Printed Odia Digit Recognition Using Finite Automaton

Ramesh Kumar Mohapatra, Banshidhar Majhi, and Sanjay Kumar Jena

Department of Computer Science and Engineering,
National Institute of Technology, Rourkela, India, 769008
{mohapatrark, bmajhi, and skjena}@nitrkl.ac.in

Abstract. Odia digit recognition (ODR) is one of the intriguing areas of research topic in the field of optical character recognition. This communication is an attempt to recognize printed Odia digits by considering their structural information as features and finite automaton with output as recognizer. The sample data set is created for Odia digits, and we named it as Odia digit database (ODDB). Each image is passed through several precompiled standard modules such as binarization, noise removal, segmentation, skeletonization. The image thus obtained is normalized to a size of 32×32 2D image. Chain coding is used on the skeletonised image to retrieve information regarding number of end points, T-joints, X-joints and loops. It is observed that finite automaton is able to classify the digits with a good accuracy rate except the digits ୨, ୧, and ୭. We have used the correlation function to distinguish between ୨, ୧, and ୭. For our experiment we have considered some poor quality degraded printed documents. The simulation result records 96.08% overall recognition accuracy.

Keywords: ODR, Finite Automaton, Correlation, and Chain Coding.

1 Introduction

A correct interpretation of digits, characters, and symbols from printed documents and other materials, is very much important in the field of document image analysis. The Optical digit recognition (ODR) is a mechanism of automatic recognition of numerals from scanned and digitized paperses and other documents. Albeit it is a challenging task, it contributes gigantically to the furtherance of the automation process and enhances the man-machine interface in many practical applications [1], [2] such as:

(a) Helping blind people in reading,
(b) Automatic text entry into the computer,
(c) Preserving historical documents in digitized form, and
(d) Automatic sorting of postal mails through ZIP/PIN code, bank cheques, and other documents.

Odia is the official language of the state Odisha (Former Orissa) accounting for over 40 million people and the second official language of Jharkhand (another state in India). The Odia script arose from the Kalinga script that is the descendant of the Brahmi script of ancient India. It is the sixth classical language among many Indian languages. Now-a-days, automatic analysis and precise recognition of on-line and off-line optical characters [3] is one of the human necessities and intriguing area of research. The Odia digits and their corresponding English numerals are shown in Fig. 1.

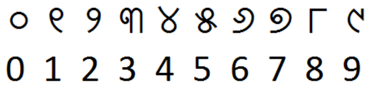


Fig. 1. Sample Odia digit data set and its corresponding English digits

Based on the feature vector matching, *Akiyama et al.* [4] have proposed a multi-font character recognition system and have done the experiment for a large set of perplexed written documents. The suggested system is adequate to read various types of preprint documents with an accuracy of 94.8%. *Chaudhuri et al.* [5] have proposed a system to recognize characters from the printed Bangla script with the constraint that the document should have Linotype font text. They claimed an overall accuracy of about 96% for the text printed on plain paper. In [6] *Chaudhuri et al.* have described a strategy for the recognition of printed Oriya script and arrogated an overall accuracy of 96.3% including both Odia text and numerals. They examined their proposed system on an extensive set of published Oriya material. To the best of our knowledge, we observed that a numerable amount of work has been carried out uptil now for the recognition of printed Odia digits.

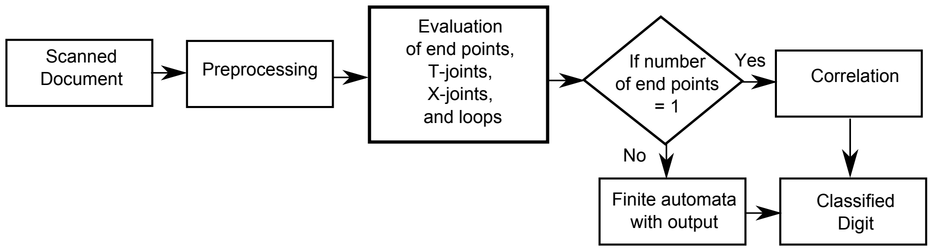

Scanned Document
Preprocessing
Evaluation of end points, T-joints, X-joints, and loops
If number of end points = 1
Yes
No
Correlation
Finite automata with output
Classified Digit

Fig. 2. Proposed model for printed Odia digit recognition

The paper is organized as follows. In Section 2 the proposed model for the recognition of printed Odia digit and the associated preprocessing methods are discussed. The observational results are presented in Section 3 followed by concluding remark in Section 4.

2 Proposed Model for printed Odia digits

In the Fig. 1, we delineated the steps for the recognition of printed Odia numerals. In our proposed methodology, a database containing the images of printed Odia digits (ODDB) has been maintained, which are collected from various printed documents. It contains printed digits of different shapes, sizes, and fonts. Any test image is first pre-processed using some conventional techniques. This module will perform a series of operations such as binarization followed by the line, word, character segmentations [7], and thinning to make the image manageable for the next phase. Each segmented character is skeletonised using some morphological operations. Use of Freeman's encoding [8], [9] over skeletonized image is explained in Section 2.2. From this sequence, we generate a string of length four, $\langle a_1 a_2 a_3 a_4 \rangle$, where a_1, a_2, a_3 and a_4 represent the number of end points, T-joints, X-joints, and loops respectively. All these information are concatenated to generate a string of length four as shown in Table 1. String thus obtained is fed to a finite automaton with output for classification. If the number of endpoints is one, then the image is passed to the correlation function for classification.

Table 1. Structural information for Odia digits

Odia Digits	EPs	TJs	XJs	Loops	String	Odia Digits	EPs	TJs	XJs	Loops	String
୦	0	0	0	1	0001	୫	2	0	1	2	2012
୧	1	1	0	1	1101	୬	2	0	0	0	2000
୨	1	1	0	1	1101	୭	1	1	0	1	1101
୩	3	1	0	0	3100	୮	2	1	0	0	2100
୪	2	0	1	1	2011	୯	2	2	0	0	2200

2.1 Pre-processing

Basic morphological operations [10] are applied to perform the tasks, such as binarization, segmentation, thinning, resizing, on the scanned document. We exploit standard methods for this purpose. We utilized the adaptive binarization technique [7] and segmentation to get the isolated digits. The segmented image is slenderized to single pixel width and then it is normalized to a size of 32×32.

2.2 Chain coding

The proposed scheme extracts one string of strokes from the thinned object of the character using Freeman encoding method (see Fig. 3). Except the digit zero all other digits must have at least one endpoint. The extraction is done by starting from the endpoint and tracing the direction from one pixel to another. Finally,

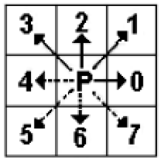


Fig. 3. Freeman chain code

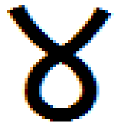


Fig. 4. segemented image of the letter Four

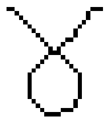


Fig. 5. Skeletonised image of the letter Four

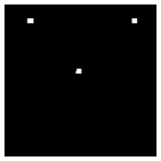


Fig. 6. End points and X-point of the letter 4

the image is represented by a series of numbers from the set {0, 1, 2, 3, 4, 5, 6, 7}. Each number represents the transition between two consecutive boundary pixels. We have considered the digit ୪ and its skeletonized image as shown in Fig. 4 and Fig. 5 respectively. To demonstrate how to calculate the parameters such as number of endpoints (EPs), T-joints(TJs), X-Joints(XJs) and loops, the encoding sequence is obtained experimentally for the digit ୪ which is given by 0 7 7 7 7 7 7 7 7 7 7 7 1 1 1 0 1 2 1 1 1 2 1 0 0 4 4 5 6 5 5 5 6 5 4 5 5 6 7 7 7 7 7 6 6 6 6 6 5 6 5 4 4 5 4 4 4 3 3 3 3 2 2 2 2 2 1 1 1 1 1 2 3 3 3 3 3 3 3 3 3 4. The sequence generated by the chain code is analyzed to have information on the number of endpoints (EPs), T-joints(TJs), X-Joints(XJs) and loops. In Fig. 6 the two end points (top-left and top-right) along with one X-joint (middle) are shown.

2.3 Recognition Phase

For the recognition of printed numerals from any scanned document, first we do the pre-processing to get a normalized thinned image of size 32×32. From the Freeman chain coding we get the string as mentioned in the Table 1. In our proposed methodology, we used the finite automaton with output, as recognizer. The state transition diagram is shown in the Fig. 7 where the initial state is Q_0 and the states with two concentric circles are the accepting states with certain output. The equivalent Algorithm 1 gives a step-by-step process of recognizing the digits by the finite automaton with output. It clearly shows that all digits are recognizable except the digits ୧, ୨, and ୭ because they represent the same string. So, to remove this ambiguity we use the correlation method. The two-

Algorithm 1: Recognition of Odia Digits

```
    Input: A test image (segmented/normalized) $img$ for recognition.
    Output: The value $Character$ is returned
1   $\langle a_1 a_2 a_3 a_4 \rangle$ = Call FIND($img$)
2   if $a_1 = 0$ then
3       $Character =$ ୦
4   else if $a_1 = 3$ then
5       $Character =$ ୩
6   else if $a_1 = 1$ then
7       $Character =$ Call $FUNC_{corr}()$
8   else if $a_1 = 2$ AND $a_2 = 1$ then
9       $Character =$ ୭
10  else if $a_1 = 2$ AND $a_2 = 2$ then
11      $Character =$ ୪
12  else if $a_1 = 2$ AND $a_2 = 0$ AND $a_3 = 0$ then
13      $Character =$ ୯
14  else if $a_1 = 2$ AND $a_2 = 0$ AND $a_3 = 1$ AND $a_4 = 1$ then
15      $Character =$ ୮
16  else if $a_1 = 2$ AND $a_2 = 0$ AND $a_3 = 1$ AND $a_4 = 2$ then
17      $Character =$ ୫
18  else $Character = \phi$
19  return $Character$
```

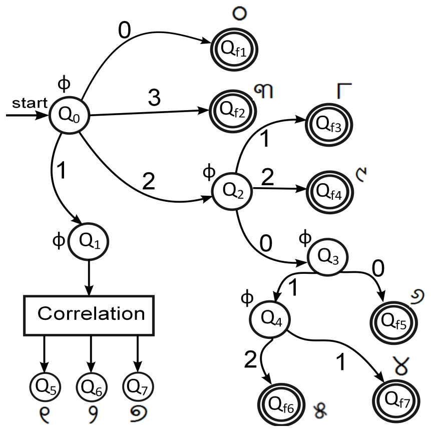


Fig. 7. State diagram for digit recognition

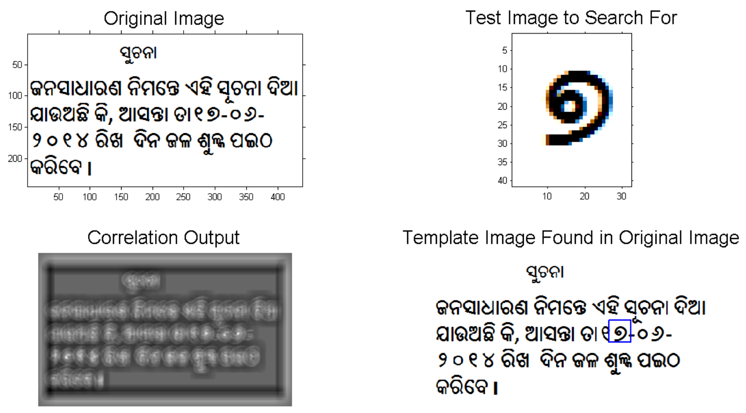
dimensional correlation coefficients are calculated as in the Equation 1,

$$r = \frac{\sum_m \sum_n \left(A_{mn} - \hat{A}\right)\left(B_{mn} - \hat{B}\right)}{\sqrt{\left(\sum_m \sum_n \left(A_{mn} - \hat{A}\right)\right)^2 - \left(\sum_m \sum_n \left(B_{mn} - \hat{B}\right)\right)^2}} \tag{1}$$

,where $\hat{A}$ = mean of A and $\hat{B}$ mean of B. The function FIND(img) will return the number of end points, T-joints, X-joints, and loops respectively. The function $FUNC_{corr}()$ used in the Algorithm 1, will return the character with maximum correlation value.

3 Experimental Results and Discussions

We have examined our proposed scheme on a large set of printed Odia documents, including regional newspapers, magazines, articles, and books. Each document was infiltrated through all the preprocessing phase to have manageable isolated characters. The function $FIND()$ will return the string $\langle a_1 a_2 a_3 a_4 \rangle$. Due



ସୂଚନା
ଜନସାଧାରଣ ନିମନ୍ତେ ଏହି ସୂଚନା ଦିଆ
ଯାଉଅଛି କି, ଆସନ୍ତା ତା ୧୭-୦୬-
୨୦୧୪ ରିଖ ଦିନ ଜଳ ଶୁଳ୍କ ପଇଠ
କରିବେ।

Fig. 8. Recognizing the digit seven

to some badly degraded printed documents the finite automaton with output is able to classify all the digits with good accuracy except the digits ୧, ୨, and ୭ because they represent the same string. To distinguish among these we use the correlation function. We consider 50 such documents where the number of occurrences of each digits along with their misclassification rate is mentioned in the Table 2. For the sake of understanding we have taken a printed document with good quality printing, where it contains ୧, ୨, and ୭ along with some other digits and Odia text. We run the procedure $FUNC_{corr}()$ for the test image where digits with identical string are present. The output is depicted in the Fig. 8 with a bounding box. The overall recognition rate is 96.08%.

4 Conclusion

Here, we have suggested a scheme for printed Odia digit recognition using finite automaton. The obtained overall recognition accuracy of our system is about

Table 2. Recognition rate of individual Odia digits

Odia digits	Number of occurrence	Number of samples unclassified	Recognition rate in %
୦	233	2	99.14
୧	246	19	92.27
୨	156	14	91.02
୩	212	3	98.58
୪	286	5	98.25
୫	223	2	99.10
୬	202	10	95.05
୭	313	23	92.65
୮	185	2	98.91
୯	219	7	95.85

96.08%, which is better than other techniques suggested in this paper. Certainly, there is a scope for improvement as far as accuracy is concerned. The proposed scheme requires to be tested on perplexed printed documents and degraded historic documents.

References

1. Ø. D. Trier, A. K Jain, and T. Taxt. Feature extraction methods for character recognition-a survey. *Pattern recognition*, 29(4):641–662, 1996.
2. U. Pal and B.B. Chaudhuri. Indian script character recognition: a survey. *Pattern Recognition*, 37(9):1887 – 1899, 2004.
3. N. Nikolaou, M. Makridis, B. Gatos, N. Stamatopoulos, and N. Papamarkos. Segmentation of historical machine-printed documents using adaptive run length smoothing and skeleton segmentation paths. *Image and Vision Computing*, 28(4):590–604, 2010.
4. T. Akiyama and N. Hagita. Automated entry system for printed documents. *Pattern Recognition*, 23(11):1141 – 1154, 1990.
5. B.B. Chaudhuri and U. Pal. A complete printed Bangla {OCR} system. *Pattern Recognition*, 31(5):531 – 549, 1998.
6. B. B. Chaudhuri, U. Pal, and M. Mitra. Automatic recognition of printed Oriya script. *Sadhana*, 27(1):23–34, 2002.
7. G. Louloudis, B. Gatos, I. Pratikakis, and C. Halatsis. Text line and word segmentation of handwritten documents. New Frontiers in Handwriting Recognition. *Pattern Recognition*, 42(12):3169 – 3183, 2009.
8. P. Zingaretti, M. Gasparroni, and L. Vecci. Fast chain coding of region boundaries. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(4):407–415, Apr 1998.
9. H. S. Cruz, E. Bribiesca, and R. M. R. Dagnino. Efficiency of chain codes to represent binary objects. *Pattern Recognition*, 40(6):1660 – 1674, 2007.
10. MATLAB. *version 7.10.0 (R2010a)*. The MathWorks Inc., Natick, Massachusetts, 2012.