Extraction of Characters and Modifiers from Handwritten Gujarati Words

Chhaya Patel
MCA Department
Anand Institute of Information Science,
Anand, India

Apurva Desai
Department of Computer Science,
Veer Narmad South Gujarat University,
Surat, India

ABSTRACT
The research activity related to Optical Character Recognition (OCR) for almost all Indian languages is very less. Gujarati script is one of the scripts for which very less literature is available, as far as OCR activities are concerned. This paper describes one of the important phase of OCR, segmentation of handwritten words into its basic components namely basic characters, conjunct characters and modifiers, which are essential for recognition of a word. The paper describes methods for identification of zone boundaries for a word and usage of zone boundaries details for segmenting the word into its subcomponents. Connected component labeling is applied to detect subcomponents of a word, which can be further dissected if needed to obtain other subcomponents of word. It is the first attempt to dissect handwritten Gujarati words into its subcomponents.

Keywords
zone identification; upper zone; lower zone; middle zone; distance transform; shirolekha; character extraction; modifier extraction ; connected component labeling

1. INTRODUCTION
Gujarati is a language from the Indo-Aryan family of languages, used by more than 50 million peoples in the Indian states of Gujarat. It is a popular way of communication for Gujarati people, staying in all most every country of the world. A formal grammar of the precursor of this language was written by Jain monk and eminent scholar Hemachandracharya. It has a rich oral culture and a literary tradition which dates back to the tenth century. The construction of Gujarati can be considered somewhere between those of Hindi and Marathi [1].

Very few attempts are found for the OCR activities related to Gujarati. The first ever work of character recognition for printed or digitized Gujarati language is found in 1999 [2]. As far as handwritten OCR is concerned almost negligible work is found. Gujarati script is used to write the Gujarati language. Like other Indian languages, Gujarati is also a multilevel script. The absence of header line or shirolekha in Gujarati words is a special feature of this script. The problem of segmentation of the words needs more attention due to absence of shirolekh. Most of the Indian languages have additional problems as far as segmentation of words into its subcomponent is concerned caused by presence of combined characters in a word. The presence of modifiers in upper, lower or middle part of the word is another reason to increase difficulties in segmentation. These modifiers may be connected with the main character or they may appear separately.
Even for various languages the place of modifiers is not common e.g. some of the south Indian languages have different positions for modifiers than the north Indian languages. Due to these characteristics the techniques developed for the segmentation of words for languages other than Indian languages are not useful. Each Indian language have their own features, restricting development of a generalized OCR for Indian scripts.

2. CHARACTERISTICS OF GUJARATI SCRIPT
The character set of Gujarati language comprises of - 35 consonants, 13 vowels and 6 signs, 13 dependent vowel signs, 4 additional vowels for Sanskrit, 9 digits and 1 currency sign. The consonants can be combined with the vowels, and consonants to form conjunct consonants.

Gujarati words are formed by combining the basic consonant(s) and conjunct consonants which may by combined with one or more than one vowel(s)/modifier(s). A text line is collection of words, written in left to right direction. Fig. 1 shows a printed Gujarati text line with various zones and zone divider lines.

- The imaginary line that separates the middle and upper zone is named mean line.
- The imaginary line that separates the middle and lower zone is named base line.
- The part of the text below the base line, used for writing dependent (lower) vowels is called Lower Zone.
- The part of the text above the mean line, used for writing dependent (upper) vowels is called Upper zone.
- The part of the text on which consonants and independent vowels are written (between the Mean line and Base line) is called Middle zone.

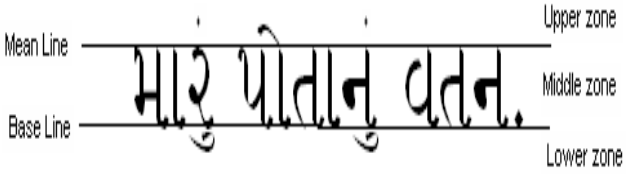

Figure 1. Zones in a Gujarati text line

One can observe that a word can also be segmented in three zones similar to the text line.

3. IMPORTANCE OF ZONE IDENTIFATION
The word in any language can be identified using two methods, either as a complete consonant-vowel cluster i.e. as a distinct entity or it may be segmented first into its subcomponents and then individual subcomponent is recognized separately for identification of a word. One can use either of these approaches for identification of Gujarati words too. The possible combinations of characters/consonants, modifiers and conjunct consonants, generates approximately 2.5 million Gujarati

words. Thus the first approach is more exigent for recognition of Gujarati words.

It is suggested in [3, 4] that the second option is also found more feasible than the first one for many Indian languages including Devnagari. The word is segmented into its basic components i.e. consonants and modifiers, which are recognized at a later stage in order to identify the word. Many researchers have suggested this approach for OCR activities for Indian languages. A good survey and review of various approaches for character segmentation are found in [8].

In [7] an approach based on dissection is described to segment lower modifiers. This approach identifies the lower modifier separator line using character height information and by filtering primary lower modifier containers using the features of the core units and the lower modifiers. Another approach is described in [9] which use the separator line as a base line an abrupt change in the sum of gray values between two consecutive rows is used for detecting the base line. One can find similar approaches in [10] and [11]. A work on the recognition of printed Oriya script is presented in [6]. This work describes line segmentation, word segmentation, character segmentation and character recognition for the Oriya script.

Detection of zone boundaries or zone identification has an important role in segmentation phase for almost all Indian languages. The separation of upper and lower zone eases detection of the middle zone and hence base characters and modifiers present in middle zone. The modifiers in upper and lower zone can be easily detected once the upper and lower zone are identified.

4. CHARACTERS AND MODIFIER EXTRACTION FROM A GUJARATI WORD

In case of handwritten words the identification of zones is not sufficient for extraction of all sub components of a word due to several unsolved challenges caused by presence of slanted characters, connected characters, uneven thickness of the characters, variety of handwritten forms etc. Before deriving a solution for the word segmentation for Gujarati handwritten word, we have to understand the composition of a typical Gujarati word.

A typical Gujarati word is a combination of one or more characters and /or conjunct characters and/or modifiers. Every basic character/conjunct character may be combined with the modifiers. Following Fig. 2 shows various combinations of basic character ‘va’ with different modifiers.

Figure 2. Various combinations of basic character ‘va’ with different modifiers

If we divide the word into the three zone as suggested in Fig. 1, it is noticed that there are two modifiers that appears purely in the lower zone namely ‘U’ and ‘Ū’, three modifiers has their presence purely in the upper zone namely ‘e’, ‘ai’ and ‘am’. The modifiers ‘ā’ and ‘ah’ are the modifiers having their presence purely in middle zone. On the other hand the modifiers ‘I’, ‘Ī’, ‘O’ and ‘AU’ have their presence in middle zone as well as in upper zone.

Thus one needs to derive methods to extract basic symbols from middle zone and modifiers from at most two zones out of the available three zones. One can derive a method to dissect the word into three zones first and then further dissect the individual zones to get the individual subcomponents, or one can directly detect the characters and modifiers from the word. Here a combination of both approaches is suggested. First the zone boundaries are detected and then subcomponents of the word are detected individually. As handwritten text is being considered, there are possibilities that the basic characters may be connected with modifier(s); a method to separate them is required in such cases. The zone boundaries are used to segment such connected components. The separated characters and modifiers are assigned to recognizer for identification at later stage.

5. ZONE IDENTIFICATION

The presence of header line or shirolekha in case of scripts like Devnagari, Bangla and Gurumukhi helps to detect the header line as it generates a prominent peak in the horizontal projection profile of a word as shown in the Fig. 3. But the same method is not helpful to determine the upper zone for Gujarati words due to absence of shirolekha, as shown in Fig. 4.

It is also found that it is easier to identify zone boundaries if the word is in printed form compared to handwritten form irrespective of any script. A procedure to identify zone boundaries for printed Gujarati text line is described in [5]. Here it is argued that use of a trough in the horizontal projection of Gujarati word is not useful to detect the upper zone boundary especially in the cases where the number of modifiers is significantly large, the text or word is misaligned, or if the number of modifiers are very less.

Additional challenges for zone detection are introduced due to variation in inter character alignment and variation in size of characters, for handwritten Gujarati words. Uneven ratio of modifiers and consonants size may also lead to error in zone identification. The style of handwritten form has infinite possibilities hence one cannot predefine any parameters to identify the zone boundaries based on the characteristics of the script.

Figure 3. Devnagari word and its horizontal projection profile

Figure 4. A Gujarati word with connected upper modifier and its horizontal projection profile

An algorithm described in [13] is used to determine zone boundaries. The segmented zones are used for further segmentation to extract modifiers and basic characters. A simple vertical projection profile based approach may be applied for further segmentation of upper and lower zone to extract upper and lower modifiers respectively.

It is found that the same approach is not suitable for the middle zone. Hence a new algorithm based on the connected

component labeling is developed to extract the subcomponents of the word.

6. CHARACTERS AND MODIFIERS EXTRACTION

The characters and modifiers are separate entities for very well written Gujarati words. For the words with cursive writing style, improper writing style or due to slip of pen, there are possibilities that the modifiers and characters are not separable easily. Traditional methods like vertical projection profile analysis may cause over segmentation in many cases. The connected component labeling method is applied to uniquely identify an individual subcomponent of the word. An individual separated subcomponent may be a character, modifier or combination of character and modifiers, if they are connected with each other due to bad writing style. Proposed approach applies morphological operations on the original word to separate individual components of the word to decrease possibilities of under segmentation.

Connected components labeling: This method groups an image pixels into components based on pixel connectivity. The labeled matrix is generated for the binary image of a word. Connected components labeling scans an image and groups its pixels into components based on pixel connectivity, i.e. all pixels in a connected component share similar pixel intensity values and are in some way connected with each other. Once all groups have been determined, each pixel is labeled with a gray level or a color (color labeling) according to the component it was assigned to.

The connected components labeling in a binary image involves two passes over the image, with an in-between step called equivalence class resolution [12]. The first pass assigns temporary labels to each foreground pixel. The connected components labeling operator scans the image by moving along a row until it comes to a point P (where P denotes the pixel to be labeled at any stage in the scanning process) for which V={1} i.e. it is ON. When this is true, it examines the four neighbors of P which have already been encountered in the scan (i.e. the neighbors (i) to the left of P, (ii) above it, and (iii and iv) the two upper diagonal terms. Based on this information, the labeling of P occurs as follows.
If all four neighbors are 0, assign a new label to P, else if only one neighbor has V= {1}, assign its label to P, else if more than one of the neighbors have V= {1}, assign one of the labels to P and make a note of the equivalences.

A binary image and matrix of labels after first pass is shown in following Fig. 5 as BW and L respectively.

BW

0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	0	1	0	1	1	1
0	1	1	1	0	0	0	1
0	1	0	0	0	1	1	1
0	0	0	0	0	0	0	0

L

0	0	0	0	0	0	0	0
0	0	0	2	0	0	0	0
0	0	0	1	0	3	3	3
0	1	1	1	0	0	0	3
0	1	0	0	0	4	4	3
0	0	0	0	0	0	0	0

Figure 5. A binary image and matrix of labels after first pass

When the first pass is done, a matrix of labels L is generated with an equivalence table containing these pairs.
1 ←→ 2
3 ←→ 4

After completing the scan, the equivalent label pairs are sorted into equivalence classes and a unique label is assigned to each class. As a final step, a second scan is made through the image, during which each label is replaced by the label assigned to its equivalence classes.

In above example temporary labels 1 and 2 map to final label 1, and temporary labels 3 and 4 map to final label 2. A second pass over the output matrix to relabel the pixels according to this mapping will generate final labeled matrix L as shown in Fig. 6.

L

0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	0	1	0	2	2	2
0	1	1	1	0	0	0	2
0	1	0	0	0	2	2	2
0	0	0	0	0	0	0	0

Figure 6. Final labeled matrix after second pass

The MATLAB function bwlabel( ) is used to get 8-connected labeled matrix for a binary image of the word. The elements of L are integer values greater than or equal to 0. The pixels labeled 0 are the background. The pixels labeled 1 make up one object; the pixels labeled 2 make up a second object, and so on.

The number of objects can be found using these labels. Each labeled component is analyzed for the size. If the object size is larger than the threshold size estimated by experimenting (twelve percent of the average size of all components), it is treated as a useful object i.e. a character or a modifier otherwise it is ignored. The useful objects are stored in a cell array for further processing.

It is found that the characters in the middle zone and modifiers in either upper zone or lower zone are connected many times. They are identified as a single labeled component by labeling method. Such connected components cannot be used directly for recognition; hence further segmentation is needed in such cases. To detect whether the detected object is an individual character like 'Ka' or an individual modifier like 'O', or is a combination of connected basic character and modifier(s) like 'Tru' or Ku', a new method is developed based on statistical properties of all objects of the word.

This method helps to determine, whether further segmentation is needed or not. Final segmented results with sufficient size are considered for recognition and the rest are ignored.

The words are extracted from text line, using algorithm described in [14]. Following algorithm is used to extract various components from individual binarized word.

Algorithm

Objective: To extract various components of a word

1) Get the binary image of the word generated by segmenting the preprocessed text line as an input
2) Perform 8 connected labeling
3) Count number of labeled items
4) For each labeled item
   Count for number of ON pixels (non zero elements)
   Store the count for ON pixels into an array SIZES for further processing
5) Find out the maximum, minimum and average value of Labeled elements using SIZES
6) Find out 12 % of the average size of the Labeled elements (to be used as a threshold value to consider the labeled component as a character or modifier)
7) Initialize count for character(component) by value zero
8) For each labeled item
   Check the size of Labeled item with the 12% of average size
   If size of labeled item < 12% of average size
   Ignore the item
   else
   Increment the count for character and store the labeled element in a cell array for further processing
   End

Since bwlabel( ) scans the pixels of input binary image in column order, it finds the object in upper left corner first, then the object in lower-left corner second and so on. As Gujarati text is written from left to right and upper modifiers and lower modifiers are mostly written slightly right aligned above or below the basic characters, extracted labeled items will be mostly in the same order as it is required for recognition of a word.

Sometimes these labeled components may have connected basic character and modifiers, which is identified as a single labeled item. It is observed that it is mostly related to upper modifiers and lower modifiers, especially for the matras of ◌ि ◌ी ◌ु ◌ू. To solve this problem zone boundaries details generated by Algorithm 1 are used for further segmentation of the combination of character-modifier. The further dissected components are again processed to get the separated character and modifiers. The steps for the further segmentation of various components are as described below.

1) Process each labeled component of a word generated using above algorithm. Determine the Bounding Box co-ordinates for each component.

2) Using Bounding Box co-ordinates detected in step 1, determine the starting column and ending columns position for each component. Also determine the starting row and ending row position for each component. Store these values into arrays.

3) Determine the value of minimum and maximum values for the upper left and lower right column and minimum and maximum values for top and bottom row.

4) Determine median for Lower X values (m1), Upper X values (m2) and standard deviation values for the lower X values (s1) and for upper X values (s2).

   range1 = s1 + 0.5; range2 = s2 + 0.5.

5) Use these values to determine whether a component is a connected character and/or modifiers or is an individual modifier or is an individual character.

6) Process each component to determine its characteristics using its upper most and lowermost X coordinate values for an individual component compared with the overall range of the X-coordinates. Use statistical measures for determining whether a component is

   i) A single upper modifier
   ii) A single character
   iii) A connected upper modifier and basic character
   iv) A connected upper modifier, basic character and lower modifier
   v) A lower modifier only
   vi) A connected lower modifier with basic character

   For i1=1:1: x1 % where x1 indicates total number of subcomponents
   6.1) Compute dub(i1) difference between upper left corner X co-ordinate of the component and median for lower X values and dlb(i1) difference between lower left corner X co-ordinate of the component and median for upper X values

   6.2) Set all flags f1...f6 to zero
   if (dub(i1) < (-1) & (dlb(i1) <(-range2)))
   f1=1;
   else if(dub(i1) < (-1) & (dlb(i1)<=2))
   f2=1;
   elseif (dub(i1)>=(-1) & (dub(i1)<=range1))

```
            f3=1;
        end
       if ( dub(i1) < (-1) & (dlb(i1)>=2))
            f4=1;
        elseif (dlb(i1) > range2 )
            f5=1;
        else
            f6=1;
        end
```

6.3) Generate a string – fstr by combining the flags f1,f2…f6.

End

7) Check fstr to determine type of component and necessary action.

If (fstr = '001001')
The component is comprising of only basic character

If (fstr = '100001')
The component is comprising of only upper modifier

If (fstr = '010001')
The component is comprising of connected upper modifier with basic middle zone charcater / modifier like 'iee'
Use the upper zone boundary to dissect the component to get upper modifier and middle character

If (fstr = '000100')
The component is comprising of connected upper and lower modifier with basic middle zone charcater / modifier like 'tru'. Use the upper zone boundary to dissect the component to get upper modifier , lower zone boundary to dissect the lower modifier from the middle character

If (fstr = '001010')
The component is comprising of connected lower modifier with basic middle zone charcater / modifier like 'ku'
Use the lower zone boundary to dissect the lower modifier from the middle character

If ((fstr = '000001') OR (fstr = '010100'))
The component is comprising only basic character in some cases or flat character with some modifiers

If (fstr = '000010')
The component comprises only of lower modifiers

8) Process further to detect more than one component in a dissected character using labeling approach as suggested above and consider only sufficiently large component as a modifier or base character. Store individual character and modifier in cell array for recognition.

7. RESULTS

The results of segmentation of the word into modifiers and basic characters/conjunct characters are satisfactory. The test data set comprises of 250 words and the results of words segmentation are as shown in Table 1.

Table 1 - Results of zone detection algorithm for 250 words

Total No. of words 250	Correctly detected Upper zone	Correctly detected Middle zone	Correctly detected lower zone
	188	209	211
% Accuracy	75	84	84

It is observed that if slanted characters or modifiers are present in a word then the accuracy of zone detection algorithm is reduced. It is observed that presence of characters like ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ' or abnormally large size of modifiers also causes improper zone detection. Abnormal size of modifiers, handwritten word having too small character size and lack of inter character alignment are some other causes for poor performance.

The algorithm to extract components from a word is tested on 500 different words written by different persons. The results are as shown in Table -2.

Table 2 - Extraction of characters and modifiers from handwritten Gujarati words

Words with characters ર, ુ, ા, ૂ and connected characters		**Words without characters ર, ુ, ા, ૂ and connected characters**	
500 words		**364 words**	
Correct extraction of characters and modifiers	**Incorrect extraction of characters and modifiers**	**Correct extraction of characters and modifiers**	**Incorrect extraction of characters and modifiers**
300 words	200 words	300 words	64 words
60%	40%	82.4%	17.6%

It is observed that the characters and modifiers were extracted correctly for 300 words. The words comprising of characters ર, ુ, ા, ૂ and connected characters are the major cause for incorrect segmentation of words. Out of 500 words 136 words were having all or any of the character ર, ુ, ા, ૂ. If the over segmentation due to these characters is ignored, the proposed algorithm gives accuracy of 87.4% on our set of words. The accuracy level is higher for the printed words.

8. CONCLUSION

It is found that the characters with multiple subcomponents and with either or all of characters ર, ુ, ા, ૂ are the major cause for over segmentation resulting into inaccuracy in word segmentation. It is thus necessary to combine the over segmented characters at letter stage to form a single character for recognition. In literature an approach based on Neural network is found for English handwritten text as well as for some languages of Indian origin like Assamese where individual characters are separated out from the text line by over segmenting the entire line. Each of the segments thus obtained, next, is fed to the trained ANN. The point of segmentation at which the ANN recognizes a segment or a combination of several segments to be similar to a handwritten character, a segmentation boundary for the character is assumed to exist and segmentation performed. The segmented character is next compared to the best available match and the segmentation boundary confirmed. The similar approach can be derived for such over segmented characters of Gujarati words.

9. REFERENCES

[1] Dr. Bholanath Tiwari ,"Bhash Vigyan", 8th edition 1971, Publisher: Kitab Mahal, Allahabad, India.

[2] S. Antani, L. Agnihotri, "Gujarati character recognition", Proc. of International Conference on Document Analysis and Recognition , pp. 418–421,1999.

[3] U. Pal, B.B. Chaudhuri, "Automatic Separation of Machine- Printed and Hand-Written Text Lines", Proc. of International Conference on Document Analysis and Recognition , pp. 645-648, 1999.

[4] Veena Bansal, R.M.K Sinha , "A Complete OCR for Printed Hindi Text in Devanagari Script", Proc. of International Conference on Document Analysis and Recognition, pp. 800-804, 2001.

[5] Jignesh Dholakia ,Atul Negi, S. Rama Mohan, "Zone Identification in the Printed Gujarati Text", Proc. of International Conference on Document Analysis and Recognition, Vol.1, pp.1520-5263, 2005.

[6] B.B. Chaudhhuri, U. Pal, M. Mitra, "Automatic recognition of printed Oriya script", Saadhanaa 27 (1), pp. 23–34, 2002.

[7] Md. Abul Hasnat , Mumit Khan, "Rule Based Segmentation of Lower Modifiers in Complex Bangla Scripts", Proceedings of the Conference on Language & Technology , pp.94-101, 2009.

[8] Casey R.G., Lecolinet E., "A survey of methods and strategies in character segmentation", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 18, Issue 7, pp. 690–706, 1996.

[9] S.M. Mahmud, N. Shahrier, D. Hossain, T. M. Chowdhury, M.A. Sattar, "An Efficient Segmentation Scheme for the Recognition of Printed Bangla characters", Proc. of ICCIT, pp. 779-781, 2003.

[10] U. Pal, B. B. Chaudhuri, "OCR in Bangla: an Indo-Bangladeshi Language", Proc. of ICPR, pp. 269-274, 1994.

[11] B. B. Chaudhuri, U. Pal, "An OCR System to Read Two Indian Language Scripts: Bangla and Devnagari(Hindi)", Proc. of International Conference on Document Analysis and Recognition, Vol.2, pp.1011 -1015, 1997.

[12] Haralick and Shapiro, Computer and Robot Vision, Vol. I, Addison-Wesley, 1992.

[13] Chhaya Patel, Apurva Desai, "Zone Identification for Gujarati Handwritten Word", Second International Conference on Emerging Applications of Information Technology, pp.194-197,2011. IEEE DOI 10.1109 /EAIT.2011.47, 2011.

[14] Patel C., Desai, A., "Segmentation of text lines into words for Gujarati handwritten text" ,Proc. of International Conference on Signal and Image Processing (ICSIP)-2010, pp. 130 - 134, Print ISBN: 978-1-4244-8595-6, DOI 10.1109/ICSIP.2010.5697455 , 2010.