A FONT AND SIZE INDEPENDENT OCR FOR MACHINE PRINTED GUJARATI NUMERALS

Shailesh A. Chaudhari, Dr. Ravi M. Gulati

ABSTRACT

Character recognition is major research area since its inspiration. So, far very limited progress has been made in it, specifically for Indian languages. Recognition of Gujarati script is a less studied area and no significant attempt is made so far to recognize Gujarati glyphs. In this paper we have presented a simple yet robust solution for recognition of offline multi-font computer generated and machine printed Gujarati Numerals. Pursued by the pre-processing techniques, we used a method called correlation based template matching where a numeral is identified by analyzing its shape and comparing its features that distinguish each numeral. The system appears to be very robust against font variations and large shape variations.

Keywords: Template, Correlation, Segmentation, Normalization, Probability, etc.

1. INTRODUCTION

Gujarati is a phonetic language and is spoken by more than 50 million people in Gujarat- a western state of India and also in surrounding states Rajasthan, Maharashtra, Madhya Pradesh etc. Though it is a very widely spoken language, no significant work is found in the literature that addresses the recognition of Gujarati language. Like other languages such as Sanskrit, Hindi, Marathi, etc. which have been derived from Devanagari, some of the Guajarati characters are very similar in appearance. The numerals in Indian languages are based on sharp curves and hardly any straight line is available. Fig.1 is a set of Gujarati numerals.

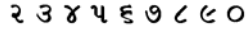


Fig. 1: Gujarati Numerals 0-9.

As can be seen in Fig. 1, Gujarati numerals are very peculiar in nature. Only two Gujarati numerals viz. one (1) and five (5) are having straight line, which makes numeral identification a little more difficult. Also Gujarati numerals are often misclassified as shown in Fig. 2. As shown in Fig. 2.(a) numerals zero (૦), three (૩) and seven (૭), Fig. 2.(b) numerals one (૧) and six (૬), and Fig. 2.(c) numerals eight (૮) and nine (૯) share similar shapes.

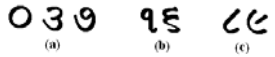


Fig. 2: Confusing Gujarati Numerals.

This paper addresses the problem of printed multi-font size independent Gujarati numerals. Gujarati numeral recognition requires cleaning, digitization, segmentation, vectorization and normalization process is performed as preprocess. Further, correlation based template matching model is used for feature extraction and numerals classification.

When we consider machine-printed documents, we can divide the OCR systems in three groups: Mono-font, Multi-font, and Omni-font. Mono-font OCR systems deal with documents written with one specific font: their accuracy is very high but they need a specific module for each font. Omni-font OCR systems allow the recognition of characters of any font, and for this reason their accuracy is typically lower. Finally, Multi-font OCR systems handle a subset of the existing fonts. Their accuracy is related to the number and the similarity of the fonts under consideration. These systems achieve the best results when a single letter has very similar features in each font and it is easy to discriminate among different classes. This paper is organized in the following sections; Section 2 describes the early attempts in Indian language OCR. Section 3 explains suggested pre-processing. Section 4 describes algorithm, which we have implemented in our work. Section 5 is devoted to feature extraction. Section 6 describes correlation based template matching for numerals recognition. Lastly in section 7, conclusion is explained.

2. RELATED WORKS

A lot of research work has been done and is still going on in OCR for various languages. More and more researchers are attracted to this challenging field. No significant work is found in the literature that addresses the recognition of Gujarati language. The earlier computer based OCR system that have been developed confined to recognize only the printed or hand written numerals of fixed size and fonts. But, the present work aims at producing a system, which could recognize numerals, especially Gujarati numerals of any arbitrary size, shape and fonts. Although recognition of handwritten numerals is well-researched topic and many techniques of recognition of both machine printed and hand printed characters are available in literatures, but not much work reported on the recognition of multi-font Gujarati numerals, in recent times.

In 1979 R. K. Sinha et al. presented work for Devnagari script [1]. They analyzed the structural characteristics of Devnagari script. In 1995 a feature based approach has been adopted by Rao et al. [2] for Telugu script recognition which works on isolated characters. In 1995 Hochberg et al. [3] presented their work to

automatic script identification from images using cluster-based templates. In 1995 Itzhak et. al. [4] presented their work on Multiple subclass pattern recognition: A maximin correlation approach for character recognition.

In 1999 Antani et al. [5] presented the first ever work of character recognition for printed or digitized Gujarati language. They used Euclidean minimum distance classifier (EMDC) and hamming distance classifier (HDC) to classify various printed Gujarati characters. In 2001 A. Negi et al [6] used a compositional approach for Telugu script recognition. They used connected components and fringe distance template matching to recognize printed characters. In 2001 Omari [7] presented his work on Hand-written Indian Numerals recognition system using template matching approaches. He used neural network and neuro-fuzzy approach in his work. In 2002 Lakshmi et al. [8] presented their work for multi-font Telugu script recognition. In 2003 Lakshmi et al. [9] presented their work to recognize printed basic symbols of Telugu language. They used seven moments for feature abstraction and then K-nearest neighbor algorithm. In 2006 Hung-Ming Sun [10] presented his work to identify multi language optical font using stroke template.

In 2007 J. Dholkia et al. [11] presented their work on Gujarati language. They used combined approach of wavelet feature extraction and neural net architecture to classify Gujarati characters. In 2007 R. S. Kunte te al. [12] presented their work to identify printed Kannada text. They used Zernike moments and neural network to classify printed Kannada text. In 2008 Park et. al. [13] presented his work for Optical character recognition system using BP algorithm. They used feed forward back propagation algorithm in their work. In 2009 Padma te al. [14] presented their work to identify Telugu, Devnagari and English scripts using discriminating features. In 2009 Freedman et al. [15] presented his work for character recognition using support vector machine. He used template matching and correlation technique to recognize the character.

3. OUR APPROACH

Any OCR implementation consists of a number of pre-processing steps followed by the actual recognition. The number and types of pre-processing algorithms employed on the scanned image depend on many factors such as age of the document, paper quality, resolution of the scanned image, the amount of skew in the image, the format and layout of the images and text, the kind of script used and also on the type of characters - printed or handwritten. The recognition stage usually involves calculating a number of statistical parameters and hence recognizing the character. Typical pre-processing stages include noise cleaning,

binarization, skeletonization, skew detection and correction and feature extraction - like component segmentation.

A. IMAGE BINARIZATION/DIGITIZATION

Binarization is a technique by which the gray scale images are converted to binary images. The most common method is to select a proper threshold for the image and then convert all the intensity values above the threshold intensity to one intensity value representing either "black" or "white" value. This is done in order to make the computer understand the form. All intensity values below a threshold are converted to one intensity level and intensities higher than this threshold are converted to the other chosen intensity.

We used Otsu's threshold algorithm to binarize our gray scale image. In our convention, black represented a character (or noise) and white represented the foreground. Otsu's threshold makes an assumption that the histogram of the gray scale image has a bimodal distribution.

B. NOISE REMOVAL

Scanned documents often contain noise that arise due to printer, scanner, print quality, age of the document, etc.. Therefore, it is necessary to filter this noise before we process the image. The commonly used approach is to low-pass filter the image and uses it for later processing.

In our case, we assumed that the document does not have high noise content, in which case the image can be binarized directly. In our approach each component in the image which has area less than 20 pixels is considered as noise. This assumption justified because this algorithm is designed to capture the text which has font size between 16 to 72 points.

C. COMPONENT LABELLING

Once noise has been removed, it is necessary to identify each component in image document. We specify unique label to each corresponding connected component in the document. There are two methods to identify connected components:

- 4-connected method, where pixels are connected if their edges touch. This means that pair of adjoining pixels is part of the same object only if they are both on and are connected along the horizontal or vertical direction.
- 8-connected component, where pixels are connected if their edges or corners touch. This means that if two adjoining pixels are on, they are part

of the same object, regardless of whether they are connected along the horizontal, vertical, or diagonal direction.

In our approach we have used 8-connected component for labelling connected components, because almost all Gujarati numerals are having very sharp curve.

D. COMPONENT AREA IDENTIFICATION

Before analyzing any character, it is important to identify the (pixel) boundaries of that character. Thus, a bounding box has to be used for identifying each numeral. For this, we first calculate the horizontal projection and then vertical projection of the labelled component and identify the start and end positions of each labelled component, which show the boundaries of each character in the image document. Now, the bounding rectangle is used to identify the exact bounding area of the numeral.

E. SEGMENTATION

The pre-processing stage yields a clean document in the sense that maximum shape information with minimal noise on normalize image is obtained. The next stage is segmenting the document into its sub components and extracting the relevant features for recognition stages. There are two types of segmentations as listed below.

Segmentation is an operation that seeks to decompose an image of sequence of characters into sub images of individual symbols. Character Segmentation strategies are divided into following categories.

- **Segmentation by dissection method:** It identifies the segments based on character like properties. This process of cutting of image into meaning full components is given a special name, dissection. Dissection is an intelligent process that analyses an image without using any specific class of shape information. Available method based on dissection of an image is connected component analysis, which we used in our approach.
- **Recognition based segmentation:** It searches the image for components that match predefined classes. Segmentation is performed by use of recognition confidence including syntactic or semantic correctness of the overall result.

F. NORMALIZATION

The normalization of the numerals is essential because of the different font style and different font size which result in several variations in shapes and sizes. Therefore to bring about uniformity in the input numerals, all the segmented components should be made of the same size. For this reason

segmented numerals are fit into a standard size window of 24 X 42. Every measure has to be taken to preserve the exact aspect ratio of the input numeral. The size of the window is selected due the fact that the height of the numeral is more than the width of the numeral.

4. ALGORITHM

Flowchart given in Fig. 3. depicts recognition algorithm.

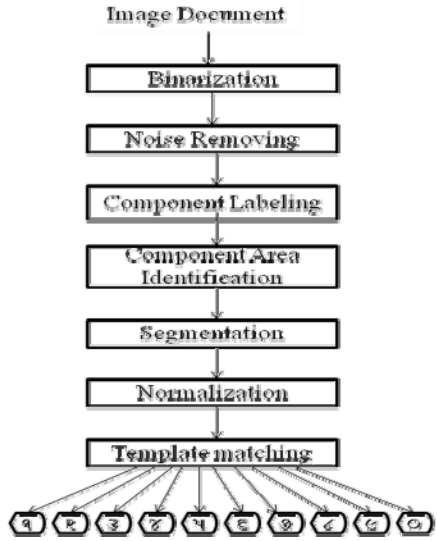


Fig. 3: Schematic block diagram of Gujarati Numerals OCR

The algorithm identifies the printed multi-font size independent numerals based on continuous approach. It identifies more than one numeral. It is having the following steps:

1. Template creation for ten individual Gujarati numerals.
2. Noise removing and labelling to existing connected components.
3. Perform segmentation for individual labelled component with bounding box.
4. Normalize each segmented component to a common size.
5. For numeral identification we follow correlation based template matching with three basic steps:

5.1. First we evaluate problem by computing the probability of output sequence of observation according to correlation based template matching. Here a vector is created for each segmented numeral image

and is compared with a set of stored template images. Then calculate matching probability of generated vector with corresponding pixels for each template image.

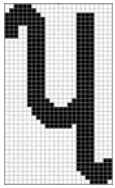
5.2. After problem evaluation, generate probability for each template against given segmented input image sequentially and then use it for numeral recognition.

5.3. Once appropriate sequence of probability have been generated there after it is necessary to find most likely probability and output maximum probabilities.

Above three steps, compare numeral's pattern being identify with the entire existing numeral set's stored as templates. The numeral which has the highest probability matching has been displayed as a recognize numeral.

6. Repeat steps 3 to 5 for all segmented components.

5. FEATURE EXTRACTION/CLASSIFICATION



The image of all the segmented characters are normalized (rescaled) into a common height and width producing a grid of 24 X 42 pixel-size (shaped-zones) as shown in Fig.4. The pixel density is calculated as binary patterns and therefore a vector is created. However, due to varying nature of font-family, there was dissimilarity between the feature vectors of the same class.

Fig. 4: Creation of machine generated Numeral's vector

This algorithm gives fast and highly accurate result for printed multi-font, size independent Gujarati numerals. We have used different font family for these experiments like NILKANTH, HARIKRISHNA, GHANSHYAM, GOPIKA, etc.

6. TEMPLATE MATCHING FOR NUMERAL RECOGNITION

Template matching, or matrix matching, is one of the most common classification methods. In template matching, individual image pixels are used as features. Classification is performed by comparing an input numeral image with a set of templates (or prototypes) for numerals [6]. For each numeral correlation coefficient is computed and saved. The correlation coefficient is defined as:

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

where $-1.0 \le r \le +1.0$.

In this equation, A is the template image and $\overline{A}$ is the average value of the template. B is the segmented input image and $\overline{B}$ is the average value of the input numeral image. Also, m and n specifies starting and ending index of the vector.

Each comparison results in a similarity measure between the input numeral and the template. One measure increases the amount of similarity when a pixel in the observed character is identical to the same pixel in the template image. If the pixels differ the measure of similarity may be decreased. After all templates have been compared with the observed numeral image, the numeral's identity is assigned as the identity of the most similar template.

7. CONCLUSION

Recognition rate is highly affected by similarity of various numerals. We have treated individual image pixels as features, where each comparison results in similarity measure between the input numeral and the template. The comparison is performed on pixel by pixel basis.

The test set used in this experiment was getting a good set of numerals for classification. The numerals used for the experiment were enclosed in a bounding region of a fixed size. Different font families represent the same numeral differently and the correlation between similar numerals is varies from font to font. This preliminary research helped us focus our attention on these matters so that issues for building a robust numeral recognition can be studied. This segmentation based approach proved to be efficient for multi-font, size independent numerals. The multi-font aspect for Gujarati characters is under investigation.

REFERENCES:

1. R.K. Sinha, H.N. Mahabala, “Machine recognition of Devnagari script”, IEEE transactions on Systems, Man, and Cybermetics (1979) 435-441.
2. P. Rao and T. Ajitha, “Telugu script recognition”, In International Conference on Document Analysis and Recognition, pages 323–326, 1995.
3. Judith Hochberg, Lila Kerns, Patrick Kelly, and Timothy Thomas automatic script identification from images using cluster-based templates Proceedings of the Third International Conference on Document Analysis and Recognition (ICDAR '95), 1995 IEEE.
4. Hadar L. Avi Itzhak, Jan A. Van Mieghem, Leonardo Rub, “ Multiple Subclass pattern recognition: A maximin correlation approach” IEEE transaction on pattern analysis and machine intelligence VOL.17 No. 4, April-1995.
5. S. Antani, L. Agnihotri, “Gujarati character recognition”, fifth International Conference on Document Analysis and Recognition (ICDAR’99), 1999, pp. 418-421.

6. Negi, C. Bhagvati, and B. Krishna, "An OCR System for Telugu" Proceedings of the Sixth International Conference on Document Analysis and Recognition (ICDAR'01) 0-7695-1263-1/01 $10.00 © 2001 IEEE
7. Faruq Al-Omari Hand-Written, "Indian numerals recognition system using template matching", Proceedings of the ASC/IEEE International conference on computer systems and applications, 2001 IEEE.
8. C.V. Lakshmi, C. Patvardhan, "A multi-font OCR system for Telugu text", Proceedings of the Language Engineering Conference (LEC'02) 2002 IEEE
9. C.V. Lakshmi, C. Patvardhan, "Optical character recognition of basic symbols in printed Telugu text", IE(I) Journal-CP 84 (2003) 66-71.
10. Hung-Ming Sun Multi-Linguistic, "Optical Font Recognition Using Stroke Templates", The 18th International Conference on Pattern Recognition (ICPR'06) 2006 IEEE
11. J. Dholkia, A. Yajnik, and A. Negi, "Wavelet Feature Based Confusion character sets for Gujarati script" International Conference on Computational Intelligence and Multimedia Applications 2007.
12. R.S. Kunte, R.D.S. Samuel, "A simple and efficient optical character" recognition system for basic symbols in printed Kannada text S¯adhan¯a Vol. 32, Part 5, October 2007, pp. 521–533. © Printed in India
13. Sang sung park, Won gyo Jung, Young geun shin, Dong-sik Jang, "Optical character recognition system using BP algorithm", IJCSNS International journal of computer science and network security, VOL. 08 No. 12, December 2008.
14. M.C. Padma and P.A. Vijaya, " Identification Of Telugu, Devanagari And English Scripts Using Discriminating Features", International Journal of Computer science & Information Technology (IJCSIT), Vol 1, No 2, November 2009
15. K. Freedman, "A Cognitive Model of Character Recognition Using Support Vector Machines", World Academy of Science, Engineering and Technology 58 2009