valleys are shown in red and black color, respectively and the segmentation paths are shown in yellow.

This gives the list of indices of all the columns in the image, where a merge has been suspected.

3) Step 3: Make a vertical cut on the suspected columns in the merged image. Figure 2 shows a totally merged word image and the candidate columns chosen to be cut based on VPP.

IV. RECOGNITION BASED SEGMENTATION

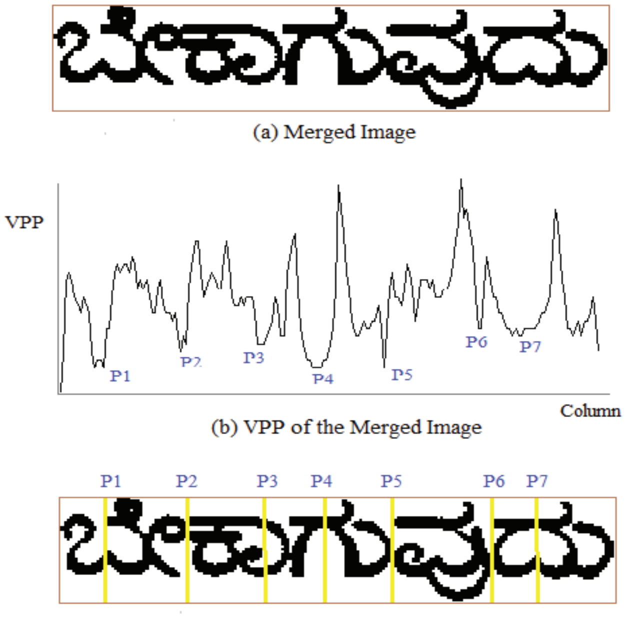
This algorithm is the core of the segmentation process. Once possible segmentation paths are available, the objective is to choose the best set of segments out of all combinations of the segments, which involves the two steps described below.

A. Initial segmentation and recognition

Given the candidate segmentation paths, the components lying between any pair of segmentation paths are possible segments. If there are 'N' segmentation paths, the number of such segments to be considered is (N2+3)/2. Thus, the number of different segments to be considered for the image shown in Fig.1 with 6 segmentation paths is 27. Each such segment is classified using an SVM classifier and its likelihood value is noted. If the aspect ratio of the segment doesn't lie within the minimum and maximum aspect ratio limits of the recognized label, then its likelihood is set to a minimum value (in our experiment, this value is -100).

B. Optimal selection of segments

Once initial segmentation and recognition is completed, the optimal paths for segmentation need to be selected. The average likelihood of all possible sequences of segments (from the left most to the right most column) are calculated and based on the maximum score, the final optimal sequence of segments segments is selected. The number of possible segmentation sequences which can be derived from 'K' segmentation paths is $2^{K-1}-1$. Fig. 3 shows a sample merged Kannada character image with 3 possible segment sequences and maximum likelihood score based selection of optimal set of segments.



(a) Merged Image

(b) VPP of the Merged Image

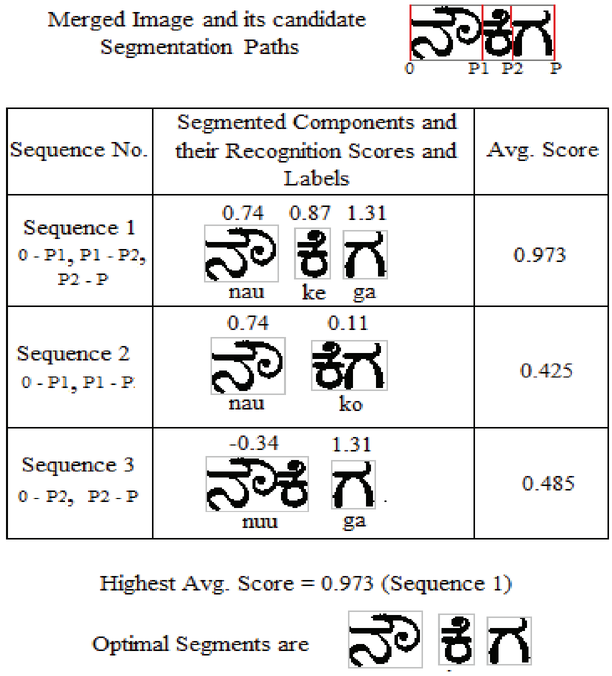
(c) Segmentation paths on the suspected columns

Figure 2. Candidate segmentation columns using VPP method for a sample Kannada merged word image.

V. RESULTS AND DISCUSSION

The segmentation and the resulting recognition performance of the proposed valley matching based segmentation method has been tested on 1125 merged images from a Kannada book. Each of these images contains two or more characters merged together, the statistics of which is given in Table III. The segmentation and recognition performance are tabulated in Table IV. A segmentation accuracy (the ratio of the number of correctly segmented characters to the total number of characters merged in the entire test set) of 89.6% is obtained for matched valley based segmentation. The character recognition rate after segmenting the merges is 83.4% (using VPP based segmentation) and 91.2% (using matched valley based segmentation) for this old Kannada book.

Fig. 4 compares the segmentation results using matched valleys and VPP based methods for five sample Kannada merged character images. It can be seen from the figure that when the characters are slanted (i.e. italicized), VPP based method performs poorly since the projections do not contain minima in the merged column due to the slant. However, the matched valley based segmentation performs well even in such cases. Fig. 5 shows the image of a sample Kannada text line with multiple merges and compares the recognized Unicode texts obtained after segmentation independently by valley matching and VPP based methods.



Sequence No.	Segmented Components and their Recognition Scores and Labels	Avg. Score
Sequence 1 0 - P1, P1 - P2, P2 - P	0.74 0.87 1.31 nau ke ga	0.973
Sequence 2 0 - P1, P1 - P	0.74 0.11 nau ko	0.425
Sequence 3 0 - P2, P2 - P	-0.34 1.31 nuu ga	0.485

Highest Avg. Score = 0.973 (Sequence 1)

Optimal Segments are

Figure 3. Selection of the optimal sequence of segments from a sample merged Kannada image.

TABLE III. STATISTICS OF THE NUMBER OF MERGED IMAGES TESTED AND THE NUMBER OF KANNADA SYMBOLS PER MERGED IMAGE

# of merged components per test image	# of images tested with merged components
2	524
3	298
>3	303

TABLE IV. SEGMENTATION AND RECOGNITION PERFORMANCE OF OUR ALGORITHM ON 1125 MERGED IMAGES FROM AN OLD KANNADA BOOK.

Selection of segmentation candidates	Segmentation accuracy (%)	Recognition rate of segmented characters (%)
Not carried out	N/A	0
By matched valleys	89.6	91.2
By valleys in VPP	84.3	83.4

VI. CONCLUSION

We have proposed, implemented and tested a method to generate possible segmentation paths for merged characters based on matching top and bottom valleys formed at the merge location. To our knowledge, this is the maiden report of a comprehensive technique for detecting and segmenting merged characters in Kannada, tested reasonably well on real word images obtained from old printed books in Kannada.

The results have been compared with those based on the VPP based segmentation. Finally, a method to optimally select the sequence of segments from all possible sequences has been proposed and implemented. The results show that a segmentation accuracy of 89.6% has been achieved by our segmentation algorithm, compared to 84.3% using VPP based segmentation. Similarly, improvements in character recognition accuracy of matched valleys based model over VPP based model show that the former performs better.

ACKNOWLEDGMENT

We thank all the research investigators of the OCR consortium, and in particular, Prof. C V Jawahar, IIIT, Hyderabad for defining the annotated database required for the development of OCR technologies and their partnership. Our thanks are also due to Sandeep Vazrapu, Sushirdeep Narayana and Nethravathi Kulkarni for their help in evaluating the effectiveness of the segmentation approaches reported here on many images. We gratefully acknowledge Shanthi Devaraj, Shanthi Raveesh and Saraswathi for participating in the creation of the annotated databases used in this work.

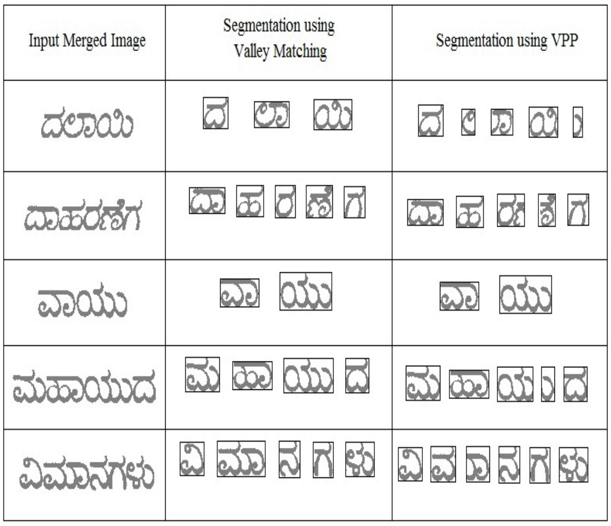
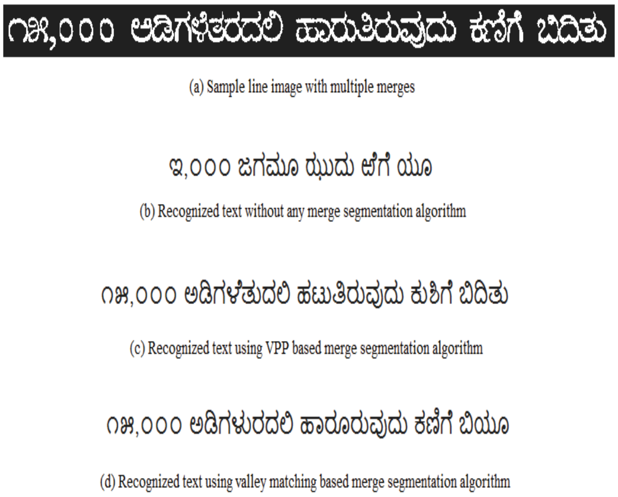

Input Merged Image
Segmentation using Valley Matching
Segmentation using VPP

Figure 4. Comparison of sample segmentation results obtained by matched valleys and VPP based techniques.


೧೫,೦೦೦ ಅಡಿಗಳೆತ್ತರದಲಿ ಹಾರುತಿರುವುದು ಕಣ್ಗೆ ಬಿದಿತು
ಇ,೦೦೦ ಜಗಮೂ ಝುದು ಆೆಗೆ ಯೂ
೧೫,೦೦೦ ಅಡಿಗಳೆತುದಲಿ ಹಟುತಿರುವುದು ಕುಶಿಗೆ ಬಿದಿತು
೧೫,೦೦೦ ಅಡಿಗಳುರದಲಿ ಹಾರೂರುವುದು ಕಣಿಗೆ ಬಿಯೂ

(a) Sample line image with multiple merges

(b) Recognized text without any merge segmentation algorithm

(c) Recognized text using VPP based merge segmentation algorithm

(d) Recognized text using valley matching based merge segmentation algorithm

Figure 5. Comparison of recognized text for a sample Kannada line image with multiple character merges, after segmentation by matched valleys and VPP based methods.

REFERENCES

[1] Peeta Basa Pati and A G Ramakrishnan, “OCR in Indian Scripts: A Survey,” *IETE Technical Review*, May-Jun 2005, 22(3):217-227.

[2] A. G. Ramakrishnan and Kaushik Mahata, “A Complete OCR for Printed Tamil Text,” Proc. Tamil Internet 2000, Singapore, July 22-24, 2000, pp. 165-170.

[3] K. G.Aparna and A. G. Ramakrishnan, “A complete Tamil Optical Character Recognition System,” Proc. Fifth IAPR Workshop on Document Analysis Systems DAS-02, Princeton, NJ, August 19-21, 2002, pp. 53-57.

[4] Tushar Patnaik, Shalu Gupta, C V Jawahar, Santanu Choudhury, A G Ramakrishnan, “Design and Evaluation of Omnifont Tamil OCR,” Proc. Tamil Internet 2010, Coimbatore, June 23-26, 2010, pp. 519 - 522.

[5] B. Vijay Kumar and A. G. Ramakrishnan, “Machine Recognition of Printed Kannada Text,” Proc. Fifth IAPR Workshop on Document Analysis Systems (DAS-02), August 19-21, 2002, Springer Verlag, Berlin. pp. 37-48.

[6] Vijay Kumar.B and A. G. Ramakrishnan, “Radial basis function and subspace approach for printed Kannada text recognition,” Proc. IEEE ICASSP-04, May 17-21, 2004, Vol 5, pp. 321-324.

[7] Shiva Kumar H R and A G Ramakrishnan, “A tool that converted 200 Tamil books for use by blind students,” Proc. 12-th International Tamil Internet Conf., Kuala Lumpur, Malaysia, Aug. 15-18, 2013.

[8] D Dhanya and A G Ramakrishnan, “Optimal feature extraction for Bilingual OCR,” Proc. Fifth IAPR Workshop on Document Analysis Systems (DAS-02), Princeton, NJ, August 19-21, 2002, pp.25-36.

[9] D.Dhanya and A.G.Ramakrishnan, “Simultaneous recognition of Tamil and Roman scripts,” Proc. Tamil Internet 2001, Kuala Lumpur, August 26-28, 2001, pp. 64-68.

[10] Deepak Arya, C V Jawahar, C Bhagvati, T Patnaik, B B Chaudhuri, G S Lehal, S Chaudhury, A. G. Ramakrishnan, “Experiences of integration and performance testing of multilingual OCR for printed Indian scripts,” Proc. Joint Workshop on Multilingual OCR and Analytics for Noisy Unstructured Text Data (MOCR AND '11). Sept. 17, 2011, Beijing, China. Article 9.

[11] R S Umesh, Peeta Basa Pati and A G Ramakrishnan, Design of a bilingual Kannada-English OCR, in the book “Guide to OCR for Indic Scripts: Document Recognition and Retrieval” Springer, 2009 in the Advances in Pattern Recognition Series. Ed: Venu Govindaraju and Setlur Srirangaraj. pp. 97-124. ISBN: 978-1-84800-330-9

[12] D. Dhanya, A. G. Ramakrishnan, P. B. Pati, “Script Identification in printed bilingual documents,” Sadhana, Feb 2002, Vol. 27, part-1, pp.73-82.

[13] Peeta Basa Pati and A. G. Ramakrishnan, “Word Level Multi-script Identification,” Pattern Recognition Letters, 2008, Vol. 29, pp. 1218-1229.

[14] J. Song, Z. Li, M. R. Lyu and S. Cai, “Recognition of merged characters based on forepart prediction, necessity-sufficiency matching and character-adaptive masking,” IEEE Trans. Syst., Man, Cybern. B, vol. 35, no. 1, pp. 2-11, 2005.

[15] T. Bayer, U. Krebel, and M. Hammelsbeck, “Segmenting merged characters”, Proc. XI Intern. Conf. Pattern Recognition, vol. II. Conf. B: Pattern Recog. Methodology and Sys, 1992.

[16] Dong-Yu Zhang, Xue-dong Tian, Xin-fu Li, "An Improved method for segmentation of touching symbols in printed mathematical expressions," Proc. 2nd International Conf. Adv. Comp Control (ICACC), vol.2, pp. 251-253, 27-29 March 2010.

[17] Jin Wang, Jack Jean, “Segmentation of merged characters by neural networks and shortest path”, Pattern recognition, Elsevier, vol. 27, Issue 5, May 1994, pp. 649–658.

[18] Wuyi Yang, Shuwu Zhang, Haibo Zheng, Zhi Zeng, "A recognition-based method for segmentation of Chinese characters in images and videos," Proc. International Conf Audio, Language and Image Processing, ICALIP 2008. pp. 723-728, 7-9 July 2008.

[19] S. Messelodi and C. M. Modena, Context driven text segmentation and recognition, Pattern Recognition Letters, Volume 17(1), Jan 1996, Pages 47-56, ISSN 0167-8655, http://dx.doi.org/10.1016/0167-8655(95)00098-4.

[20] N. M. Davessar, S. Madan, and Hardeep Singh, "A hybrid approach to character segmentation of Gurmukhi script characters," Proc. 32nd Applied Imagery Pattern Recognition Workshop, pp. 169-173, 15-17 Oct. 2003.

[21] Suresh Sundaram and A. G. Ramakrishnan, “Attention feedback based robust segmentation of online handwritten words,” Indian Patent Office Reference. No: 03974/CHE/2010.

[22] Suresh Sundaram and A. G. Ramakrishnan, “Attention-feedback based robust segmentation of online handwritten isolated Tamil words,” ACM Transactions on Asian Language Information Processing (TALIP), Vol. 12 (1), March 2013, Article No. 4.

[23] T. C. Chang, and S. Y. Chen, “Character segmentation using convex-hull techniques,” Int. J. Pattern Recognition and Artificial Intelligence, vol. 13, no. 6, pp. 833-858, 1999.

[24] E. B. Lacerda and C. A. B. Mello, "Segmentation of touching handwritten digits using self-organizing maps," Proc. 23rd IEEE International Conf on Tools with Artificial Intelligence (ICTAI), pp. 134-137, 7-9 Nov. 2011.

[25] G. Congedo, G. Dimauro, S. Impedovo, and G. Pirlo, "Segmentation of numeric strings," Proc. Third International Conf. Document Analysis and Recognition, vol.2, pp. 1038-1041, 14-16 Aug 1995.

[26] B. Nethravathi, C. P. Archana, K. Shashikiran, A. G. Ramakrishnan, V. Kumar, "Creation of a huge annotated database for Tamil and Kannada OHR," International Conf Frontiers in Handwriting Recognition (ICFHR), pp. 415-420, 16-18 Nov. 2010.

[27] M. Mahadeva Prasad, M. Sukumar, A. G. Ramakrishnan, “Divide and conquer technique in online handwritten Kannada character recognition,”ACM - Proceedings of the International Workshop on Multilingual OCR, 2009.

.