OCR of Printed Telugu Text with High Recognition Accuracies

C. Vasantha Lakshmi, Ritu Jain, and C. Patvardhan

Dayalbagh Educational Institute
Agra – 282005, India
cvasantha@rediff.com, rituritu2006@rediff.com,
cpatvardhan@hotmail.com

Abstract. Telugu is one of the oldest and popular languages of India spoken by more than 66 million people especially in South India. Development of Optical Character Recognition systems for Telugu text is an area of current research.

OCR of Indian scripts is much more complicated than the OCR of Roman script because of the use of huge number of combinations of characters and modifiers. Basic Symbols are identified as the unit of recognition in Telugu script. Edge Histograms are used for a feature based recognition scheme for these basic symbols. During recognition, it is observed that, in many cases, the recognizer incorrectly outputs a very similar looking symbol. Special logic and algorithms are developed using simple structural features for improving recognition accuracies considerably without too much additional computational effort. It is shown that recognition accuracies of 98.5 % can be achieved on laser quality prints with such a procedure.

1 Introduction

During the past few decades, substantial research efforts have been devoted to Optical Character Recognition (OCR) [1,2]. The object of OCR is automatic reading of optically sensed document text materials to translate human-readable characters into machine-readable codes. Research in OCR is popular for its application potential in banks, post-offices and defense organizations. Other applications involve reading-aid for the blind, library automation, language processing and multi-media design [3].

Commercial OCR packages are already available for languages like English. Considerable work has also been done for languages like Japanese and Chinese [1]. Recently, work has been done for development of OCR systems for Indian languages. This includes work on recognition of Devanagari characters [4], Bengali characters [5], Kannada characters [6] and Tamil characters [7]. Some more recent work on Indian languages is also reported [8,9,10,11,12].

Telugu is one of the popular languages of India that is spoken by more than 66 million people especially in South India. Work on Telugu character recognition is not substantial [13,14]. Vasantha Lakshmi et al. [17] have recently reported the development of a Telugu OCR System for Printed text (TOSP) based on identification

P. Kalra and S. Peleg (Eds.): ICVGIP 2006, LNCS 4338, pp. 786 – 795, 2006.

of symbols defined as Basic Symbols by them. A Basic Symbol is a single connected entity in Telugu script and is treated as the unit of segmentation. The system works in three steps as described. Recognition rates of over 97% have been reported over a wide variety of fonts and sizes. Their approach is essentially a feature based approach where features of all the basic symbols in several different fonts and sizes are stored and symbols of same fonts but different sizes are recognized on the basis of these features. The features used by them are the local gradients at various pixels called the Radial Direction Features [17, 18].

Recent work in the context of MPEG-7 features has shown the utility of Edge Histograms to aid the recognition in various image processing applications [22]. These are utilized in this work. Further improvement in recognition accuracies is achieved by identifying pairs of symbols that are frequently confused for each other. The logic for determining the correct basic symbol and the results of OCR before and after incorporation of this logic are given.

The sets of symbols that are confused for each other are a characteristic property of the script. They are confused for each other because they are similar. These sets remain more or less same irrespective of the feature extractor and recognizer that operates over the whole set of Basic Symbols. Therefore, many of the ideas presented in this paper could be used with advantage in improving the recognition accuracy with any OCR system for Telugu. Another important point is that Confusion Logic is called into play only when one of the confusing symbols is recognized in order to verify or contradict it. Further, simple features are used to resolve the confusion. Therefore, this does not add substantially to the computational requirements.

The rest of the paper is organized as follows. The approach adopted in this work is presented briefly in section 2. In section 3, the Confusion Table is presented. Detailed logic for resolving the confusion in each set is presented in section 4. Results of OCR after incorporation of additional logic are presented in section 5. Some conclusions and pointers toward future work are highlighted in section 6.

2 Recognition of Printed Telugu Text

The recognition works by isolating and recognizing Basic Symbols. Basic Symbols are connected regions in the image. If a modifier is physically attached to the character it modifies, they together constitute a single basic symbol. It has been shown [17] that such an approach is extremely useful in reducing the number of symbols that the recognizer has to deal with to manageable levels (around 400) from the lakhs of combinations of characters and modifiers possible. Therefore, the task in segmentation is to isolate such basic symbols.

The processing starts with the conversion of the gray scale image of a page of text into a binary image using thresholding. Any small blobs introduced due to scanning noise are removed to clean the image.

The actual basic symbol is represented by black pixels and background is represented by white pixels. Any skew in the image is detected and removed using a modified Hough transform method adapted for Telugu text [19].

Table 1. Steps in recognition of printed Telugu text

1. Conversion of a gray scale image of input text to binary image.
2. Image rectification
3. Skew detection and its removal
4. Separation of text into lines, words, and basic symbols.
5. Preliminary classification using size property for each basic symbol.
6. Computation of Edge Histogram Features for each basic symbol.
7. Recognition by means of Nearest Neighbour (NN) classifier.
8. Output.

Table 2. Confusion Table depicting Symbols that are confused for each other

S. No.	Element 1 of Confusion Set		Other Element(s) of Confusion Set	
	Phonetic English	Telugu symbol	Telugu symbol	Phonetic English
1	/pa/	ప	న	/sa/
2	/va/	వ	న	/na/
3	/gha/	ఘ	ను	/su/
4	/ma/	మ	ను	/nu/
5	/ra/	ర	ల	/la/
6	/la/	ల	ట	/Ta/
7	/lu/ /lU/	లు లూ	ట టా	/Ta/ /TA/
8	/cha/	చ	వ	/va/
9	/vA/	వా	ా	/ha/
10	/da/	ద	ఉ డ	/u/ /Da/
11	/ri/	ఱి	ి	/imatra/
12	/lu/	లు	యి	/yi/

Profiling is used to segment the text image into lines and words. This is done taking advantage of the spacing between lines and between words. In every word, each basic symbol is identified by determining connected components. For each basic symbol, a preliminary classification scheme is implemented on the basis of the

relative sizes of the symbols. All the symbols are converted to size corresponding to 36 columns and the row sizes for this column size is used to classify the basic symbols into 14 different sets.

The features used are the Edge Histogram features. The database is created with three popular fonts i.e. Harshapriya, Godavari and Hemalatha [21] and three different sizes i.e. 25, 30 and 35. The feature vectors are divided into 14 sets as described above and stored in the database. The algorithm for feature extraction is represented succinctly in the following pseudo-code.

1. For each word in every line of a scanned printed page of Telugu text,
2. Isolate the next basic symbol from the word as given above.
   Repeat steps 3 to 10 for each basic symbol.
3. Obtain the bounding box eliminating the blank surrounding space.
4. Partition the bounding box into N1 x N2 blocks. In this work N1 and N2 are taken as 4 each.
5. Determine the edges of the symbol using the Canny edge operator.
6. Calculate the gradient magnitude and direction at each pixel location on the edges within each block.
7. Quantize the edge directions into K ranges. K=9 in this work i.e. 0-20, 20-40,…, 160-180. Directions 180-360 are mapped again onto 0-180 range of directions.
8. Calculate the adaptive threshold of gradient magnitude and perform thresholding to obtain the new threshold gradient direction at each pixel location.
9. Calculate the relative edge histogram by dividing the edge direction values in Step 6 by total number of pixels in that block.
10. Concatenate the feature vectors from all the partitions to obtain the complete feature vector.

The OCR of a text page begins with the scanned image. The segmentation steps described above are performed on this image to isolate the image of each basic symbol. The feature vector of the symbol to be recognized is computed as given above. This is then provided to the recognizer. The recognizer uses a preliminary classification scheme and a Nearest Neighbour (NN) classifier scheme on the feature vectors stored in the database to identify the basic symbol.

The preliminary classification classifies a basic symbol based on its height into one of the 14 different sets. The classifier considers only the basic symbols in the set identified by the preliminary classification scheme. This results in a considerable saving in the computational expense. However, it does not result in a degradation of the recognition performance. The process is repeated till all the basic symbols are recognized.

This approach has been implemented and tested over a variety of images with different fonts i.e. Harshapriya, Hemalatha and Godavari and sizes 15, 18, 20, 23, 25, 28, 30, 32, 35. Recognition accuracy by directly using the above scheme is 95.4 % as presented in Tables 3, 4, and 5. This recognition accuracy needs to be improved further for actual use.

In computing this recognition accuracy, if a particular basic symbol appears three times in the text and it is mis-recognized all the three times it is taken as three errors

Table 3. Raw and improved results on DS1 to DS9 with Hemalatha font

Data Set #	Type	Size	# of BS	NN	
				Initial Recognition Ratio	Recognition ratio with additional logic
DS1	Hm	15	127	0.937	0.976
DS2		18	127	0.969	0.992
DS3		20	127	0.969	1.000
DS4		23	127	0.969	0.992
DS5		25	127	0.953	0.976
DS6		28	127	0.945	0.992
DS7		30	127	0.937	0.961
DS8		32	127	0.945	0.992
DS9		35	127	0.953	0.984
TOTAL			**1143**	**0.953**	**0.985**

Table 4. Raw and improved results on DS10 to DS18 with Harshapriya font

Data Set #	Type	Size	# of BS	NN	
				Initial Recognition ratio	Recognition ratio with additional logic
DS10	Hr	15	127	0.921	0.976
DS11		18	127	0.945	0.992
DS12		20	127	0.969	1.000
DS13		23	127	0.929	0.992
DS14		25	127	0.945	0.984
DS15		28	127	0.961	0.992
DS16		30	127	0.969	1.000
DS17		32	127	0.969	1.000
DS18		35	129	0.946	0.969
TOTAL			**1145**	**0.950**	**0.990**

and not one. So actual misrecognized symbols are even lesser than what this number indicates. This also provides the motivation for deeper analysis into why a particular symbol is misrecognized every time it appears because if logic can be found to rectify this then at one stroke several of the errors could be eliminated. Such an effort is made in the next section.

3 The Confusion Table

As mentioned above, an analysis of the results shows that some symbols are often recognized incorrectly. The reason for mistakes in recognition can be scanning noise, defect in the paper where the symbol is printed that leads to extra dark pixels, spread of ink on the paper etc. These are, however, random causes and cannot be the reason

Table 5. Raw and improved results on DS19 to DS27 with Godavari fon

Data Set #	Type	Size	# of BS	NN classifier	
				Initial Recognition ratio	Recognition ratio with additional logic
DS19	Go	15	126	0.976	0.984
DS20		18	127	0.961	0.984
DS21		20	128	0.961	0.984
DS22		23	129	0.938	0.977
DS23		25	127	0.953	0.984
DS24		28	127	0.976	0.984
DS25		30	127	0.969	0.984
DS26		32	127	0.945	0.969
DS27		35	127	0.953	0.984
TOTAL			**1145**	**0.959**	**0.982**

for consistent wrong recognition of a particular symbol as another symbol. It is observed that, in many cases, a symbol is recognized erroneously because the recognizer incorrectly outputs a very similar looking basic symbol. The low level features that are used in the recognition process are not able to distinguish between the two. Recourse is taken, therefore, to higher level structural features that can provide the distinction.

The sets of similar symbols are arranged in the form of a table with each row corresponding to a set. This is referred to as the Confusion Table [Table 2]. Though the symbols in each set of the Confusion Table look very similar, on closer observation, it is seen that each basic symbol has some unique feature that distinguishes it from the other(s) in the set. This feature is identified and made use of in correctly identifying it.

4 Resolution of Confusions

In this section, an attempt is made to identify a distinguishing feature that can be used to distinguish between the elements of each set. This is non-trivial because the distinction may be very fine, especially for the smaller sized characters.

4.1.1 Confusions 1 to 4

The confusion among the first four row entries in the Table 2 are resolved using a similar logic. This is possible because there is no possibility of confusion across sets reported in different rows. It is observed from column 3 of the table for all these 4 entries that there is a small closed loop at the bottom near the left end for these symbols. However, as is seen in column 4 for the corresponding entries, the characters with which they are confused are open near the bottom left end. An algorithm is designed to detect the presence or absence of the closed loop. Care is taken to ensure that the logic works for a variety of fonts and sizes.

Figure 1 shows the images of two Telugu characters /pa/ and /sa/. Long lines in these figures divide each of the images into two halves i.e. the top half and the bottom half therefore, only the bottom half of the image is considered for resolving the confusion. The figure also shows a short line inside the loop of /pa/. The ends of this line, denoted by 'a' and 'b', indicate the beginning and end of the possible loop in the image. The point 'a' is the left most pixel found in the bottom half of the image**.** The logic for determining the existence of loop is given succinctly in Algorithm 1.

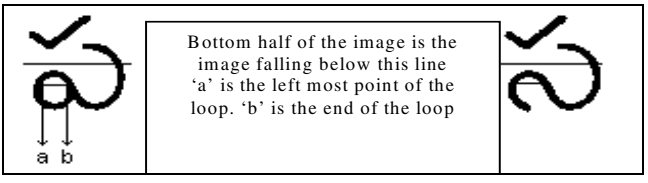

Fig. 1. Images of Telugu characters /pa/ and /sa/

Although the algorithm is explained for the pairs reported in the table only, the same can also be used for resolving confusion among the pairs generated by attaching same modifier symbols.

Algorithm 1

1. Split the image into two halves. Consider only the bottom half of the image for further processing.
2. Find the left most pixel position 'a'. Let 'i' denote its row number and 'j' the column number.
3. By moving along the row 'i' to the right from the pixel 'a' identify the pixel at position 'b' as shown in Figure 1.
4. For each of the pixels between 'a' and 'b' scan down to the bottom row of the image starting from row 'i'. In this process, if for any column, a foreground pixel is not encountered at all, the element belongs to column 4 of Table 2. A single gap is sufficient as the characters are thick enough.
5. If no gap is found for any column, the symbol belongs to column 3 of the same Table.

4.1.2 Confusion Between /ra/ and /la/ Families of Consonants

These form the fifth row of Table 4. The /ra/ and /la/ families are quite similar because attaching modifiers to each of these, i.e., /ra/ and /la/, results in similar compound characters. The concept of Zero Crossings (ZC) is made use of in differentiating between these two basic symbol families. A ZC is a transition from a character or foreground pixel to a background pixel or vice-versa. The images of /ra/ and /la/ consonants are identical near the bottom and differ near their top ends. /la/ has a loop at the top left end whereas /ra/ has a tick mark. The images of /la/ and /ra/ are shown in Figures 2(ii) and 2 (iii) respectively.

For differentiating between these two or any pairs in their families, the bottom most pixel of the image, labeled as 'a' in Figure 2 (i) is taken as the starting point.

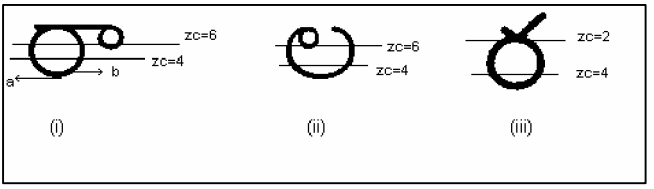

Fig. 2. Images of basic symbols (i) rA (ii) la

The thickness of the character starting from the bottom most point is avoided in searching for ZC. For this purpose, the pixel labeled as 'b' in Figure 2 (i) is identified by scanning up in the same column as 'a' and stopping at the last foreground pixel among the continuous foreground pixels. From here onwards a count of ZC is noted for every row starting with the row above this row. The ZC count for both the consonants /ra/ and /la/ is 4 for all the rows in the beginning. This is clearly indicated by the line ZC=4 in Figure 2. The ZC count for consonant /la/ increases to 6 and that for /ra/ decreases to 2, as the rows above 'b' are scanned.

The same logic is also applicable for the members of the two consonant families generated by attaching the vowel modifiers of /i/, /I/, /e/, /E/, /AW/ on the two consonants. However, this algorithm does not work as such for the members of /ra/ and /la/ families that are formed by attaching the vowel modifiers of /A/, /o/ and /O/. These compound characters assume shapes that contradict the logic given above.

Consider the image of consonant /ra/ modified by the vowel modifier of /A/ shown in Figure 2(i). For this image a loop is encountered and the corresponding ZC count is 6. Therefore, as per the above logic, it will be treated as /lA/ and not /rA/, as the ZC count is greater than 4. However, it is observed that in this case the ZC count increases because of the loop in the right portion of the compound character and not because of a loop in the left portion as in /la/. Similar is the case with modifiers of /o/ and /O/. Therefore, the logic given above is suitably modified to work for pairs of symbols generated by all the vowel modifiers. The ZC count is not taken for all the columns. Only, the columns in which the left side loop is possibly encountered are considered. With this modification the above logic is useful for distinguishing between all the members of the two families. Since noise has already been removed for small specks and isolated larger ones would be recognized as separate characters, zero crossings do not create any problem.

Algorithm 2

1. Find the bottom most pixel 'a' of the image, as shown in Figure 2(i).
2. Find the pixel marked as 'b' in the same figure that is in the same column as 'a'.
3. Let startrow be the row above the row in which 'b' is located.
4. Consider the left portion of the image from startrow upwards.
5. For startrow, find ZC. Store it as previous ZC.
6. For each row above startrow, find ZC.
   if ZC >previous ZC, id=/la/ else if ZC<prevZC, id=/ra/ else continue.
7. end.

Similarly, logic based structural differences is designed for distinguishing between the rest of the sets of basic symbols.

5 Computational Results

OCR experiments were carried out on a number of data sets of laser quality prints containing text in three fonts and nine different sizes. Results of OCR experiments on 27 data sets DS1 to DS27 without additional improvement logic for Confusing Symbols were analyzed and the correction logic incorporated for rectifying the commonly occurring errors. An important point to be noted is that the correction logic for a particular pair of confusing symbols is executed only when the recognizer recognizes one of the symbols in the set as being present in the image under consideration. Thus, the computational burden is not enhanced too much with the inclusion of this logic. Still, the correction of the commonly occurring errors enhances the recognition rate. Results of OCR experiments are presented on the same data sets DS1 to DS27 in Tables 3 to 5 but this time with the incorporation of the logic described above for the correction of errors due to the presence of confusing symbols. Edge Histogram features are still used. Recognition accuracies are enhanced in all cases going up to 99 % in the case of Harshapriya font. Overall recognition accuracy in the case of all the fonts is improved from 95.4% to 98.5 %.

6 Conclusion

This paper is concerned with achieving better recognition rates in OCR of printed Telugu text by use of Edge histogram features and additional logic for resolving confusion among similar symbols. Simple structural features are utilized to improve recognition accuracies. Incorporation of this logic does not add too much to the computational requirements. This is in direct to contrast to the more computationally intensive dictionary matching schemes. But, still, recognition accuracies show considerable improvement. The logic presented in this paper can be incorporated to improve recognition accuracy with any OCR system for Telugu, as it is quite general and works with different fonts and sizes i.e. the structural features used to aid resolution of confusion are font independent. The approach is novel as it utilizes simple structural features instead of commonly employed complicated dictionary matching procedures towards the same result.

Work is being pursued in improving the recognition accuracies further by incorporation of additional post-processing logic based on the frequency of association of symbols that are found together in the text. For example, it is known that the some modifiers occur very frequently with some characters and some modifiers occur very infrequently. Development of better feature sets by better choice of features is also another direction being pursued.

References

[1] G. Nagy, Twenty years of Document Image Analysis in PAMI, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000, 22(1), pp 38-63.

[2] S. Mori, C. Y. Suen, K. Yamamoto, Historical review of OCR Research and Development, Proc. of the IEEE, 1992, pp 1029 - 1058.

[3] V.K. Govindan, A.P Shivaprasad., Character recognition -- A review, Pattern Recognition, 1990, 23(7), pp. 671-683.
[4] V. Bansal, R. M.K Sinha, A survey of OCR in Indian Languages and a Devanagari OCR scheme, in Proceedings of the STRANS – 01, IIT, Kanpur, 2001.
[5] B.B. Chaudhuri, U.Pal, A complete printed Bangla OCR system, Pattern Recognition, 1998, 31, pp 531-549.
[6] P. Nagabhushan, A. Radhika, Improved region decomposition method for the recognition of non-uniform sized characters, in Proceedings of the International Conference on Cognitive science, ICCS-97, New Delhi, 1997, Vol. 1, pp. 36-42.
[7] S. Anna Durai, et al., Tamil character recognition using multilayer neural network, in Indian Conference on Pattern Recognition, Image Processing and Computer Vision (ICPIC), 1995., pp. 155-160.
[8] A. Bishnu, B. Chaudhuri, Segmentation of Bangla Handwritten text into characters by recursive contour following, in Proceedings of the Fifth International Conference on Document Analysis and Recognition, ICDAR'99, pp 402 – 405.
[9] U. Pal, B. Chaudhuri, Script line separation from Indian Multi-Script Documents, in Proceedings of the Fifth International Conference on Document Analysis and Recognition, ICDAR'99, 1999, pp 406 – 409.
[10] V. Bansal, R. Sinha, On how to describe shapes of Devanagari Characters and use them for Recognition, in Proceedings of ICDAR'99, 1999, pp 410 – 413.
[11] S. Anatani, L. Agnihotri, Gujarati Character Recognition, in Proceedings of ICDAR'99, 1999, pp 418 – 421.
[12] C. Sundaresan, S. Keerthi, A study of representation for Pen based Handwriting recognition of Tamil Characters, in Proceedings of ICDAR'99, 1999, pp 422 – 425.
[13] M.B. Sukhaswami, Seetharamulu, A.K Pujari, Recognition of Telugu characters using Neural Networks, Int. Journal of Neural Systems, September, 1995, 6(3), page 317 – 357.
[14] A. Negi, et al., An OCR system for Telugu, in Proceedings of International Conference on Document Analysis and Recognition, ICDAR – 2001, Seattle, USA.
[15] C. Vasantha Lakshmi, C. Patvardhan, Ranjit Singh, "A novel basic symbol approach for Telugu OCR with neural networks", Journal of the Computer Society of India, March, 2003, pp 31-39.
[16] C. Vasantha Lakshmi, C. Patvardhan, "Recognition of basic symbols in Telugu by Neural networks", STRANS-2002, March, 15 – 17, 2002, IIT Kanpur, Kanpur.
[17] C. Vasantha Lakshmi, C. Patvardhan, "An OCR system for Telugu text: A basic symbol approach", Int. Jl. on Pattern Analysis and Applications, July, 2004, pp 190 - 204.
[18] C. Vasantha Lakshmi, Unpublished Ph.D. Thesis, Dayalbagh Educational Institute, Agra, India, 2003.
[19] M. Sonka, V. Hlavac, R. Boyle, Image processing, Analysis, and Machine Vision, Second Edition, Brooks/Cole Publishing Company, 1998.
[20] G. Srikanthan, S. W. Lam, and S.N. Srihari, Gradient based contour encoding for character recognition, Pattern Recognition, 1996, 29(7), pp 1147 - 1160.
[21] LEAP, Indian language software, CDAC, Pune, India.
[22] Manjunath B.S., Salembier, P., and Sikora, T. (Eds.), Introduction to MPEG-7, John Wiley & Sons, 2002.