



## **A Correlation Coefficient based Model to Separate and Classify Noncursive (Grantha Script) Symbols**

Lalit Prakash Saxena

Department of Computer Science, University of Mumbai, Vidyanagari, Mumbai 400 098  
lasaxmail@aol.com

*Abstract:* Symbols are confined to documents either as isolated notations or hand-written texts with a number of notable features, however distinguishes, from other writing variations. This paper describes a method to separate and classify handwritten non-cursive symbols (of Grantha) from document images. This method use statistical correlation coefficient for separation and classification instead of recognizing the symbols. The model comprises of selection, separation of symbols and preprocessing steps, like normalization, skeletonization, and finally, the classification. The method employs bounding box algorithm for the location of script symbols in the document images. The efficiency of the method is, as such, it selects only the script symbols and excludes non-symbol components. In the proposed method, preprocessing steps makes the separated symbols suitable for classification. For experimental verification, 50 degraded document images of varying deteriorating complexities were tested. The resulting symbol classification rate (i.e., the proportion of symbols automatically classified) was obtained close to,  $\approx 70\%$ .

### **1. Introduction**

Symbols are involved in document images either as isolated notations, or hand-written textual content. Grantha script<sup>1</sup> has a number of notable features, which distinguish it from other conventional scripts. Each symbol represents a consonant with an inherent vowel. Other vowels are indicated using a diacritic or separate symbols. Symbols are grouped according to the way of pronunciation [1]. The separation and classification of Grantha script symbols is quite difficult due to the 10 numerals, 14 vowels, 34 consonants, 13 vowel modifiers and conjunct symbols. These complexities separate it from other languages like Latin alphabets or Tamil characters. However, the applications or methods developed for particular languages or scripts are confined to that language or script, it cannot be applied, employed, or modified for other languages and scripts [2]. Printed Grantha symbols are easy to segment using horizontal and vertical projection profiles; however, smaller fonts and those containing composite symbols may introduce touching problem [5].

Selection of a symbol separation method is the single most important factor in achieving high classification performance. When compounded with more generic problems such as noise and merged or broken symbols, hand-written script writing offers a challenging area for symbol separation and classification. This paper describes current results of a system that separates script symbols from degraded text on a photographed (using a 14 megapixel camera with resolution of 1280×1280) document images. This paper is dedicated to the script symbols separation and classification. It describes a method to separate and classify handwritten non-cursive symbols from images using the correlation coefficient method [3].

The bounding box algorithm locates script symbols in enhanced image using initial labeling. The method selects the present symbols using bounding box around the edges of the symbol [7]. Only script symbols are labeled with the bounding box algorithm and non-symbol components gets excluded. The symbol image represents the single Grantha script symbol, on the condition if it is complete and unbroken. The normalization module makes the separated symbols appropriate for further classification procedures. A morphological operation for

---

Received: April 11<sup>st</sup>, 2014. Accepted: September 9<sup>th</sup>, 2015

DOI: 10.15676/ijeel.2015.7.3.14

classification of script symbols is a specific process that tries to solve two main problems: a) Connectivity preserving: If a symbol is connected in the normalized image, it must be connected in the morphologically operated image also, and b. Shape preserving: Though it is not essential to preserve exact symbol shape or size, uniqueness of the symbol shape should be preserved.

Standard heuristics have been defined to guide this automated process [8,9]. In this paper, the different modules making up the symbol separation algorithm are described, along with its representative examples of results. Experiments done on degraded document images show that this method delivers quite satisfactory performance, making symbol separation feasible for real-world applications. When this algorithm was implemented over 50 degraded documents of varying complexity, the symbol classification rate (i.e., the proportion of symbols that can be automatically classified) was close to,  $\approx 70\%$ . The classified symbols classes are used to map script symbols to user-defined scripts or scripts known to expert user. The symbol transliteration is then followed to acquire embedded knowledge out of the degraded document images. Hence, it facilitates the process of document preservation.

The paper is organized as follows. The review of previous literature is provided in section 2. Section 3 describes the symbol classification model. Section 4 explains symbol selection and separation. Processing of symbols is given in section 5 and section 6 details symbols classification. The paper concludes in section 7.

## 2. Related work

The tradition of preserving old literary work is a usual practice all around the globe. This work is an attempt to preserve old literary contents available on documents written in Grantha script. This paper has no influence of the actual content of such documents but it can be assumed to preserve the textual works. This work used enhanced images as basic input to the symbol separation and classification model [27]. The distinctly classified symbols can be preserved at this stage also and used for further processing desired in script enhancement. Furthermore, script symbols are classified into distinct classes based on the similarity measures. This is an approach beyond the scope of mere digitizing or enhancing only the images of the old documents.

It is evident from the literature that rarely any work shown symbols processing of such documents. Only image enhancement is present in most of the work [24–26]. And those who have tried symbol processing used clean script symbols [11,14]. This work is first of its kind where symbol processing of heavily degraded script is done. This is an exact preservation procedure of such a valuable old documents. In context of preservation this is the only précised preservation procedure in the field of script image processing. There are several techniques for symbol selection and separation but most of them are script specific and may not work with different scripts [10–12]. Even in printed handwritten documents symbol selection is required due to the touching of the symbols. In contrast to other methods [4–6], proposed method emphasizes on separation rather than recognition.

It is because of the limitation that hand-written symbols are of different shapes and sizes that varies from writer to writer. Probability of accurate classification depends upon the involved classification scheme. In a classification scheme, [18] addressed complexities of the classification problems and measured accuracy criterion. A detailed literature given by [19] describes the probability of misclassification that is also considered while working on Grantha script images. In case of machine printed symbols, the classification is quite simple since the size of the symbol images after separation tends to vary a little less as compared to hand written symbols.

## 3. Symbol classification model

The symbol classification model classify script symbols from an input enhanced document image.

*Block 1*, input image: an enhanced document is made input to this block, irrespective of the image format. The model takes binary image as input, however binarized, using the binarization method described in [27].

*Block 2*, locate and separate symbols: each symbol (referred as connected component) of the enhanced image is located and treated as a distinct symbol-image for purposes of classification, unless it gets treated as noise.

*Block 3*, normalize: All the symbol-images (i.e., character-images) are normalized to the same size (35×25 pixels), so that different instances of the same symbol are not treated as being different.

*Block 4*, morphological operations: These operations restore pixels that got removed during enhancement and removes excess pixels, while maintaining pixel connectivity.

*Block 5*, skeletonize: Skeletonization reduces the thickness of symbols while maintaining their shape and size.

*Block 6*, classify: Symbol images are placed in the same class, i.e., classified as being one and the same, if their correlation coefficient value  $r \geq 0.70$ .

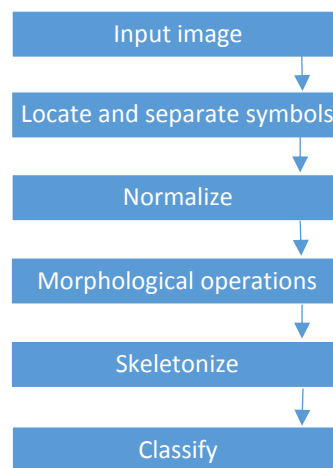


Figure 1. The block diagram of symbol classification model.

This summarizes the working of symbol classification model for separated symbols with each block specifying its input, processing and output. Factors on which the classification of script symbols for a document image depends: 1) Effective enhancement of the image of the document, 2) Precise morphological operations, 3) Symbols and non-symbols classification, and 4) Correlation coefficient value.

#### 4. Symbol selection and separation

In case of handwritten documents, with unusual writing styles, the complexity increases in finding the symbols. Different shapes and writing style of symbols are difficult to select and separate from an image. Bigger shape of the symbol would interlace with upper and/or lower textual lines and may increase confusion in proper selection. Creative style sometimes intervenes with the nearby written symbols and might produce complex horizontal or vertical projection profiles for a single symbol. This paper uses Bounding Box Algorithm [13] to select the symbols distinctly in document images.

##### A. Bounding box algorithm

The algorithm first considers all the potential symbols (connected components) within the image and finds the coordinates of the point of location for bounding box's top and bottom vertices. This algorithm selects four pair of coordinates from combination of points;  $min_x$ ,

$min_y$ ,  $max_x$  and  $max_y$  on connected components in the image. The only possible combination of pair of coordinates are  $(min_x; min_y)$ ,  $(max_x; min_y)$ ,  $(min_x; max_y)$  and  $(max_x; max_y)$ .

---

**Algorithm 1 Bounding Box Algorithm**

---

- 1: Procedure INPUT:(Image of Grantha script): Image as an input to the algorithm
  - 2: Search the connected components in the image with the information about their location coordinates from left to right and top to bottom.
  - 3: For every distinct connected component, compute the following:
  - 4: Find the coordinates  $(min_x; min_y)$  and  $(max_x; min_y)$  of the top corners of the bounding box of the symbol.
  - 5: Find the coordinates  $(min_x; max_y)$  and  $(max_x; max_y)$  of the bottom corners of the bounding box of the symbol.
  - 6: Calculate the absolute values  $l_1$ ,  $l_2$ ,  $l_3$  and  $l_4$  of the distance of the lines connecting  $(min_x; min_y)$  to  $(max_x; min_y)$ ,  $(max_x; min_y)$  to  $(max_x; max_y)$ ,  $(min_x; min_y)$  to  $(min_x; max_y)$  and  $(min_x; max_y)$  to  $(max_x; max_y)$ .
  - 7:  $l_1$  &  $l_3 = \text{mod} \sqrt{(max_x - min_x)^2}$
  - 8:  $l_2$  &  $l_4 = \text{mod} \sqrt{(max_y - min_y)^2}$
  - 9: Connect the lines with the absolute distance at the located point of the symbol with no loss of connected component pixel. (Color of box lines can be chosen as per the visible requirement; default is red)
  - 10: Repeat Step 2-3 until all the symbols get bounding box around it.
  - 11: Output: Bounding Boxes along with symbols for separation from the image. :Image as an output with bounding boxes around each and every symbol present in that image
  - 12: end procedure
- 

Points are located on the image to create the bounding box. See figure 2.

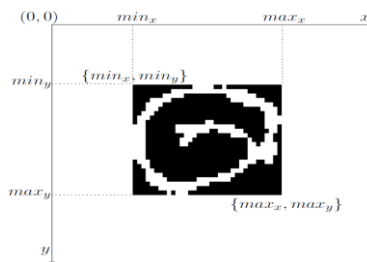
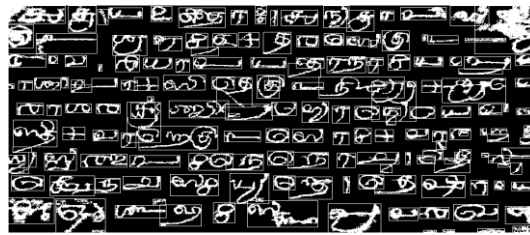


Figure 2. The bounding box around a symbol represented by its coordinates.



(a).



(b).

Figure 3 (a). Symbols located in Figure 2 (b). Roman Script

In this work the size of a written Grantha script symbol varies from as low as  $40 \times 10$  pixels to sometimes  $100 \times 100$  pixels. There are instances that the size of symbol with composition of

multiple symbols having the size as large as 350×190 pixels. Removing all the regions less than 40×10 pixels provide an image with all possible symbols. The degradations leads to incomplete and absence of symbols in the document image. Symbol separation algorithm separates each and every possible symbol from the image with an estimated symbol size of 35×25 pixels. This size is chosen to retrieve maximal number of readable symbols. The result of this algorithm is given in figure 3a for Grantha script and Figure 3b for Roman script.

### B. Symbol separation algorithm

The script symbol size is considered on the visual observation in the symbols selected image using the following four criteria: 1) Composition of symbols: estimate the size of all composite symbols, large composition is better than small. 2) Broken symbols: measure the symbols with fissures. 3) Blurring of symbols: assess the blurring of the symbols. 4) Noise in homogeneous areas: calculate the size of noisy spots and false components in background and foreground [15] use symbol separation step and [4] improved the method to remove degradations for incorporating into suitable applications.

---

#### Algorithm 2 Symbol Separation Algorithm

---

1: Procedure INPUT: (symbols selected using bounding box image)

1: Select character size,  $C_s$ , calculate  $C_s$  for an image, in the case of Grantha script document images,  $C_s = 35 \times 25$  for 1280×1280 resolution image.

2: Estimate average pixel area,  $P_a = (\text{total connected components area}/\text{number of components})$  for all symbols and non-symbols, remove connected components having an average  $P_a$  below the  $C_s$ .

3: Manually select some symbols if needed and estimate the  $C_s$  accordingly.

4: end procedure

---

## 5. Preprocessing of symbols

Before proceeding to actual classification, symbol images are preprocessed. It is required to preprocess the separated symbol images to transform to a standard dimension. Instances of non-symbol separated images having the same average size of the symbol or more are also evident. To distinguish non-symbols from script symbols it becomes necessary to preprocess all the separated symbol images. For reducing the complexity, the background of the symbol image is kept black having pixel value 0 and 1 for white.

### A. Normalization

The normalized symbol size is adjusted such that it is an average of the all symbol sizes. In this work 35×25 pixels is considered as the average symbol size. The process of normalization is explained as follows:

The processed binarized symbol image  $f'(x, y)$ , see Figure 4a, is normalized to the estimated size of the symbol as 35×25, see Figure 4b, combining both black and white pixels. The normalized symbol image  $f(x, y)$  is produced using [4] method explained below:

$$f(x, y) = f' \left( \frac{\text{width} \times x}{35} + \delta x, \frac{\text{height} \times y}{25} + \delta y \right)$$

here *width* and *height* are the measures of the size of the symbol before normalization,  $\delta x$  and  $\delta y$  are the measures of the horizontal and vertical spaces between left-top corners of the white connected component and the image plane, respectively.

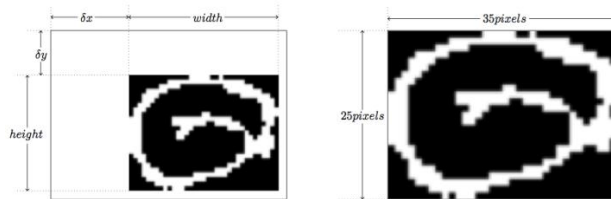


Figure 4 a) original separated symbol image, b) normalized image

The above stated normalization procedure is appropriate for symbol shapes, see Figure 5a, for width  $> 3 \times \text{height}$  is calculated, so as to get the image plane ratio of normalized width of 35 pixels. And for symbol shapes, see Figure 5b, for height  $> 3 \times \text{width}$  is calculated to get the normalized height.



Figure 5 a) original symbol images, b) normalized images

### B. Skeletonization

Table 1. Summarizes the results of separation and preprocessing steps like normalization and skeletonization.

Image	Separated symbols image†	Original dimension*	Normalized <sup>#</sup>	Skeletonized
Schar 1		111 × 58		
Schar 2		89 × 45		
Schar3		158 × 59		
Schar4		47 × 41		
Schar5		73 × 44		
Schar6		82 × 38		
Schar7		72 × 57		
Schar9		85 × 55		
†the separated symbols are selected through the edges by bounding boxes *in pixels, after separation from enhanced script image <sup>#</sup> the standardized dimension of 35 × 25 pixels				

Skeletonization or medial axis transformation is an iterative erosion process until the symbol gets the thickness of one pixel while maintaining its original connectivity [17]. The effectiveness of this operation is in the symbol skeleton it produces, that maintains the originality of symbol image by a consecutive set of connecting pixels in terms of shape, topology and connectivity. The connected pixels follow 8-connectivity in order to maintain the symbol connectedness. Skeletonization reduces symbol image to its minimum pixels description and maintains the symbol representation. Table 1 summarizes the results of separation and preprocessing steps like normalization and skeletonization.

### C. Unsuccessfully separated components

Unsuccessful separation components were examined thoroughly and it was observed that the major reason for inappropriate or non-separation was the poor quality of the manuscripts images. Low resolution and low contrast, non-uniform illumination, and particularly the interference of various noisy background with the text, contributed substantially to the unsuccessful separation of the components. A few samples of the images of the unsuccessful separation components obtained from the experiments are provided in Figure 6. Figure 6a suffers from strong background interference even though the text is quite clear. Thus, in this case non-separation is evident and the text was not getting separated. Further, the images in Figure 6b and Figure 6c suffer from both a complex background with non-uniform illumination and low contrast. The processing of Figure 6b shows both inappropriate and unsuccessful separation. However, for the image in Figure 6c no separation was possible as all the characters appears to be strongly connected due to dark background, noise and low contrast.

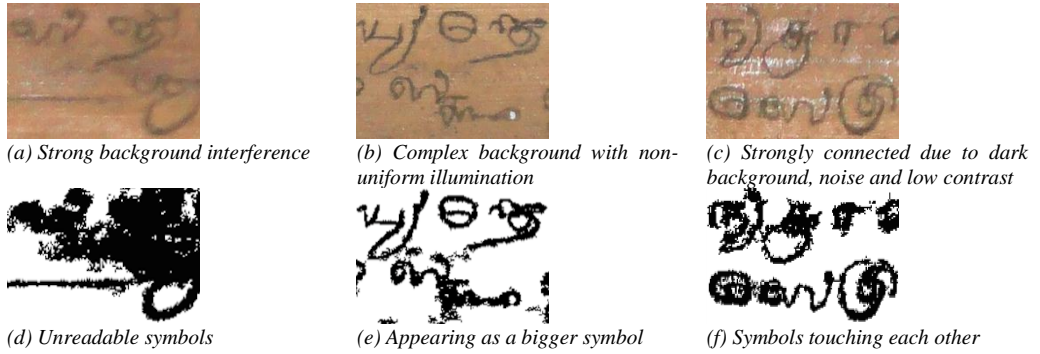


Figure 6. Instances of unsuccessfully classified symbols

## 6. Symbol Classification

In case of Grantha script symbols, there are 71 symbols involved overall in textual construction [1]. Use of all 71 symbols in any document or in this context on a single document image is very rare. Therefore, the maximum number of classes is restricted to 71 only. This is for individual symbol found in the image but for composite symbols it may increase up to 30 more classes [1]. According to the observations, in a Grantha script image, there are 8 lines having around 50 symbols; subject to enhancement and removed degradations. On an average, it needs  $50 \times 8 = 400$ , 50 symbols in 8 lines in image, searches, to obtain the average symbol size. It is observed that an average symbol image size has  $35 \times 5$  pixels. So, the number of searches is for  $35 \times 25 \times 400$  pixels. For an image with  $3000 \times 500$  resolution; the number of searches needed is  $\approx 4.287$  per pixel.

### A. Correlation Coefficients

The correlation determines the strength of a linear relationship or similarity between two skeletonized symbol images [20,21]. The value of correlation coefficient, given by  $r$ , lies between -1 and +1 or  $|r| = \pm 1$ , representing minimum or negative correlation to maximum or positive correlation similarity, respectively [16,22]. Let  $\mathcal{P}$ ,  $\mathcal{Q}$  are the skeleton symbol images, and  $m$ ,  $n$  are the rows and columns of an image respectively. The correlation coefficient between two images is calculated as

$$r = \frac{\sum_m \sum_n (\mathcal{P}_{mn} - \bar{\mathcal{P}})(\mathcal{Q}_{mn} - \bar{\mathcal{Q}})}{\sqrt{[\sum_m \sum_n (\mathcal{P}_{mn} - \bar{\mathcal{P}})^2][\sum_m \sum_n (\mathcal{Q}_{mn} - \bar{\mathcal{Q}})^2]}}$$

where  $\bar{\mathcal{P}}$  and  $\bar{\mathcal{Q}}$  are the values of mean pixel intensities of  $\mathcal{P}$  and  $\mathcal{Q}$ , where

$$\bar{\mathcal{P}} = \frac{1}{N} \sum_{i=1}^m \sum_{j=1}^n \mathcal{P}_{i,j} \text{ and } \bar{\mathcal{Q}} = \frac{1}{N} \sum_{i=1}^m \sum_{j=1}^n \mathcal{Q}_{i,j}$$

and ' $\mathcal{N}$ ' is the number of pixels in the image. Worst case analysis of the correlation method states that the correlation will fail, if any of the input images has no variance, i.e. all 0's or all 1's, evaluating the summation in the denominator equal to 0. Though, in this work, neither of the sample nor the test symbol images is variance-free, therefore, there is no probability of occurrence of this case.

Symbol images are classified using correlation coefficient,  $r = 0.70$ . This value is selected using a heuristic approach. Arbitrarily an image is chosen and correlated all symbols images and found that the maximum value obtained for  $r$  is  $\approx 0.70$ . In results it is observed that 24 classes out of 266 symbol images are obtained for a document image. In example document image in Figure 3a, 266 Grantha separated symbols from a document image were placed in 24 distinct classes (see figure 7).



Figure 7. 24 classes of classified symbols

#### B. Class Decision and Rejection

The class formation is to group symbols having similar features into separate classes. Symbol separation is a generalization task and class formation is a specialization task [22,23]. In order to ensure correct classification, it is required to have the exact combination of symbol separator and similarity measuring method. A good symbol separator should offer two properties in symbols separation: a) the script symbols assigned to the same class should have maximum similarity, and b) the script symbols assigned to different classes will be less similar or have significantly high dissimilarity.

### 7. Conclusion

In this paper, preprocessing techniques are used for separation and classification of document symbol image. Many preprocessing techniques have been developed, but still there is more to achieve for degraded symbol images. Generally the preprocessing techniques are specific to the application and every technique is not applicable to all applications. In particular, the application dependency of preprocessing techniques affects the processing of symbol images.

In normalization a symbol image is normalized to a predetermined symbol size ( $35 \times 25$ ) for easy processing. Morphological operations are used to increase and reduce pixels in the symbol images, respectively. Finally, skeletonization is done to obtain thinned symbol images resembling the normalized symbol without losing the pixels connectivity and reduction in symbol size. These skeletonized symbol images are classified into appropriate classes based on similarity measures, i.e. using correlation coefficient. The correlation coefficient value for classification of Grantha script symbols was obtained to be around 0.70.

The future work aims at new applications such as symbol recognition can be used for processing of such documents. Also, an OCR for character recognition of these documents can be built to interpret textual information. This would lead to an actual enhanced digitized preservation of old documents.

### 8. References

- [1]. K. Venugopalan, A Primer in Grantha Characters, 2 ed., vol. I, St. Peter, Minn: James H. Nye, 1983.



- [2]. Y.T. Tu, K.C. Fan, and L.S. Wang, "Classification of Machine Printed and Handwritten Texts using Character Block Layout Variance," *Pattern Recognition*, Vol. 31, pp. 1275–1284, 1998.
- [3]. T. Wakahara and Y. Yamashita, "K-nn Classification of Handwritten Characters via Accelerated GAT Correlation," *Proceedings of International Conference on Frontiers in Handwriting Recognition (ICFHR) 2012, IEEE Computer Society*, September 2012, pp. 143–148.
- [4]. Y. Mizukami, "A Handwritten Chinese Character Recognition System using Hierarchical Displacement Extraction based on Directional Features," *Pattern Recognition Letters* Vol. 19, pp. 595–604, 1998.
- [5]. U. Pal and B.B. Chaudhuri, "Indian Script Character Recognition: A Survey," *Pattern Recognition* Vol. 37, pp. 1887–1899, 2004.
- [6]. H. Fujisawa, "Forty years of Research in Character and Document Recognition—An Industrial Perspective," *Pattern Recognition*, Vol. 41, pp. 2435–2446, 2008.
- [7]. S. Mori, C.Y. Suen, and K. Yamamoto, "Historical Review of OCR Research and Development," *Proceedings of the IEEE*, Vol. 80, pp. 1029–1058, 1992.
- [8]. N. Nixon and A. Aguado, *Feature Extraction and Image Processing*, 2 ed., ISBN 978-0-12-372538-7, Elsevier Ltd., London, UK, 2008.
- [9]. C.L. Liu, *Arabic and Chinese Handwriting Recognition*, Lecture Notes in Computer Science, vol. 4768, ch. "Handwritten Chinese Character Recognition: Effects of Shape Normalization and Feature Extraction," pp. 104–128, Springer-Verlag Berlin, SACH 2006 Summit College Park, MD, USA, 2006.
- [10]. R. Plamondon and S.N. Srihari, "On-line and Off-line Handwriting Recognition: A Comprehensive Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22, pp. 63–84, 2000.
- [11]. K. Jung, K.I. Kim, and A.K. Jain, "Text Information Extraction in Images and Video: A Survey," *Pattern Recognition*, Vol. 37, pp. 977–997, 2004.
- [12]. R.G. Casey and E. Lecolinet, "A Survey of Methods and Strategies in Character Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 18, pp. 690–706, 1996.
- [13]. M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis, and Machine Vision*, Thomson-Engineering, 2007.
- [14]. O.D. Trier, A.K. Jain, and T. Taxt, "Feature Extraction Methods for Character Recognition—A Survey," *Pattern Recognition*, Vol. 29, pp. 641–662, 1996.
- [15]. G. Vamvakas, B. Gatos, S. Petridis, and N. Stamatopoulos, "An efficient Feature Extraction and Dimensionality Reduction Scheme for Isolated Greek Handwritten Character Recognition," *Ninth International Conference on Document Analysis and Recognition, ICDAR*, September 2007, pp. 1073–1077.
- [16]. R.C. Gonzalez and E.R. Woods, *Digital Image Processing*, 2 ed., Prentice Hall, Upper Saddle River, New Jersey, 2002.
- [17]. L. Lam, S.W. Lee, and C.Y. Suen, "Thinning Methodologies—A Comprehensive Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 14, pp. 869–885, 1992.
- [18]. T.K. Ho and M. Basu, "The Complexity of Classification Problems," *Proceedings of 15th International Conference on Pattern Recognition*, September 2000, pp. 43–47.
- [19]. G.T. Toussaint, "Bibliography on Estimation of Misclassification," *Transactions on Information Theory*, Vol. 20, pp. 472–479, July 1974.
- [20]. N. Mezghani, A. Mitiche, and M. Cheriet, "Bayes Classification of Online Arabic Characters by Gibbs Modeling of Class Conditional Densities," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 30, pp. 1121–1131, 2008.
- [21]. S. Mori, C.Y. Suen, and K. Yamamoto, "Historical Review of OCR Research and Development," *Proceedings of the IEEE*, Vol. 80, pp. 1029–1058, 1992.

- [22]. R.O. Duda, P.E. Hart, and D.G. Stork, *Pattern Classification*, 2 ed., John Wiley and Sons, New York, 2001.
- [23]. J. Rocha and T. Pavlidis, "Character Recognition without Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 17, pp. 903–909, 1995.
- [24]. M. Sezgin and B. Sankur, "Survey over Image Thresholding Techniques and Quantitative Performance Evaluation," *Journal of Electronic Imaging* Vol. 13, pp. 146–165, 2004.
- [25]. P.K. Sahoo, S. Soltani, and A.K.C. Wong, "A Survey of Thresholding Technique," *Computer Vision, Graphics, and Image Processing* Vol. 41, pp. 233–260, 1988.
- [26]. O.D. Trier and T. Taxt, "Evaluation of Binarization Methods for Document Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 17, pp. 312–315, 1995.
- [27]. L.P. Saxena, "An effective binarization method for readability improvement of stain-affected (degraded) palm leaf and other types of manuscripts," *Current Science*, Vol. 107, pp. 489–496, 2014.



**Lalit Prakash Saxena** was born in Obra, Sonbhadra in 1984. He graduated in Physics and Mathematics from Government Post-Graduate College, Obra in 2006. He has done his Masters in Computer Applications from Bundelkhand University in 2009. He is currently doing PhD in Computer Science from Department of Computer Science, University of Mumbai, Mumbai, India. His main research interest lies in Image Processing, Pattern Recognition, Document Image Analysis and Manuscript Image Enhancement and Script Classification.