# Compression of Scan Digitized Handwritten Text for Indian Language Document

*Smita V. Khangar*
Department of Computer Science and Engg.
G.H. Raisoni College of Engineering
Nagpur, India
smita146@gmail.com

*Dr. L.G. Malik*
*Professor*: Department of Computer Science and Engg.
G.H. Raisoni College of Engineering
Nagpur, India
lgmalik@rediffmail.com

***Abstract*—Document image compression is used for the speedy transmission of the data over the web. This paper deals with effective compression scheme for handwritten gray level documents in Devnagri script. The current OCR technology is not effective for handling the handwritten textual images. The proposed compression scheme is based on the separation of foreground and background of the image. Experiments have been done for the handwritten textual images. These document images are written in Devnagri (Hindi and Marathi). The results of the some modules progress towards achieving the good compression ratio are presented. Compression scheme are available for printed textual images in Indian language. But for handwritten text images very little work is reported. Thus the compression for handwritten text in the context of Indian language is important.**

***Keywords-Document Image Compression,Foreground and Background Separation, Indian Language, Handwritten text,Devnagri Script,Gray Level Document***

## I. INTRODUCTION

As there is growing demand of digital libraries document image compression is becoming more popular. With the advancement of technology publication over the web is widely spreading. Most of the digital libraries scan the document and publish over the web. Today the large amount of data in the form of manuscript, periodicals is placed online. Many of the documents are in the form of printed and handwritten text. This data is stored in the form of scanned images. Such images are known as textual images [1]. For speedy communication this data must be in compressed form. Current digital libraries extract the text by using Optical Character Recognition. For printed text this method works well. The document containing handwritten text results in various complexities such as irregularity in lines, strokes and shapes. In Indian context document having historical significance contains mainly handwritten text. Most of the work is done for compressing the printed textual images. The images are in Chinese, Arabic language. For Indian language there is little work reported towards the compression standard.

The method for compression of printed text in Indian Language (IL) is available in literature. Hence main focus is on handwritten text instead of printed text. In this paper compression of gray level handwritten document images in Devnagri (Hindi and Marathi) script is proposed. The rest of the paper is organized as follows: Section 2 gives the overview of various approaches for textual image compression. Section 3 gives brief introduction of Devnagri script. Section 4 describes the proposed methodology for compression of handwritten text. The method is based on foreground and background separation. Section 5 describes expected results followed by conclusion and future work.

## II. PREVIOUS WORK

For compression different approaches are used followed by the various compression techniques. Some major approaches are discussed here. Previous work of compressing textual images is divided into three categories.(1) based on pattern matching and substitution (PMS) (2) based on soft pattern matching (SPM) (3) based on foreground and background separation [2][3].

### *A. Pattern Matching and Substitution*

The method PMS and SPM is supported by JBIG2 compression standard [4]. In PMS image is segmented into group of characters. Each group contains letters and symbols. Only one symbol bitmap represent certain character. For coding an image coding of bitmap representative of each group is required. It is then followed by the position of each character. Coding of new symbol is done by looking into symbol dictionary having smallest mismatched. This method achieves high compression for text images having repeated symbols. But document cannot contain only alphabet specific information. To achieve high compression ratio pattern matches must be aggressive. But this will lead to the substitution errors. This is lossy compression [4]. Classification of patterns differs according to the languages. For English languages it is more suitable than the IL. Most of the IL scripts like Devnagri (Hindi or Bangla) show the different shapes of characters.

### *B. Soft Pattern Matching*

In soft pattern matching image is divided into the marks. Theses marks are refereed with figures, letters or symbols. If matching mark is found coding of matched mark is done directly. Unlike PMS for totally mismatched mark it does not produce any error. If perfect match found then good compression ratio is achieved. Thus this method mainly applies for the printed or typed text document images [4]. For most of

the bi-level images JPEG and JPEG2000 compression standards are widely acceptable. In Color documents e.g. magazine text is written on background image. For such images to remove non-textual material preprocessing has to be done. Now this non-textual material can be coded separately. Paper [5] used symbolic compression for IL Devnagri printed text using soft pattern matching. It then uses the character segmentation and pattern classification for printed Devnagri script.

### C. *Foreground - Background Separation of Image*

In most of the images such as pictures and color documents foreground and background are important. In comparison with these kinds of images handwritten document show distinct features. Foreground shows the textual matter in terms of characters, lines, regularity and sharpness for reading purpose. Most of time background is uniform depending upon paper texture [3]. The historical handwritten documents may suffer from aging effect and noise while scanning the documents. Whenever the text has written with pin pointed pen or broad tip pen stroke marks are different. Thus unlike printed document handwritten document may not show very well contrast in foreground and background. For such document effective separation of foreground and background is a challenging task [6].

During the literature survey most of the separation techniques often focus on thresholding. Thresholding can be global or local. The selection of threshold may vary with the gray level images. Approach used by Kittler and Illingworth chooses threshold based on every pixel or region [7]. B.Gatos proposed the method of adaptive binarization for gray scale images in historical documents [8]. XU Danhua separated the background and foreground for handwritten text and scanned receipt with the median filter of large size window. Foreground is computed by subtracting background from preprocessed image [3]. Technique proposed by U.Garain is based on the connected component analysis to capture the similar color pixels. It then followed by the dominant background components detection. It is in terms of member pixels. The entire image is divided into the blocks. These blocks are treated as foreground and background parts [6]. DjVu [9] compressor effectively makes the separation of foreground and background in the context of compression. It separates text and line from the background. This approach gives results for the bi-level documents with uniform background. It is based on the color clustering algorithm. The algorithm divides the image into regular grid. Each grid delimits some small blocks.

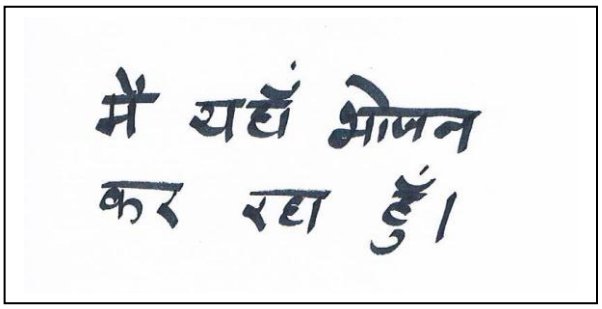


Figure 1. Handwritten textual image in Devnagri

Applying algorithms on these blocks produces the pair of colors for each block. The blocks are then form background and foreground colors. But this method fails for low contrast images such as handwritten text.

## III. FEATURES OF DEVNAGRI SCRIPT

As the proposed methodology focus on compression of handwritten Devnagri text some features of Devnagri script is discussed here. Devnagri script does not have any uppercase or lowercase distinction. Script has 5 basic vowels and 29 consonants. It has 12 modifiers. The writing style is horizontal. The characters are connected by a horizontal headline called as “shirorekha” in Devnagri. The neighboring characters are touched through this headline. It then forms the connected components. Most of the Indian script is divided into three zones namely upper zone, middle zone and bottom zone [5]. Fig. 1 shows an example of handwritten Devnagri text. The upper zone shows matra information above the headline. Middle zone is showing characters. Bottom zone may show other consonants in case of complex characters.

## IV. PROPOSED METHODOLOGY

### A. *Proposed Work Flow*

Depending on the assumption for handwritten document images having uniform background proposed workflow is shown in fig.2.

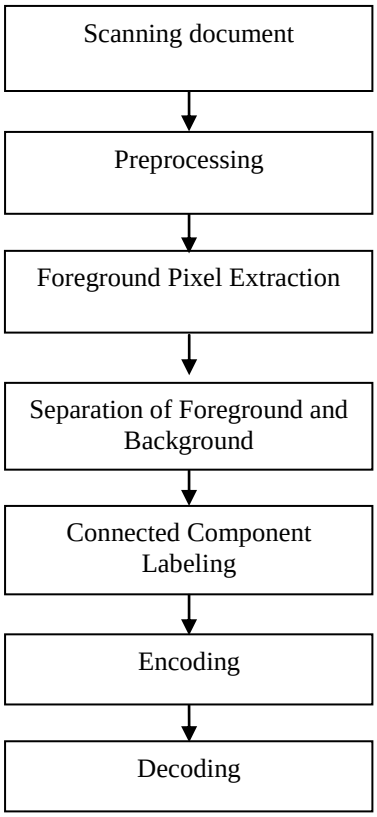


Figure 2.Flow of proposed methodology

The steps consist of scanning the document. The preprocessing step is optional here. In case of degraded image it is often required. It then consists of following steps: foreground pixel extraction, foreground –background separation, connected component labeling, encoding and decoding of the image is illustrated below.

### *B. Foreground Pixel Extraction*

Input image is scanned and image width and image height is calculated. The values of each image pixels are read in terms of RGBs and stored in the buffer. By observing the range of pixel values average value is calculated. This average value is referred as threshold. Form the image pixel values foreground and background pixel values are calculated. They are further used in next step. For showing the foreground and background pixel values image having small dimension is used here. The image shown in figure 3 having dimension 52×52 and scanned at 100dpi. The values of foreground and background pixels are shown in figure 4. Although in a small output window it is not possible to show entire values. But these pixel values are important for storing of image.

### *C. Separation of Foreground and Background Image*

There is a difference in between binarization and foreground and background separation for color and gray level document images. Binarization is a conversion of gray image into black and white image [6]. In this step calculated foreground and background pixel are separated from the buffer. The foreground and background vales are in vector form. These values are written to the array element of raster data buffer. The raster defines the value of pixel in a particular area [10]. This output raster then stored as an image on the disk.

This output image has only foreground text elements separating the background. The stored new form of image shows that the image size is reduced by 50-60%. Some experimental results are shown in table I. The images shown in figure 5 are the input images. Images (a) and (c) are input images. The handwritten text written on input images are with different pens. Image (a) uses the small tip pen while image (c) uses the bold tip pen. Such images having difference in size when they scan with different dpi. The image (a) is scanned at 300dpi and image (c) is scanned at 150 dpi. Both the images are written on plain white A4 size paper. Image (b) and (d) from figure 6 are the corresponding foreground background separated images.

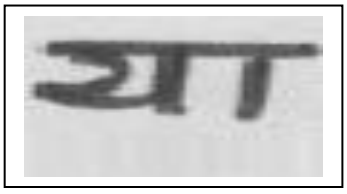

Figure 3. Sample image shown for calculation of foreground and background value

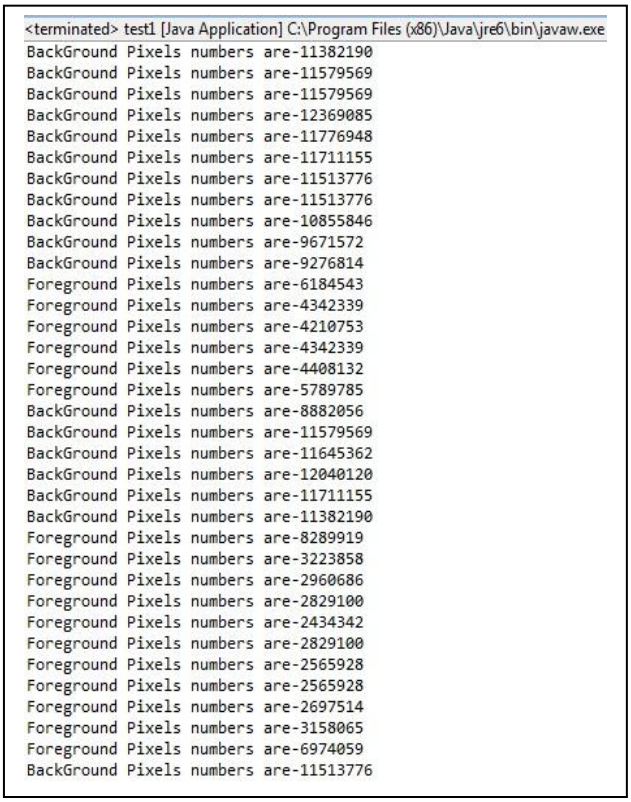


Figure 4. Output window showing values of foreground and background pixels

Up to this step some amount of compression is achieved. For further reduction in size next steps namely connected component labeling and encoding is to be done. Table I shows the reduction in original image size after foreground extraction.

### *D. Connected Component Labeling*

The connected component labeling (CCL) algorithm is applied on the background separated images. CCL detect all connected component or touching component from the images. CCL detects two types of components. Component can be label from complete word images or among the two word images [5]. CCL applies on the foreground extracted text. The relationship between the components of words or among the two words is stored.

TABLE I. SIZE OF IMAGE STREAMS

| ***Image Name*** | ***Scanned at*** | ***Original Size*** | ***Image size after JPEG Compression*** | ***Image size after separating foreground and background*** |
|---|---|---|---|---|
| Image (a) | 300 dpi | 387 KB | 99 KB | 97 KB |
| Image (c) | 150 dpi | 170 KB | 103 KB | 101 KB |

Group-3 and Group-4 [11] standard are emerging standards for handwritten documents.

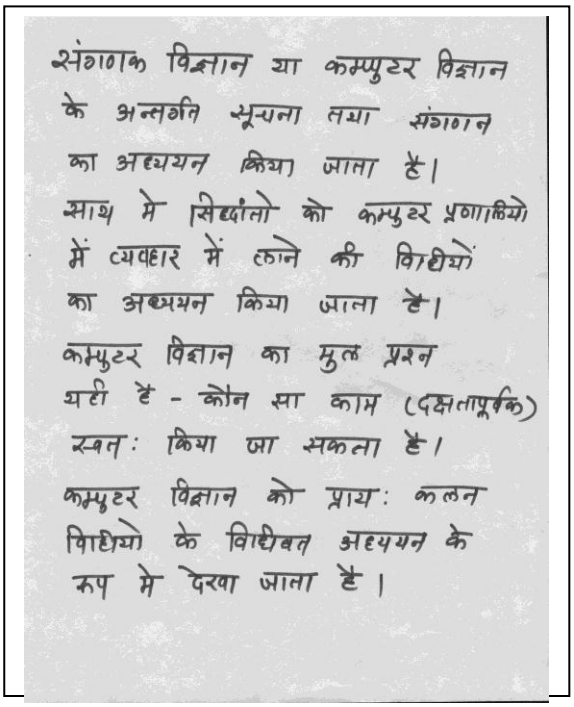


Image (a)

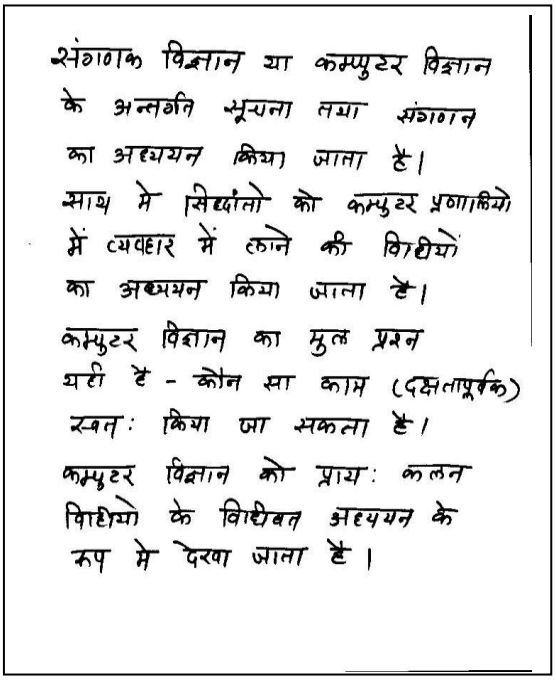


Image (b)

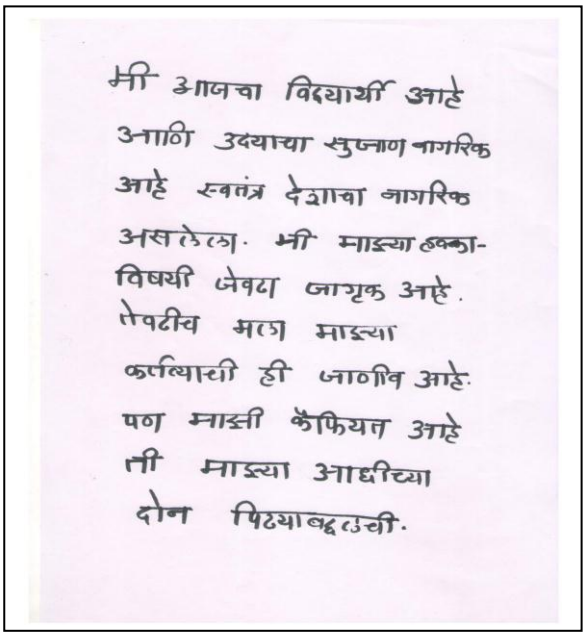


Image (c)

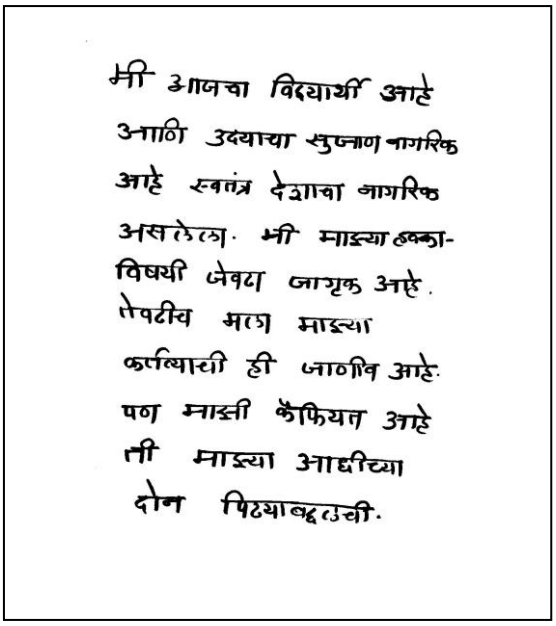


Image (d)

Figure 5.Image (a) and (c) input images written in Hindi and Marathi respectively.

Figure 6. Image (b) and (d) Background separated images

*E. Encoding and Decoding*

Compression algorithm based pattern matching such as PMS and SPM cannot give results for the handwritten documents because of variations in handwritten text. CCITT

The system uses the run length encoding (RLE) for further compression. It replaces the series of similar pixels with a count of 1 and 0 for bi-level image. The same image (a) and

(c) are encoded with the JPEG encoder and results are shown in table I. From the comparison of the results it is clear that final compressed image will achieve good compression ratio.

## V. Expected Results

The system is under implementation with some modules are tested with Devnagri script. The module work very well and show the results accordingly. The complete system with comparison of compression algorithm will be presented in next paper. On completion of all modules expected reduction in size up to 70-75% is expected. The aim is to achieve the minimum storage space by compressing handwritten text.

## VI. Conclusion and Future Work

As mentioned earlier there is a little work done on compression of handwritten text for Indian language. The proposed methodology is only focused on gray level handwritten images. The results obtained by some modules shows the effectiveness of the scheme. Future work mainly concerns with CCL on extracted text, results of compression scheme and reconstruction of original image.

## References

[1] I.Witeten, T.Bell, H.Emberson, A.Moffat, "Textual Image Compression : Two Stage Lossy/ Lossless Encoding of Textual Images". Proc. of IEEE. vol.82, No.6,pp 878-888, June 1994.

[2] Y.Ye and P.Cosman, "Dictionary Design for Text Image Compression with JBIG2", IEEE Transcation on Image Processing, vol.10(6),pp.818-828,2001.

[3] X.Danhua, B.Xudong, "High Efficient Compression Strategy for Scanned Receipts and Handwritten Documents", IEEE International Conference on Information and Engineering, pp.1270-1273,2009.

[4] P.G.Howard, "Text Image Compression Using Soft Pattern Matching" , The Computer Journal, vol. 40,pp.146-156,1997.

[5] U.Garain, S.Debnath, A.Mandal, B.Chaudhari, "Compression of Scan Digitized Printed Text : A Soft Pattern Matching Technique", ACM Symposium on Document Engineering, pp.185-193, Nov .2003 .

[6] U.Garain, T.Paquet, L.Heutte, "On Foregrond-Background Separation in Low Quality Document Images", International Journal of Document Analysis, vol. 8(1),pp.47-63,2006.

[7] J.Kittler,J.Illingworth, "Threshold Selection based on Simple Image Stastics",Computer Vision Graphics and Image Processing, vol.30,pp.125-147,1985.

[8] B.Gatos, I.Pratikakis, "An Adaptive Technique for Low Quality Historical Documents", 6th International Workshop on Document Analysis Systems, vol.3163,pp.102-113,2004.

[9] L.Bottou, P.Howard, "High Quality Document Image Compression with DjVu", International Journal of Electronic Imaging, vol.7(3), pp.410-425,July 1998.

[10] Java Sun Documentation. [Online]. Available : https://docs.oracle.com/javase/1.3/docs/api.

[11] "Lossy/Lossless Coding of Bi-Level Images", ITU-T Recommendation, 2000.