Segmentation of Degraded Malayalam Words:Methods and Evaluation

Devendra Sachan , Shrey Dutta , Naveen T S and C.V. Jawahar
Center for Visual Information Technology, IIIT Hyderabad, India
d.sachan@iiitg.ernet.in , shreydutta@hotmail.com , naveen.ts@students.iiit.ac.in , jawahar@iiit.ac.in

***Abstract*—In most of the Optical Character Recognition softwares, a substantial percentage of errors are caused by the incorrect segmentation of degraded words. This is especially true for recognizing old books, newspapers and historical manuscripts. In this paper, we propose multiple segmentation methods which address the problem of cuts and merges in degraded words. We have created an annotated dataset of 1034 word images with pixel level ground truth for quantitative evaluation of the methods. We compare the methods with a baseline implementation based on connected component analysis. We report substantial improvement in accuracy both at character and at word level.**

***Keywords*-Character Segmentation; Degradation Correction; Malayalam; Indian Language;**

I. Introduction

Optical Character Recognition (OCR) is one of the most successful applications of pattern recognition [1]. High recognition accuracies are reported for a wide range of documents in Roman scripts. Indian language OCRs are now reporting comparable results on a class of documents (eg. printed books) [2], [3]. However, the accuracies are still rather low for poor quality documents, where there are significant degradations. Newspapers and historic documents fall into this category of documents. The primary reason behind the poor recognition accuracy is the failure to address cuts (i.e. splitting of genuine components) and merges (i.e. touching of two different components to form a single connected component). These degradations arise out of various sources like poor ink quality, ageing of documents, low spacing between characters (for example scanned documents of newspaper clips), poor pre-processing etc.

One of the contributions of this work is a set of script specific heuristic methods addressing cuts and merges in character segmentation. This is not the first time, character segmentation is investigated in literature [4], [5], [6], [7]. Most of the previous work has been on (i) Roman scripts and/or (ii) handwritten data. A survey on methods and strategies in character segmentation, with specific emphasis on English, is given in [8]. Indian language OCRs usually employ simple strategies (eg. connected component analysis) for character segmentation. Even if some refinements to this naive segmentation is used, it is often not documented in enough detail (eg. see the descriptions of Indian Language OCRs in [2]). This paper aims at formally documenting some of the possible methods; more importantly, how such methods can be designed and evaluated.

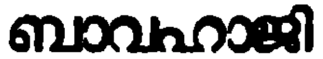
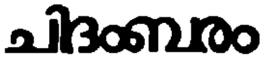
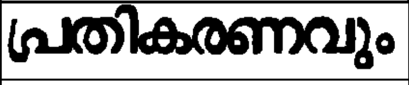
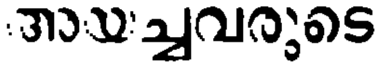
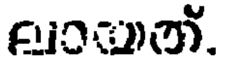
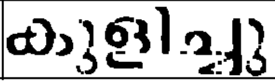
Sl. No.	Image	# CC	True # CC
a.		1	7
b.		2	7
c.		4	10
d.		19	8
e.		23	6
f.		21	6

Fig. 1. Samples of degraded images from our dataset. First three images have merges and the last three have cuts. Note the actual number of connected components(# CC) and true number of connected components(True # CC)

Indic scripts pose additional challenges to the segmentation. Unlike English, Indic characters are often composed of curves rather than straight lines which makes it prone to cuts and merges. Bansal *et al.* [9] as well as Garain *et al.* [10] have presented methods for segmenting touching or fused Devanagari characters. Our focus is on Malayalam, a Dravidean language with significant differences from Devanagari in script and writing style. We propose methods which can segment words with merges as well as cuts.

Another contribution of this work is the introduction of a database of degraded as well as normal words for formal evaluation of the segmentation algorithms. We also propose an annotation schema for representing and processing the true segmentation information. Our database contains more than 1000 annotated Malayalam word images obtained from different newspapers and books. More details are provided in the next section. Sample images from the database are shown in Figure 1. We also show the actual number of connected components as well as the true number of connected components. As seen from the examples, it is quite possible that the entire word is fused to form a single connected component. Segmentation of such a word is quite challenging for a machine, while it is trivial for a human being, specially

978-0-7695-4599-8/11 $26.00 © 2011 IEEE
DOI 10.1109/NCVPRIPG.2011.23

TABLE I
SUMMARY OF THE DATA SET

	Total	Cuts	Merges	Normal
Characters	7719	877	1214	5628
Words	1034	422	400	212

for a native reader. The primary goal of this work is the segmentation of such degraded words. We evaluate the accuracy of segmentation directly by annotating the word images at pixel-level. This avoids the need to evaluate the performance of the segmentation module with the help of OCR accuracy, as is done in some of the previous works.

A related work to ours is [11], where errors in recognition due to segmentation were overcome with a (top-down) language model. Our present approach is complementary to this, and addresses the segmentation problem in a bottom up manner. The methods presented here are better suited for a traditional OCR architecture.

In this paper, we propose a set of strategies for detecting and correcting the degradations. We report an accuracy of 94.4% in correctly segmenting characters on a collection of 7719 characters. This results in a word level segmentation accuracy of 72% which is considerably higher than the baseline accuracy of 20% obtained through connected component analysis.

II. Dataset and Evaluation

A. Dataset

Formal evaluation of the algorithms is critically needed to know the utility of methods. This is especially true if the methods use script specific heuristics. For the evaluation of segmentation methods, we created a dataset of 1034 word images containing 400 merged, 422 cuts and 212 normal images. Normal images are those which do not have any degradation. The details are shown in Table I. A method which is aimed at addressing merges may over segment the normal images or images with cuts. Our data set is composed of all the three categories to make sure that we are aware of the performance of the methods on each of these categories.

The word images are selected from commonly available Malayalam newspapers and books. Both the original image as well as its binarized image are present in the database. We ground truth the dataset to make the automatic evaluation possible. In the ground truth, every foreground pixel is labeled with an identification number for the component. For the sake of rendering, we show each component in a separate color. The components have been coloured in such a way that no two adjacent components have the same color. An example of a word image with cut along with its ground truth is shown in Fig 2(a) and a merged image with its ground truth is shown in Fig 2(b). For images with merges, the merged components are first explicitly separated and then they are labeled. Similarly, in the case of cut, all the components of a character are joined together and then assigned the same color.

B. Evaluation Criteria

The primary objective of our paper is to evaluate the segmentation accuracy. The following measures will be used to evaluate our methods.

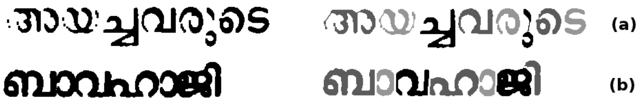


Fig. 2. Sample ground truth images. Top figure shows the original and ground truth image for cut and bottom figure shows the original and ground truth image for merge

1) Character Segmentation Accuracy: This measure represents the percentage of characters which are correctly segmented using a method.

It is calculated as *Total number of correctly segmented characters / Total number of characters.*

2) Word Accuracy: This represents the percentage of words where all of its characters have been correctly segmented. If even a single character in a word is segmented wrongly, the word is considered as wrong.

It is calculated as *Total number of correctly segmented words / Total number of words.*

These measures will be used to evaluate both the individual methods as well as the hybrid methods.

3) Degradation Detection Accuracy: This measure shows the degradation (cut/merge) detection percentage of the respective methods. We also measure the percentage of merge/cut characters detected correctly by the merge/cut detection algorithms. The evaluation has been done automatically as manual visual verification of such a large set of data is not practical. This has helped us in computing the accuracy of various combination of the proposed methods.

III. Methods for Segmenting Word Images

In this section, we propose a set of methods which can be used for segmenting degraded words in documenting images. Though these methods are demonstrated for Malayalam, they could also be applicable for other scripts with similar character shapes. We group the methods into two classes. The methods in the first category takes care of merges while those in the second category are designed to take care of cuts. We also use a combination of methods from these two categories to create a set of hybrid methods. The different methods for identifying degradations are explained below.

A. ***Segmenting Merged Characters***

1) M1: In this method, we first extract connected components(CCs). Then we analyze each CC for the presence of merges in it. In every vertical column, we record the first and last black pixel positions in v_1 and v_2. We then find out the position of local maxim in v_1 and position of local minim in v_2. If a local maxim and a local minim are very near, then we treat this as a possible merge, and do an explicit cut in between these two local extrema.

The notion of zones, is prominent in Devanagari and in some of the other scripts. Malayalam is typically written in the middle zone. However, some of the vowel modifiers extend to the top and bottom zones. A popular merge in Malayalam is due to the vowel modifiers touching the consonant in the

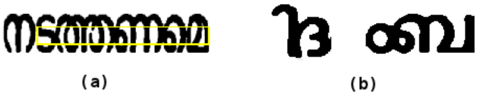


Fig. 3. Figure (a) shows identification of potential Merge location around centroid. Figure (b) shows a 'V' type merge and an Inverse-'V' type merge example.

top/bottom zone. We explicitly search for such merges and generate cuts at possible joints.

2) M2: Since Malayalam has large number of 'convex' curved characters, merges often happen in the middle zone at the center. To address such merges, we consider a horizontal strip around the centroid line as shown in Fig 3(a). In this strip of interest, for every column, we compute the transitions. If there exists one and only one white-to-black-to-white transition, it is considered as a potential merge.

3) M3: Another possibility of detecting merges is with the help of distance transform [12]. Distance transform represents the distance to the nearest boundary pixel. In presence of merge, the distances first decreases and then increases, without becoming zero in the between. This pattern is spotted in the distance transform. By looking at the local extrima of the distance transform, we hypothesize the possible merge location. The problem of detecting merged vowel modifiers with consonants is solved by checking for specific merges occurring in top and bottom zone.

B. Joining Broken Characters

We will now describe how to identify *the cut* in a word image. Unlike merges, detection of cuts is a bit tricky problem. The major reason being that cuts can occur at any location in an image. The following methods are proposed to detect cuts.

1) C1: In this method, the shortest distance(d_i) between each pair of CCs is calculated. If we conclude that the two components are near, these two CCs are identified as a potential cut. After this, traverse the grey scale image along these two points and compute the mean gray value of all the pixels encountered. If the value computed falls below a specified threshold, the components are joined along the shortest path.

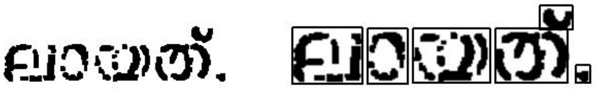


Fig. 4. The Components which lie very close to each other will be joined to form the actual character. Image on left is the original image and the right side image shows the individual components which will be joined together to form one character.

2) C2: In this method for every pair of CCs we measure the overlapping area of the bounding boxes. If the fraction of overlapping area is greater than 10 % of the union, we conclude that these two CCs are part of a single character (as shown in Figure 5). Based on a priori knowledge of language, care is taken so as to not merge specific characters. To tackle the issue of horizontal cuts in specific characters we put a constraint on the relative locations of the bounding boxes. If the size of any fragment is very small, then we assign it to be part of the nearest CC.

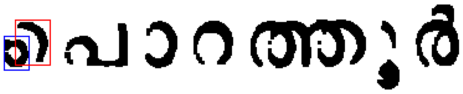


Fig. 5. The rectangles marked in red and blue have got considerable overlap area and hence will be considered as a part of single component

3) C3: Chances of cuts occurring in an image due to binarization is not uncommon. A good traversal in the gray-scaled image for the lost pixels can solve this problem. A cut results in generating more end points in binary image than what is present in grey scale. We find these end points after thinning the binary image using Huang *et. al* [13] algorithm.

Once we get a thinned image, we traverse the gray scale image from those endpoints and differentiate the background pixels from the foreground ones. During this traversal, if at anytime we identify a foreground pixel belonging to a different connected component, we say that these two CC are actually a part of a single CC. The distance between the two end points are checked to make sure that it lies within a threshold. If it does, the two components are joined.

C. Hybrid Methods

The biggest limitation of the proposed individual methods is that alone, they cannot be used effectively in a practical application as a normal dataset will contain a mixture of cuts, merges and normal characters. The algorithms which we proposed are designed to handle only a specific type of degradation (cut/merge).

Hence, we decided to introduce a set of hybrid methods which will use a combination of the individual methods which we had earlier proposed (e.g: M1, C2 and C3 algorithms being applied to same image). The methods which we propose shall contain atleast one merge algorithm and atleast one cut algorithm.

Any input image is send to the selected set of algorithms for processing. The individual algorithms will detect and correct the degradations which are present in the image. The result of these algorithms are combined together to generate a segmented output image.

IV. Results and Discussions

We report results of all the methods in symbol and word level accuracy in detecting cuts and merges. In other words, we test our algorithms on degraded words and compare the output of cut/merge detection algorithm with the ground truth. We say a symbol is correctly segmented if the symbol in the ground truth and output of the proposed methods are same. Similarly, a word is said to be correctly segmented, if we detect all the cuts and merges in that word accurately (i.e. same as ground truth).

Table II shows the accuracy on merged and cut characters for various algorithms. This metric is based on number of merge/cut characters detected. This is a method specific evaluation where we evaluate the merge detection algorithm against

TABLE II
DEGRADATION DETECTION ACCURACY(IN %)

Methods	Merge Detection	Cut Detection
M1	86.97	N.A.
M2	85.80	N.A.
M3	90.67	N.A.
C1	N.A.	84.72
C2	N.A.	62.03
C3	N.A.	93.56

the total number of merged characters present in the dataset. Similarly, the cut detection algorithms are evaluated only for those characters which have cuts present in them. The table shows that method M3 and C3 perform best in identifying degradation by correctly identifying 91% of merges and 94% of cuts.

Once the initial accuracy was obtained, the dataset is expanded to include all the 1034 words which is a combination of cut, merged and normal characters. This is done to simulate a practical environment where the algorithm has to deal with characters which may or may not be degraded.

Table III summarizes results of segmentation accuracy of merge and cut algorithms at symbol and word level for this dataset. The baseline accuracy on the dataset needs to be computed for comparing accuracy with our proposed methods. This is done by performing a simple connected component analysis of the input word image and comparing the result with ground truth. In other words, baseline result identifies only those characters which have no degradation. Any word which has atleast one degraded character in it will be identified as wrongly segmented. The results are shown in Table III and Table IV. Individually, method M3 gives us the best accuracy at symbol and word level for merges with about 13% improvement over baseline method at character level and around 24% improvement at word level. Similarly C3 for cuts show around 10% and 34% improvement respectively at character and word level while comparing against baseline results.

However, there is a drop in accuracy when comparing with Table II as the methods identifying merges are not detecting cuts and vice-verse. Hence, we decided to combine the individual methods to form a set of 'Hybrid' methods which can handle both cuts as well as merges. The results of the top hybrid methods are given in Table IV. We observe that these hybrid methods improve the accuracy significantly. This is also useful for practical applications where the dataset can contain cuts, merges and normal characters.

Table IV shows top 5 hybrid methods, sorted according to the word accuracy. In our individual methods, M3 and C3 have highest accuracy. Hence, the hybrid methods which contain those methods perform much better. As the table shows, we obtain more than 94% symbol level accuracy and 72% word level accuracy. The results show an improvement of around 52% in word accuracy and around 22% improvement in character accuracy while comparing with the baseline methods.

V. CONCLUSION AND FUTURE WORK

We have shown a set of individual as well as hybrid methods to correctly segment degraded Malayalam words. We show an improvement of around 52% in word and 22% improvement in character accuracy while comparing with the baseline methods.

TABLE III
SEGMENTATION ACCURACY (IN %) FOR VARIOUS CUT AND MERGE METHODS AT SYMBOL AND WORD LEVEL

Methods	Symbol Accuracy	Word Accuracy
Baseline Result	**72.79**	**20.50**
M1	84.17	42.65
M2	82.54	36.46
M3	85.26	44.78
C1	80.36	46.80
C2	77.41	35.01
C3	82.58	53.96

TABLE IV
SEGMENTATION ACCURACY (IN %) FOR VARIOUS HYBRID METHODS AT SYMBOL AND WORD LEVEL

Methods	Symbol Accuracy	Word Accuracy
Baseline Result	**72.79**	**20.50**
M3C3	94.44	72.24
M3C2C3	92.91	69.15
M1C3	93.09	66.63
M3C2	92.26	66.15
M3C1C3	91.74	65.86

We would like to expand our work to developing an OCR system which will help in correctly recognizing the degraded characters. We would also like to improve the computational efficiency of our algorithms.

ACKNOWLEDGMENT

This work is supported by Ministry of Communication and Information Technology, Government of India, New Delhi.

REFERENCES

[1] G. Nagy, "Twenty years of document image analysis in pami," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 1, pp. 38–62, 2000.

[2] V. Govindaraju and S. Setlur, *Guide to OCR for Indic Scripts*, 2009.

[3] D. Arya, T. Patnaik, S. Chaudhury, C. V. Jawahar, B.B.Chaudhuri, A.G.Ramakrishna, C. Bhagvati, and G. S. Lehal, "Experiences of Integration and Performance Testing of Multilingual OCR for Printed Indian Scripts," in *MOCR Workshop,ICDAR*, 2011.

[4] M. Gilloux, J. Bertille, and M. Leroux, "Recognition of handwritten words in a limited dynamic vocabulary," in *IWFHR III*, 1993.

[5] R. Fenrich, "Segmenting of automatically located handwritten numeric strings," in *Proceedings of the 2nd IWFHA*, 1992, pp. 33–34.

[6] J. Favata and S. Srihari, "Recognition of general handwritten words using a hypothesis generation and reduction methodology," in *USPS Advanced Technology Conference*, 1992.

[7] J. Tse, C. Jones, D. Curtis, and E. A. Yfantis, "An ocr-independent character segmentation using shortest-path in grayscale document images," in *ICMLA*, 2007, pp. 142–147.

[8] R. G. Casey and E. Lecolinet, "A survey of methods and strategies in character segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, pp. 690–706, July 1996.

[9] V. Bansal and R. M. K. Sinha, "Segmentation of touching and fused devanagari characters," *Pattern Recognition*, vol. 35, no. 4, pp. 875–893, 2002.

[10] U. Garain and B. B. Chaudhuri, "Segmentation of touching characters in printed devnagari and bangla scripts using fuzzy multifactorial analysis," in *ICDAR*, 2001, pp. 805–809.

[11] K. Mohan, K. J. Jinesh, and C. V. Jawahar, "Towards recognition of degraded words by probabilistic parsing," in *Proceedings of the Seventh ICVGIP*, 2010, pp. 375–382.

[12] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 2008.

[13] L. Huang, G. Wan, and C. Liu, "An improved parallel thinning algorithm," in *ICDAR*, 2003, pp. 780–783.