International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 06 Issue: 03 | Mar 2019 www.irjet.net p-ISSN: 2395-0072

Classification of Hindi Maatras By Encoding Scheme

Ashna Joysi Gaikwad[1], Dr. Neeta Nathani[2]

[1]Research Scholar, Gyan Ganga College Of Technology, Jabalpur, MP, India
[2]Assistant Professor, Gyan Ganga College Of Technology, Jabalpur, MP, India

---

Abstract - *Hindi vowels are appended to the consonants in the form of modifiers or maatras. The modifiers may be present at top or bottom level resulting in top modifiers or ascenders and lower modifiers or descenders. For an optical recognition system these modifiers are firstly segmented then classified. The proposed method is a novel approach that classifies the modifiers with the help of codes. Each modifier is segmented from the character then is processed with encoding steps. The end product of the encoding steps result in a distinctive code for each modifier. The proposed method classifies the modifiers easily with less space and time by using the structural property of each modifiers. The method has been tested on printed and handwritten modifiers and has found to yield 95.23% and 90.47% accuracy for descenders and ascenders respectively.*

Key Words: **Hindi character segmentation, Hindi Modifier segmentation, Optical character recognition, Pattern classification,Hindi Modifier Classification**

1.INTRODUCTION

Hindi language is a part of Devnagri script consisting of 33 consonants (vyanjans) and 13 vowels or swaras. Segmentation and classification of Hindi character is easy due to the blank space present between the characters but the presence of modifiers make the segmentation process complex. For classification either the modifiers as a whole were used or in the form feature vectors. In either case the OCR system becomes complex[1]. Some modifiers have similar shapes above the header line or shiroreka irrespective of the position of vertical bar along the characters. Hence to simplify the segmentation and classification of modifiers, the modifiers are converted to codes directly, a particular code for each modifier. The distinctive codes are sufficient in themselves to classify the modifiers without an actual classifier.

The rest of the paper is organized as : Section 1 introduces the complexity of modifiers and the need for its classification. This section is elaborated in Section 2. Section 3 deals with existing techniques for modifier segmentation and the proposed method is described in Section 4. Section 5 includes results and related discussions and Section 6 concludes the paper.

2. HINDI LANGUAGE AND THE MODIFIERS

The modifiers are integral part of Hindi language. Hindi is written in Devnagri script which has a horizontal bar over every word which is known as header line. The structure of a Hindi word is distributed in zones: top zone, middle zone and lower zone. This can be seen by figure 1.

Fig -1: Zones of a Hindi Word

The top zone is completely reserved for the upper modifiers and the lower zone is reserved for the lower modifiers or descender.

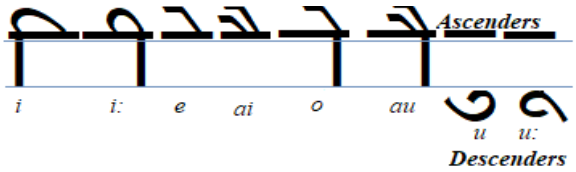

Fig -2: Modifiers

Figure 2 shows the positioning of each modifiers. Even though there are other modifiers present along with some special symbols this work deals with the above modifier which are commonly used. It must be noted that in case of ascenders there are some modifiers which are similar in upper zone unless their bar in middle zone is taken. For example *e* and *o* from figure 2 are same except the middle part of *o*. Such arrangements of modifiers make them robust to various segmentation process. If projection profile[2] is considered, the method derives only the upper part of the ascender which results in miss classification. Hence a new method is used to solve this problem.

3. LITERATURE REVIEW

The commonly used method for modifier segmentation is projection profile. This technique uses the density of object pixels in a region. This method was used by Garg, Kaur and Jindal in [3]. The removal of header line removes the ascenders from middle region. The modifiers present in the top zone are separated by spaces. Using these blank spaces the modifiers are derived from the word. For descenders the mean height of the character is used as a delimiter.The above method gave accurate result but there was no

provision for middle bars of the ascenders. Bag and Krishna in [4] suggest that after removal of header line the appropriate joining from the modifier to their middle bar can be done.This was achieved by using the width of the character and middle bar. As usual the middle bar is the thinnest object in the middle zone this was joined to their respective modifiers. For descender segmentation the character with greater height is taken and by using a 3 by 3 window the descender is derived.

4. PROPOSED METHOD

The proposed method deals with the segmentation as well as classification of the modifiers. Due to the positioning of the ascenders and descenders two separate algorithm are used.

4.1 ALGORITHM FOR ASCENDER CLASSIFICATION

There are three encoding levels for ascender classification each one giving a single value.

Level 1: Input: Character with ascender

i) Remove the header line

ii) Store ascender and middle portion in separate cells

iii) Find the number of objects present in the middle portion, if only one character is present (as in modifier *e* and *ai*) then Level 1 code = 3 , end the process.

iv) If more than one objects are present(a character and a middle bar) then find width of each object.

v) The middle bar will always be thinnest. Find the position of the thinner object. If it is present before the character (as in modifier i) then Level 1 code = 1, else it is present beyond the character then Level 1 code = 2.**End**.

The graphical representation of level 1 encoding with an example is shown if figure 3.

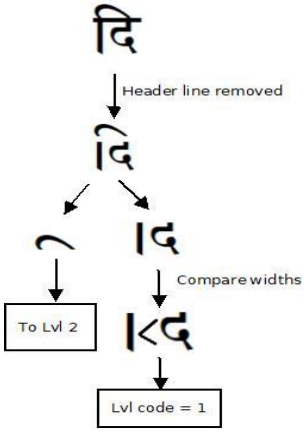

Fig -3: Level 1 Algorithm

Level 2: Input: Upper modifier

i) The modifier stored in a separate cell is firstly skeletonize[5].

ii) The middle row from the skeletonized modifier is derived,

iii) Number of object pixels are counted in middle row

iv) If only one object pixel is found then modifiers *e* and *o* are present(figure 2). For such modifiers Level 2 code = 1 is assigned.

v) for the rest of modifiers Level 2 code = 2 is assigned, **End**.

For modifier *o* only one object is obtained in selected window in figure 4 so code obtained is 1 where as in modifier *i* two objects are found.

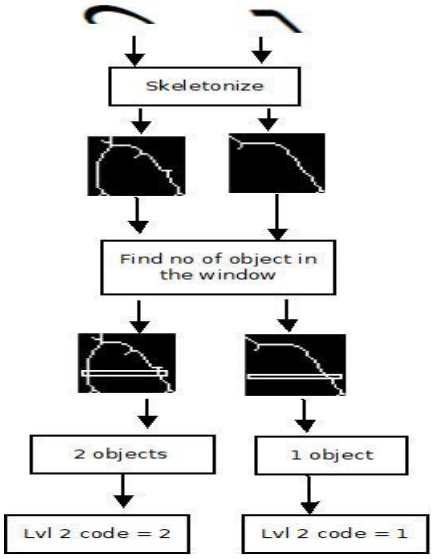

Fig -4: Level 2 Algorithm

Level 3: Input: Skeletonized Modifier

i) The portion of skeletonized modifier which is last five rows and first three columns is taken,[6]

ii) Object pixels will be present for modifiers with both the ends joined to header line.

iii) If object pixels are present then Level 3 code = 1 is assigned else Level 3 code = 0 is assigned.**End**

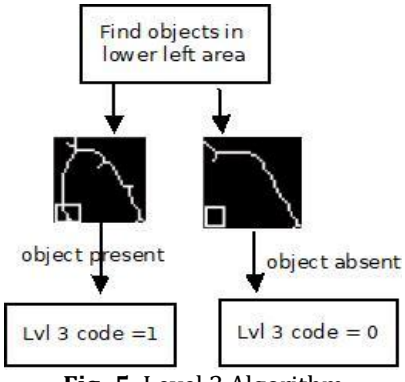

Fig -5: Level 3 Algorithm

Figure 5, shows the codes of modifiers *i* and *e*. So the encoding scheme for upper modifier can be tabulated in Table 1.

Table -1: Encoding Scheme of Ascenders

Conditions	Codes					
	क	की	ऊ	ऋ	ए	ऐ
Level 1						
Single Object under ascender	-	-	3	3	-	-
Position of thin bar	1	2	-	-	2	2
Level 2						
Objects in middle bar	2	2	1	2	1	2
Level 3						
Object in lower left corner	1	1	0	0	0	0
Codes	**121**	**221**	**310**	**320**	**210**	**220**

4.2 ALGORITHM FOR DESCENDER CLASSIFICATION

The descenders are only present in lower zone. These can be derived by the height of the character. The character with descender is taller than the rest of the characters. Hence using this information the descender is segmented from the character.
The method covers the following descenders shown in figure 6:

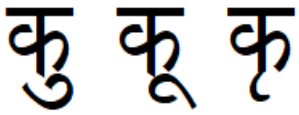

Fig -6: Lower Modifiers

Level 1: Input : Segmented Descender
(i)Find the width of the descender and calculate the threshold width, Wth = Width * 0.7,
(ii)Find the last object pixel in the center row,
(iii)Compare the threshold width with number of last object pixel obtained in (ii),
(iv)If Wth < last pixel then the right side of the descender is empty , only *ra* modifier of figure 6 fulfill this condition , so Level 1 code = 1,
(v)Conversely if Wth > last point then Level 1 code = 0 is assigned. **End**

Fig -7: Width of Descender

The blue line in figure 7 shows the middle row of each modifier ending at last object pixel. The black line under each descenders shows the actual width. Comparison between the width of middle row and width of descender the code is assigned.

Level 2: Input : Segmented Descender
For modifiers *u* and *u:* , another level of code is derived. The space at the lower right end is considered.
From this region the presence of object is checked. If object is present then Level 2 code = 9 is assigned else 1 is assigned.

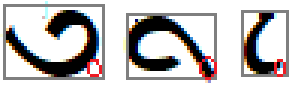

Fig -8: Presence of object in lower right end

From all three modifiers in figure 8 only *u* has no objects in lower right end. This difference is used to assign different character.

Table -2: Encoding Scheme of Descenders

Condition	Codes		
Level 1	ु	ृ	ू
Threshold width > last pt.	0	0	1
Level 2			
Object in lower right end	1	9	9
Codes	**01**	**09**	**19**

5. RESULT AND DISCUSSIONS

The proposed method was applied on Hindi modifiers from handwritten as well as printed modifiers. The proposed method is implemented using MATLAB and the findings can be tabulated. Following figures shows the implemented output for both the modifiers. Since each modifier has a different code, this method can be used for classification also.

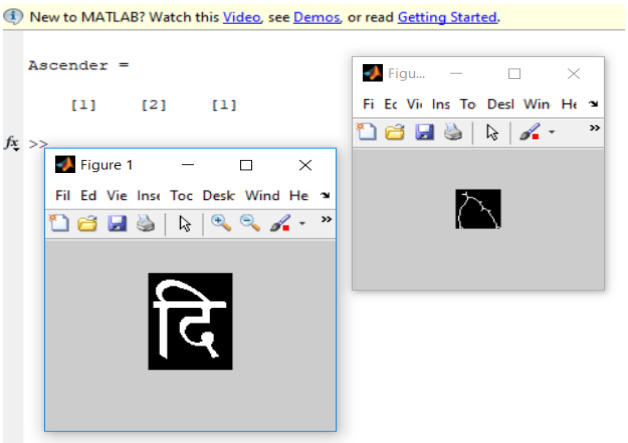

Fig -9: Encoded Output for Ascender

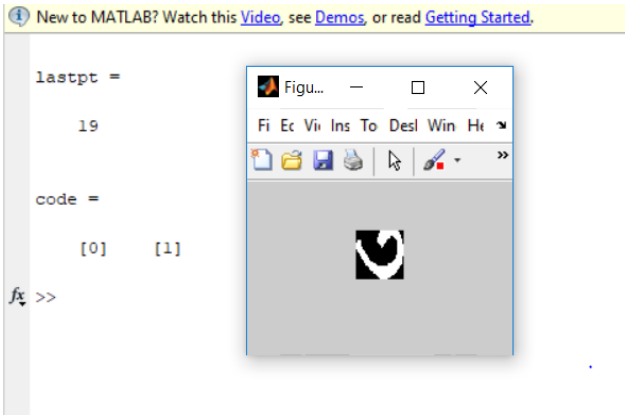

Fig -10: Encoded Output for Descender

The proposed method was tested on 7 different texts containing each of the specified Hindi modifiers with words. The codes obtained from the method were validated by assigning true mark (T) for correct ones and false mark (F) for the incorrect. The outcome is tabulated as follows:

Table -3: Classification Accuracy of the Modifiers

Modifier	Document							Encoded
	1	2	3	4	5	6	7	
ु	T	F	T	T	T	T	T	6/7
ू	T	T	T	T	T	T	T	7/7
ृ	T	T	T	T	T	T	T	7/7
ॄ	T	T	T	T	T	T	T	7/7
ॅ	T	T	T	T	T	T	T	7/7
े	T	F	T	T	T	T	T	6/7
ै	F	T	T	T	T	T	T	6/7
ॉ	T	F	T	T	T	T	T	6/7
ो	F	T	T	T	T	T	T	6/7
Classification Accuracy								**92.06%**

The classification accuracy for descender and ascender is 95.23% and 90.47% respectively. Overall accuracy is 92.06%

Some modifiers like ु (*u*) and े (*e*)had bigger loop, as a result they are miss classified. The skewed strokes of ौ (*au*) make it appear joint and thus yielding incorrect code.

6. CONCLUSION

The proposed method is a novel approach for classification of the Hindi modifiers. The method converts each modifier image into a distinctive code directly . As each modifier has a different code this method can be used as a classification method without feature extraction.

The encoding algorithm contains different tiers for each level of codes. The algorithm can be used in the combination of codes and images as well.
But using this scheme it must be noted that the modifiers must not be distorted, noisy or irregular.

7. FUTURE SCOPE

The method can be expanded to classify the modifiers of other Devnagri languages. This technique can be improved to deal with the distorted and irregular shaped modifiers.Using this approach the Hindi characters may be classified .

REFRENCES

[1]Garg N.K,Kaur L, Jindal MK. The Hazards in Segmentation of Handwritten Hindi Text[J]. Journal of Computer Application,2011,29(2):0975-8887

[2]Anupama M, Rupa CH, Reddy ES.Character Segmentation for Telgu Document Using Multiple Histogram Projection[J]. Journal of Computer Science and Technology Graphic and Vision,2013,13(5):0975-4350

[3]Garg N.K,Kaur L, Jindal MK.Segmentation of Handwritten Hindi Text[J]. International Journal of Computer Application,2010,01(4):0975-8887

[4] Bag S. and Krishna A.,Character Segmentation of Hindi Unconstrained Handwritten Words Springer International Publishing, LNCS,2015, 9448:247-260

[5]Nixon M, Aguado A. Feature Extraction and Image Processing, 2nd Edition,ELSEVIER

[6]Lempitsky V, Kohli P, Rother C, Sharp T[R]. Image Segmentation with Bounding Box Prior, Microsoft Research, Cambridge,2013

[7]Puneet, Garg NK. Binarization Technique Used for Grey Scale Image[J]. International Journal of Computer Application, 2013,71(1):0975-8887

[8]Nixon M, Aguado A. Feature Extraction and Image Processing, 2nd Edition,ELSEVIER.