Intelligent Information Management, 2010, 2, 120-133
doi:10.4236/iim.2010.22015 Published Online February 2010 (http://www.scirp.org/journal/iim)

Text Extraction in Complex Color Document Images for Enhanced Readability

P. Nagabhushan, S. Nirmala
Department of Studies in Computer Science, University of Mysore, Mysore, India
Email: *pnagabhushan@compsci.uni-mysore.ac.in, nir_shiv_2002@yahoo.co.in*

Abstract

Often we encounter documents with text printed on complex color background. Readability of textual contents in such documents is very poor due to complexity of the background and mix up of color(s) of foreground text with colors of background. Automatic segmentation of foreground text in such document images is very much essential for smooth reading of the document contents either by human or by machine. In this paper we propose a novel approach to extract the foreground text in color document images having complex background. The proposed approach is a hybrid approach which combines connected component and texture feature analysis of potential text regions. The proposed approach utilizes Canny edge detector to detect all possible text edge pixels. Connected component analysis is performed on these edge pixels to identify candidate text regions. Because of background complexity it is also possible that a non-text region may be identified as a text region. This problem is overcome by analyzing the texture features of potential text region corresponding to each connected component. An unsupervised local thresholding is devised to perform foreground segmentation in detected text regions. Finally the text regions which are noisy are identified and reprocessed to further enhance the quality of retrieved foreground. The proposed approach can handle document images with varying background of multiple colors and texture; and foreground text in any color, font, size and orientation. Experimental results show that the proposed algorithm detects on an average 97.12% of text regions in the source document. Readability of the extracted foreground text is illustrated through Optical character recognition (OCR) in case the text is in English. The proposed approach is compared with some existing methods of foreground separation in document images. Experimental results show that our approach performs better.

Keywords: Color Document Image, Complex Background, Connected Component Analysis, Segmentation of Text, Texture Analysis, Unsupervised Thresholding, OCR

1. Introduction

Most of the information available today is either on paper or in the form of still photographs, videos and electronic medium. Rapid development of multimedia technology in real life has resulted in the enhancement of the background decoration as an attempt to make the documents more colorful and attractive. Presence of uniform or non-uniform background patterns, presence of multiple colors in the background, mix up of foreground text color with background color in documents make the documents more attractive but *deteriorates the readability*. Some of the examples are advertisements, news paper articles, decorative postal envelopes, magazine pages, decorative letter pads, grade sheets and story books of children. Further, the background patterns opted in the preparation of power point slides appear to be attractive but cause difficulty in reading the contents during presentation on the screen. These compel to devise methods to reduce the adverse effect of background on the foreground without losing information in the foreground.

There are many applications in document engineering in which automatic detection and extraction of foreground text from complex background is useful. These applications include building of name card database by extracting name card information from fanciful name cards, automatic mail sorting by extracting the mail address information from decorative postal envelopes [1]. If the text is printed on a clean background then certainly OCR can detect the text regions and convert the text into ASCII form [2]. Several commercially available OCR products perform this; however they result in low recognition accuracy when the text is printed against shaded and/or complex background.

The problem of segmentation of text information from complex background in document images is difficult and still remains a challenging problem. Development of a generic strategy or an algorithm for isolation of foreground text in such document images is difficult because of high level of variability and complexity of the background. In the past, many efforts were reported on the foreground segmentation in document images [3–16]. Thresholding is the simplest method among all the methods reported on extraction of foreground objects from the background in images. Sezgin and Sankur [14] carried out an exhaustive survey of image thresholding methods. They categorized the thresholding methods according to the information they are exploiting, such as histogram shape based methods, clustering based methods, entropy based methods, object-attributes based methods, spatial methods and local methods. The choice of a proper algorithm is mainly based on the type of images to be analyzed. Global thresholding [7,11] techniques extract objects from images having uniform background. Such methods are simple and fast but they cannot be adapted in case the background is non uniform and complex. Local thresholding methods are window based and compute different threshold values to different regions in the image [8,14] using local image statistics. The local adaptive thresholding approaches are also window based and compute threshold for each pixel using local neighborhood information [9,13]. Trier and Jain [17] evaluated 11 popular local thresholding methods on scanned documents and reported that Niblack's method [9] performs best for OCR. The evaluation of local methods in [17] is in the context of digit recognition. Sauvola and Pietikainen [13] proposed an improved version of Niblack method especially for stained and badly illuminated document images. The approaches proposed in [9,13] are based on the hypothesis that the gray values of text are close to 0 (black) and background pixels are close to 255 (white). Leedham *et al*. [8] evaluated the performance of five popular local thresholding methods on four types of "difficult" document images where considerable background noise or variation in contrast and illumination exists. They reported that no single algorithm works well for all types of image. Another drawback of local thresholding approaches is that the processing cost is high. Still there is a scope to reduce the processing cost and improve the results of segmentation of foreground text from background by capturing and thresholding the regions containing text information. Often we encounter the documents with font of any color, size and orientation. Figure 1 shows some sample color document images where the foreground text varies in color, size and orientation. Conventional binarization methods assume that the polarities of the foreground and background intensity are known apriori; but practically it is not possible to know foreground and background color intensity in advance. This drawback of conventional thresholding methods call for specialized binarization.

Text-regions in a document image can be detected either by connected component analysis [3,18] or by texture analysis method [1,19]. The connected component based methods detect the text based on the analysis of

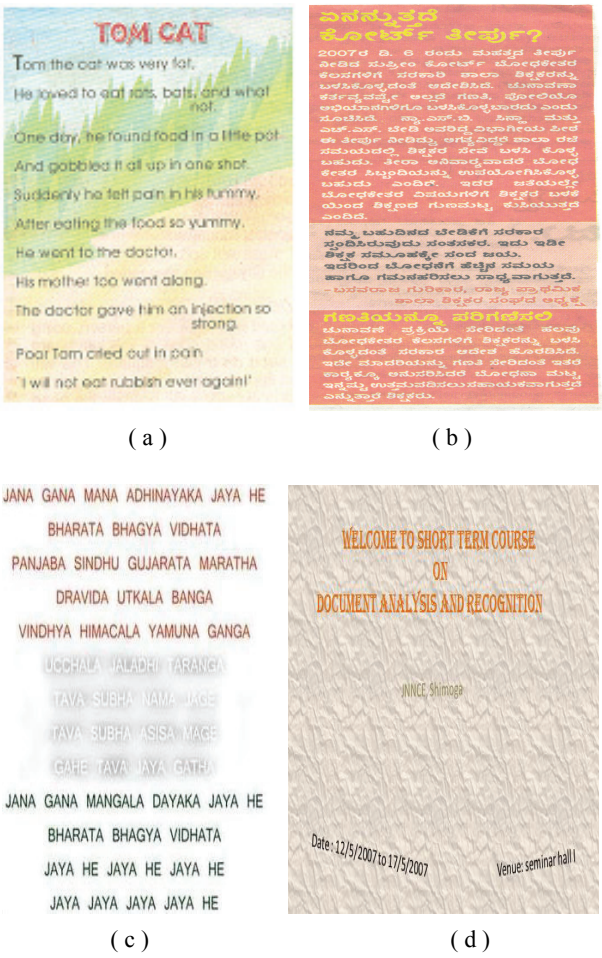


Figure 1. Color documents with printed text of different size, color and orientation.

the geometrical arrangement of the edges [16] that compose the characters. They are simple to implement and detect text at faster rate but are not very robust for text localization and also result in false text regions for images having complex background. Pietik¨ainen and Okun [12] used edge detectors to extract the text from document images. Their method fails in extracting the tilted text lines and erroneously classifies the textured background as text. Chen *et al*. [3] proposed a method to detect the vertical and horizontal edges in an image. They used different dilation operators for these two kinds of edges. Real text regions are then identified using support vector machine. The method lacks in detecting text tilted in any orientation. Zhong *et al*. [18] used edge information to detect the text lines. Their method deals with complex color images pretty well but restricted to certain size constraints on characters. The texture based methods detect the text regions based on the fact that text and background have different textures [1]. In [19] it is assumed that text is aligned horizontally or vertically and text font size is in limited range. The method proposed in [19] uses texture features to extract text but fails in case

of small font size characters. Their method is based on the assumption that the text direction is horizontal or vertical. In [2] texture based method is proposed to detect text regions in gray scale documents having textured background. In their method text strokes are extracted from the detected text regions using some heuristics on text strings such as height, spacing, and alignment [2]. The extracted text strokes are enclosed in rectangular boxes and then binarized to separate the text from the background. Their method fails to extract the text in low contrast document images. Also they fail to extract the tilted text in document images. Most of the above methods are very restrictive in alignment and type of the text they can process. Sobotta *et al*. [15] proposed a method that uses color information to extract the text in colored books and journal covers. Their method fails to extract the isolated characters. In [4] a method is proposed to separate foreground from background in low quality ancient document images. The test documents used in their method are scanner based handwritten, printed manuscripts of popular writers. Their method fails to segment the foreground text in documents with textured background. Liu *et al*. [20] proposed a hybrid approach to detect and verify the text regions and then binarize the text regions using expectation maximization algorithm. The computation complexity of verification process of the text region is high. The performance of the algorithm proposed in [20] degrades when the documents have high complex background and fails to extract the text in low contrast document images. Kasar *et al*. [6] proposed a specialized binarization to separate the characters from the background. They addressed the degradations induced in camera based document images such as uneven lighting and blur. The approach fails to extract the text in document images having textured background. It also fails to detect the characters in low resolution document images. From the literature survey, it is evident that identifying, separating the foreground text in document images and making it smoothly readable is still a research issue in case the background of a document is highly complex and the text in foreground takes any color, font, size and tilt.

In this paper we propose a novel hybrid approach to extract the foreground text from complex background. The proposed approach is a five stage method. In the first stage the candidate text regions are identified based on edge detection followed by connected component analysis. Because of background complexity the non-text region may also be detected as text region. In the second stage the false text regions are reduced by extracting the texture feature and analyzing the feature value of candidate text regions. In the third stage we separate the text from the background in the image segments narrowed down to contain text using a specialized binarization technique which is unsupervised. In the fourth stage the text segments that would still contain noise are identified. In final stage the noise affected regions are reprocessed to further improve the readability of the retrieved foreground text. The rest of the paper is organized as follows. Section 2 introduces our approach. In Section 3 experimental results and discussion are provided. Time complexity analysis is provided in Section 4. Conclusions drawn from this study are summarized in Section 5.

2. Proposed Approach

In this work we have addressed the problem of improving the readability of foreground text in text dominant color document images having complex background by separating the foreground from the background. The proposed work is based on the assumption that the foreground text is printed text. Two special characteristics of the printed text are used to detect the candidate text regions. They are, 1) Printed characters exhibit regularity in separation and 2) Due to high intensity gradient, a character always forms edges against its background. The sequence of the stages in proposed hybrid approach is shown in Figure 2. The proposed five stage approach is described in the subsections to follow.

2.1. Detection of Text Regions

The proposed method uses Canny edge detector to detect edges [21] because Canny edge operator has two advantages: it has low probability of missing an edge and at the same time it has some resistance to the presence of noise. We conducted experiments on both gray scale and RGB color model of source document images. It is observed from the experimental evaluations that the edge detection in gray scale document images resulted in loss of text edge pixels to certain extent. Hence edge detection in RGB color model of source document is proposed instead of transforming the color document to

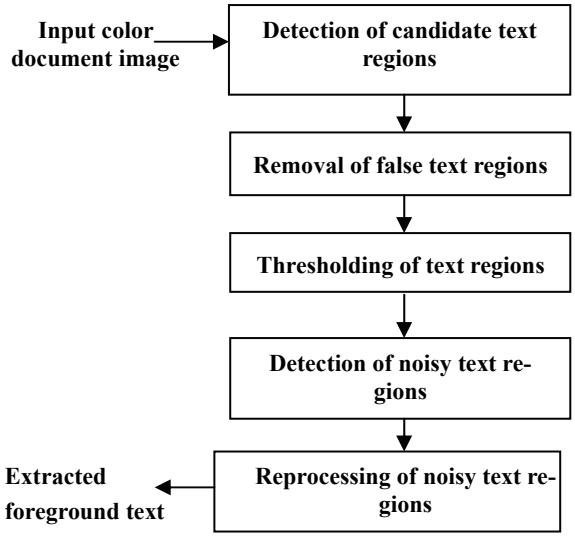


Figure 2. Stages of the proposed approach.

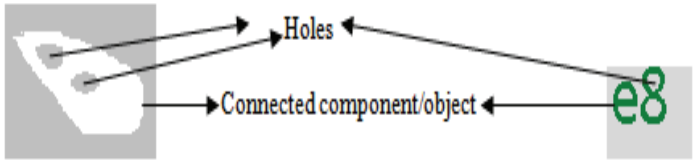
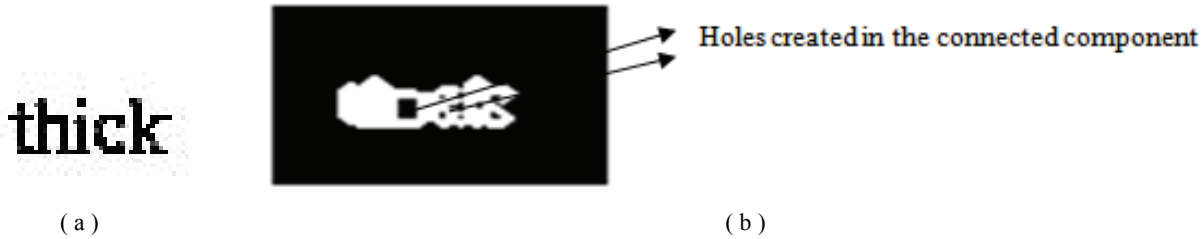


Figure 3. Holes in a connected component.



(a) (b)

Figure 4. (a) Word that composes characters without holes, (b) holes created by connecting the characters in the word.

a gray scale document. Edge detection is carried out in each color channel separately as the foreground text may be of any color and therefore the edges could be visible in one or more of these three color channels. The results of the edge detection of all the three color channels are assimilated so that no character edge gets missed. Suppose E_R, E_G and E_B are the images after applying the Canny edge operator on red, green and blue components of the input color image, the resulting edge image "E" after assimilation is given by,

$$E = E_R \vee E_G \vee E_B \quad (1)$$

where "V" represents logical "OR" operator.

The resulting edge image "E" contains edges corresponding to character objects in the input image. When the background is highly complex and decorative, the edge image "E" might contain edges corresponding to non-text objects also. An 8-connected component labeling follows the edge detection step. The non-text components in the background such as underlines, border lines, and single lines without touching foreground characters do not contain any hole. A hole in a connected component is illustrated in Figure 3.

Generally in a document image some printed characters contain one or more holes and some other characters do not contain a hole. If a word is composed of characters without holes, using dilation operation [3] it could be possible to thicken the characters so that they get connected and additional holes are created in the space between the characters. This process is depicted in Figure4.

As the text lines in most of the documents are conventionally aligned horizontally, we conducted experiments on dilation of the edge image "E" in horizontal direction. It is observed from experimental evaluations that dilation of edge image "E", only in horizontal direction is not enough to create holes in most of the connected components corresponding to character strings. Therefore we extended the dilation operation on the edge image in both horizontal and vertical directions. From the experimental results it is observed that dilation of the edge image in both horizontal and vertical directions has created holes in most of the connected components that corresponds to character strings. The size of the structuring element for dilation operation was fixed based on experimental evaluation. As no standard corpus of document images is available for this work we conducted experiments on the document images collected and synthesized by us which depict varying background of multiple colors and foreground text in any color, font, size. We dilated the edge image row-wise and column-wise with line structuring element of different sizes. Table 1(a) and Table 1(b) show the percentage loss of characters in a document image after dilating the edge image "E" with various sizes of horizontal structuring element and vertical structuring element.

Table 1(a). Percentage loss of characters for various sizes of horizontal structuring element.

Size of vertical structuring element is 3X1, total number of characters processed=6171					
Size of the horizontal structuring element	1x2	1x3	1x4	1x5	1x6
Loss of characters in percentage	1.93	1.93	2.37	2.38	4.58

Table 1(b). Percentage loss of characters for various sizes of vertical structuring element.

Size of horizontal structuring element is 1x3, total number of characters processed=6171					
Size of the vertical structuring element	2x1	3x1	4x1	5x1	6x1
Loss of characters in percentage	1.99	1.96	1.98	1.98	4.18

From Table 1(a) and Table 1(b) it is observed that with horizontal structuring element of size 1x3 and vertical structuring element of size 3x1, the percentage loss of characters in a document image is very low. This indicates that the dilation of edge image with line structuring element 1x3 in horizontal direction and line structuring element 3x1 in vertical direction creates additional holes in most of the text components which is depicted in Figure 4. Figure 5 shows the document image after assimilating the results of horizontal and vertical dilation of edge image of the input image which is shown in Figure 1(a).

The 8-connected component labeling is performed on the dilated edge image. Based on the size of the characters in the source document and spacing between the words the so labeled connected components may be composed of a single character or an entire word or part of the word or a line. The labeled component may also contain words from different lines if the words in different lines are connected by some background object. In this work the built-in function "Bwboundaries" in MATLAB image processing tool box is used to find the holes in a connected component. The connected components are analyzed to identify the object/component containing hole. We removed the connected components without hole(s). Other non-text components are eliminated by computing and analyzing the standard deviation of each connected component which is elaborated in the next subsection.

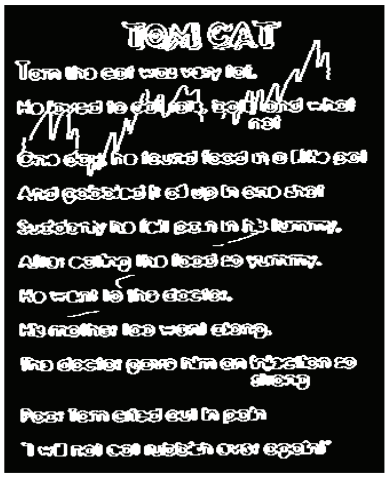


Figure 5. Document image after dilation.

2.2. Removal of False Text Regions

Because of background complexity certain amount of non-text region in the source document might be identified as text region in connected component analysis process. The proposed approach is based on the idea that the connected components that compose textual information will always contain holes. Holes in the connected components comprise the pixels from the background. Hence each connected component represents an image segment containing only background pixels in case there is no text information (false text region) or both foreground and background pixels in case the connected component contains text information (true text region). To remove the image segments containing only background pixels, standard deviation of gray scale values of all pixels in each image segment/connected component is calculated. The standard deviation in the image segments occupied with only background pixels (*ie*, image segments without text) is very low where as the standard deviation in the image segments occupied by both background and foreground pixels (*ie*, image segments containing text) is high [10]. Based on this characteristic property of document image it could be possible to discriminate the non-text image segments from image segments containing text. To set the value for "SD" we conducted experiments on document images having uniform/non-uniform background of multiple colors and foreground text of any font, color, size and orientation. We set the value for standard deviation from a set of 120 images (first 120 images in the corpus).The document image samples are selected randomly in multiples of 5, from the corpus of images synthesized and collected by us, to set the empirical value for standard deviation' SD'. The sample images selected are all distinct images from the corpus of images. From the plot shown in Figure 6, it is observed that a threshold value of 0.4 on "SD" is sufficient enough to filter out the non-text regions without loss of detected text. In addition repeating the experiment 10 times on 50 distinct samples selected randomly each time (from first 120 samples in the corpus), demonstrated that the value for standard deviation falls in the range 0.405 to 0.42. We extended the experiment on 100 more images in the corpus apart from sample images used for setting the value for "SD" and observed that SD=0.4 resulted in reduction of the false text regions without loss of text information in the document. However, although choosing a higher "SD" value reduces the false text regions it results in the loss of foreground text and choosing "SD" value lower than 0.4 leads to additional processing of more number of false text regions. Hence standard deviation of 0.4 is chosen as the threshold value.

2.3. Extraction of Foreground Text

In the proposed approach color information is not used to extract the foreground text. As already during the first

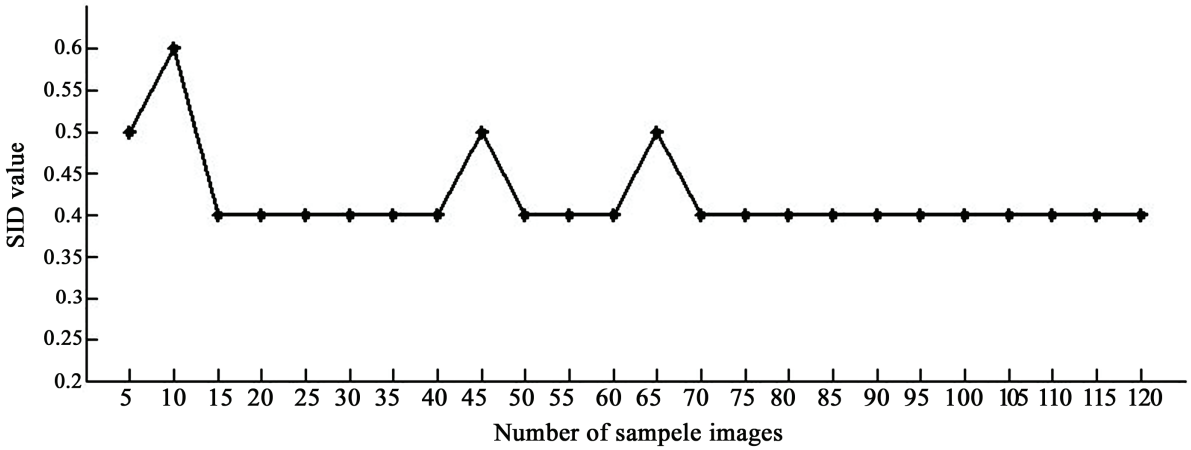


Figure 6. Plot showing the number of training sample images versus the SD value for no loss of textual information.

stage of our approach the evidences of textual edges have been drawn from intensity values of each color channel (RGB model). Also it is computationally inexpensive to threshold the gray scale of the image segment corresponding to the connected component by tightly encapsulating the segment. Figure 7 illustrates background and foreground pixels in a connected component. In each connected component average gray scale intensity value of foreground pixels and average gray scale intensity value of the background pixels are computed.

Suppose "m" and "s" are mean and standard deviation of gray scale intensities in an image segment corresponding to a connected component with hole(s), the threshold value for that segment is derived automatically from the image data as given in [9],

$$\text{threshold}=m-k*s \quad (2)$$

where (k) is a control parameter and value of (k) is decided based on the average gray scale intensity value of foreground pixels and average gray scale intensity value of background pixels. Suppose "V_f" is average gray scale intensity value of foreground pixels and "V_b" is average gray scale intensity value of background pixels. We conducted experiments on document images with varying background and foreground text of different colors. From experimental evaluations it is observed that choosing k=0.05 for $V_f > V_b$ and k=0.4 for $V_f \leq V_b$ results in a better threshold value. In this work to discriminate foreground pixels from background pixels two contrast gray scale intensity values (gray value near to 0 for foreground pixels and gray value 255 for background pixels) are assigned to pixels in the output image. Irrespective of the foreground text color and background color we produced black characters on uniform white background by suitably thresholding each image segment containing text and producing the corresponding output image segment O_{bw} using the logic as given by,

$$if\ V_f > V_b \quad O_{bw} = \begin{cases} 10 & if \quad I(x,y) > threshold \\ 255 & if \quad I(x,y) \leq threshold \end{cases}$$

$$if\ V_f \leq V_b \quad O_{bw} = \begin{cases} 10\ if & I(x,y) < threshold \\ 255\ if & I(x,y) \geq threshold \end{cases}$$

Irrespective of the foreground text color and background color the extracted characters are produced in black color on uniform white background for the purpose of improving the readability of the document contents. The resulting image might contain noise in the form of false foreground. This needs reprocessing of the resulting image to further improve the readability of document contents by OCR.

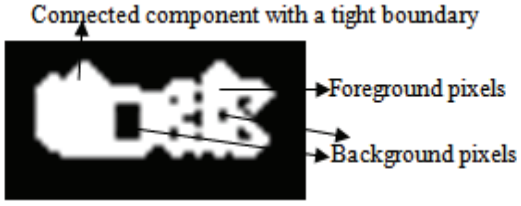


Figure 7. Illustration of foreground and background pixels in a connected component.

2.4. Detection of Noisy Text Regions

Detection of text areas/segments that need further processing is performed using a simple method. The main idea is based on the fact that the text areas that still contain noise include more black pixels on an average in comparison to other text areas/segments. The image is divided into segments of variable sizes; each segment corresponds to one connected component. In each image segment that contains text the density of black pixels, *f(S)* is computed. Suppose b(S) is frequency of black pixels in an image segment "S" and area(S) is area of image-segment "S", the density of black pixels in "S" is given by,

$$f(S) = b(S)/\text{area}(S) \quad (3)$$

The segments that satisfy the criterion $f(S) > c*d$, are

selected for reprocessing, where "d" is the average density of black pixels of all the image segments containing text. The parameter "c" determines the sensitivity of detecting noisy text regions. High value of "c" results in less text segments to be reprocessed. Low value of "c" results in more text segments to be reprocessed which would include the text segments in which noise is already removed. Figure 8 shows the noisy areas to be reprocessed for different values of "c". Optimal value for parameter "c" is selected based on higher character (or word) recognition rate after reprocessing noisy text regions in the output document image. We conducted experiments on the document images which we collected and synthesized. Table 2 shows the character and word recognition rates in percentage for various values of "c". It is observed from Table 2 that character (or word) recognition rate is high for value of c<=0.5. Also it is seen from Figure 8 that number of components to be reprocessed will be less as the value of "c" increases. So 0.5 is chosen as optimal value for parameter c.

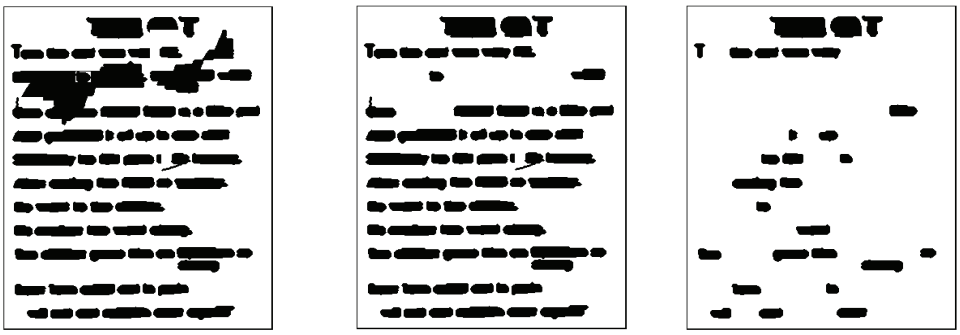
2.5. Reprocessing of Noisy Text Regions

The selected text segments containing noise in the form of false foreground pixels are reprocessed. Repeating the stage-1 on these text segments leads into text segments of smaller size. These segments are thresholded in the next stage. Since only few text segments are reprocessed instead of all the detected and verified text segments, the computation complexity of the stage-4 reduces substantially. *In fact, the entire approach can be proposed to be iterative if it is required*; but we observed that repeating stage-1 and stage-3 once on noisy regions is more than sufficient which in turn reduces the time complexity of extracting the foreground text from complex background in document images. Figure 9 shows the results at each stage in the proposed approach.

3. Results and Discussions

3.1. Experimental Results

Since no standard corpus of document images is available for this work we created a collection of images by scanning the pages from magazines, story books of children, newspapers, decorative postal envelopes and invitation cards. In addition, one more dataset of synthesized images which are of low resolution is created by us. The details of documents in the corpus of images used for testing our proposed algorithm are depicted in Table 3. The number of document images in our datasets is 220 and they are of different resolutions (96x96 DPI, 100x100 DPI, 150x150 DPI and 200x200 DPI). The output image is obtained by depositing the black characters on the white background, irrespective of the background and foreground color in the original document. The performance of text region detection is evaluated in terms of Recall (correct detects/(correct detects + missed detects)) and Precision (correct detects/(correct detects+ false alarms)). Recall is inversely proportional to missed detects whereas Precision is inversely proportional to false alarms. Missed detects indicates number of text regions incorrectly classified as non text and false alarms indicates number of non-text regions incorrectly classified as text regions. Table 4 shows the average value of precision and recall in percentage for document images in the corpus.



(a) (b) (c)

Figure 8. Noisy text segments selected based on value of c: (a) c=0.5 (b) c=1.0 (c) c=1.5.

Table 2. Character and word recognition rates for various values of "c".

	c<0.5	c=0.5	c=1.0	c=1.5
Character recognition rate (%)	82.93	82.93	81.26	80.98
Word recognition rate (%)	71.33	71.33	70.10	67.38

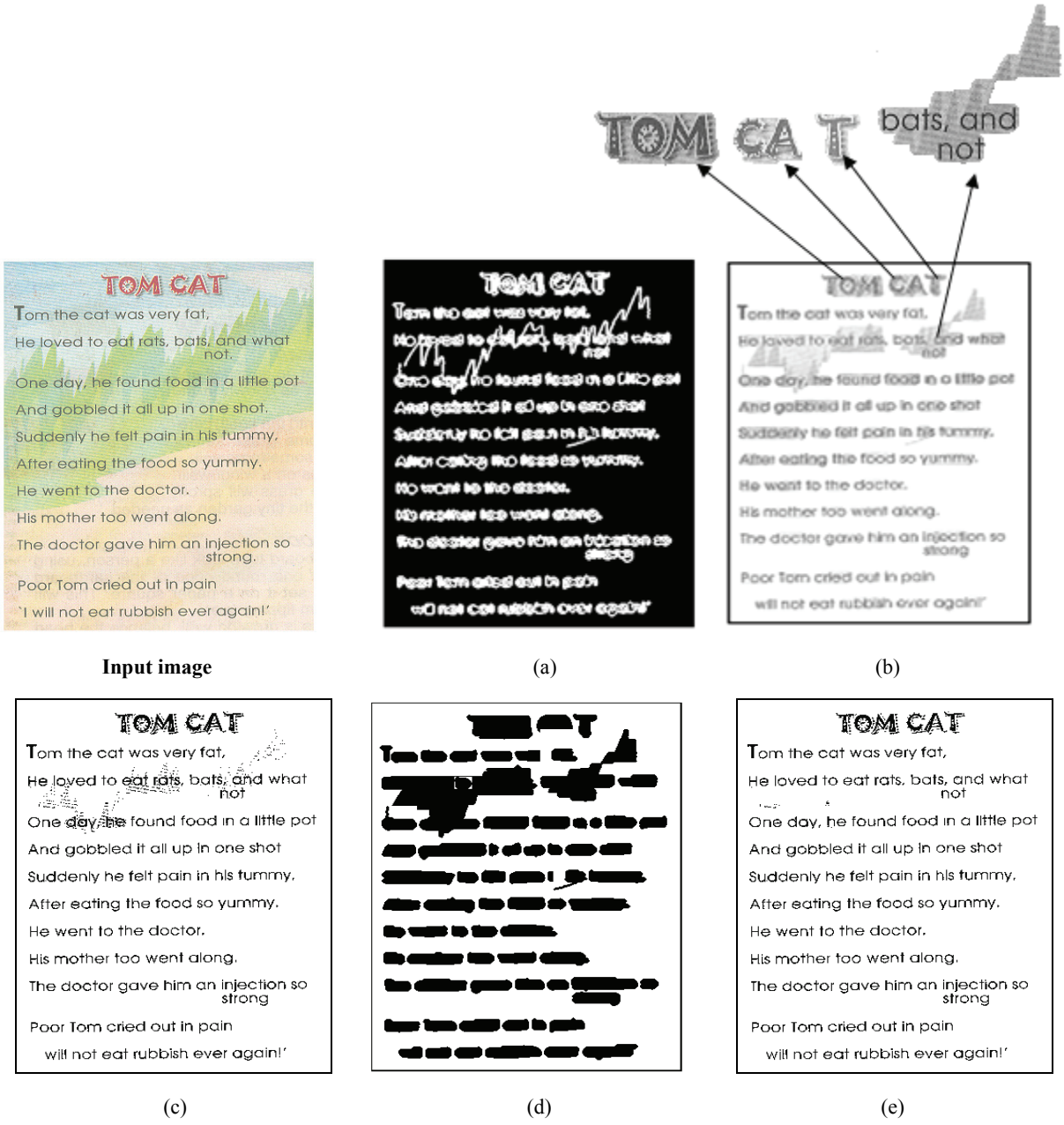


Figure 9. Result at each stage using proposed algorithm. (a) candidate text regions, (b) verified text regions, (c) extracted foreground text with noisy areas, (d) detection of noisy text segments (c=0.5), (e) extracted foreground text after reprocessing noisy text segments.

The proposed approach focuses on documents with English as text medium because we could quantify the performance of the improvement in the readability of document images by employing an OCR. Reading of the extracted text for documents in English as text medium is evaluated on Readiris 10.04 pro OCR. Readiris OCR converts the input document image to binary form before recognizing the characters. The Readiris 10.04 pro OCR can tolerate a skew of 0.5 degrees on foreground text. Readability of the segmented foreground text is evaluated in terms of character and word recognition rates. OCR results for document images with printed English text are tabulated in Table 5. From Table 5 it is seen that average recognition rate at character level is higher compared to word level.

Observations from the experimental evaluation are as

Table 3. Details of document collection used for this work.

Document types	Language	Background complexity	Foreground complexity
1) Pages from Magazines 2) Pages from Story books of children 3) Postal envelopes 4) Articles from newspapers 5) Power point slides 6) Journal cover pages 7) Invitation cards	Mainly in English. Also in other languages	1) Uniform patterned background 2) Non uniform patterned background 3) Background designs from Microsoft power point 4) Single and multicolored background	1) Single colored and multicolored text 2) Text tilted in any orientation 3) Text of varying sizes 4) Foreground with dense text and sparse text

Table 4. Results showing text region detection.

	Documents in English language	Documents in Kannada language	Documents in Malayalam language
Number of samples	180	30	10
Total number of characters	31784	4710	1068
Total number of words	6354	1200	317
Recall (%)	97.06	96.26	100
Precision (%)	96.78	95.1	90.23

$$\text{Character (or word) recognition rate} = \frac{\text{Number of characters (or words) correctly recognized}}{\text{Total number of characters (or words) in source document image}}$$

Table 5. OCR results for English documents.

	Average Recognition Rates (%)		
	Original document	Processed document	After further processing the noisy areas in the processed document
Character level	42.99	80.31	82.93
Word level	36.47	67.55	71.33

follows:

- For some document images the readability by the OCR without using our approach is 100% and the same is maintained even after applying our approach. [The proposed approach has not deteriorated the readability!].
- For rest of the documents due to high complexity of the background the readability through OCR is very low or even nil. After applying our approach the readability of document contents by OCR is improved to nearly 100%.

From Table 5 it is evident that the word and character recognition rates are enhanced after applying our approach. Further, it can be noted that readability is further improved after reprocessing the noisy areas in the output document images.

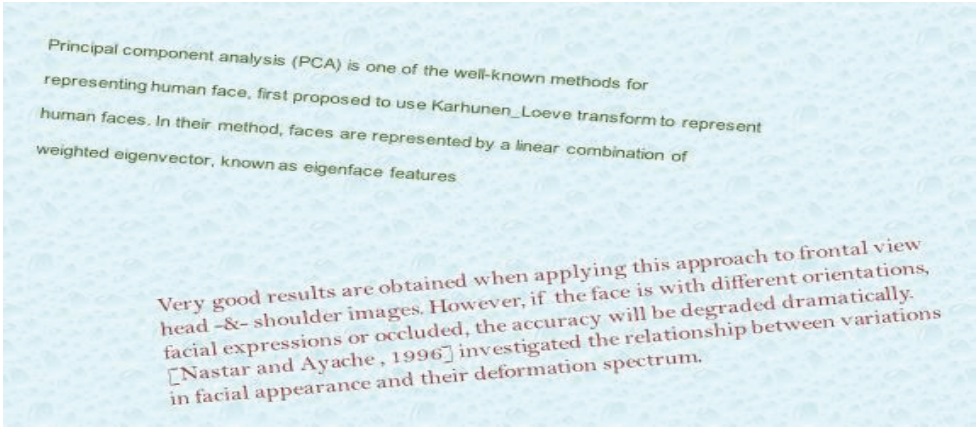
Many times the text lines in a document are tilted /rotated as an attempt to make the contents of the document more attractive. We extended our approach to extract the foreground text in document images with text lines tilted in any orientation. From experimental results it is evident that dilation of the edge image "E" in horizontal and vertical direction is sufficient to identify the text regions in document images having tilted text lines. Sample document images with foreground text lines titled in any orientation and the corresponding results are shown in Figure 10.

For documents with English as medium of text, we were able to quantify the enhanced readability through OCR and for documents in other languages we verified the extracted foreground text by visual inspection of output images, which indicates successful segmentation of foreground text from complex background. Figure 11 shows results of the proposed approach for documents in Malayalam and Kannada languages.

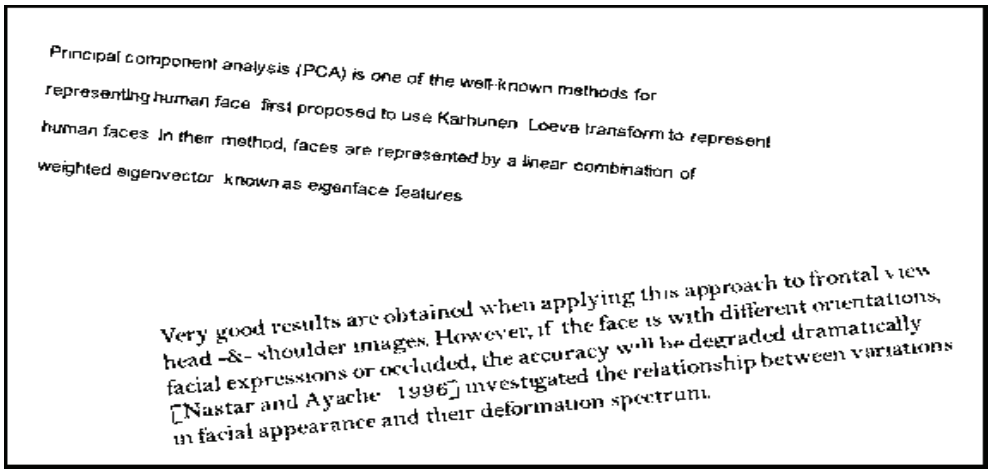
3.2. Discussions

Results of the proposed approach are compared with results of some existing methods of foreground separation in document images [6,9,13]. For a sample text rich document image and sparse text document image the output images obtained from the proposed method and other methods [6,9,13] are shown in Figure 12(a) and Figure 12(b) respectively.

From visual inspection of the results shown in Figure 12(a) and Figure 12(b) it is observed that, Niblack method fails to separate the foreground from complex background. Kasar method resulted in loss of foreground text information. Even though Sauvola method extracted foreground text, it introduced lot of noise compared to proposed method.



(a)



(b)

Figure 10. Document images with tilted foreground text: (a) Input color image, (b) Output image.

(a) (b) (c) (d)

Figure 11. Results of the proposed approach: (a,c) Input color images, (b,d) Output images.

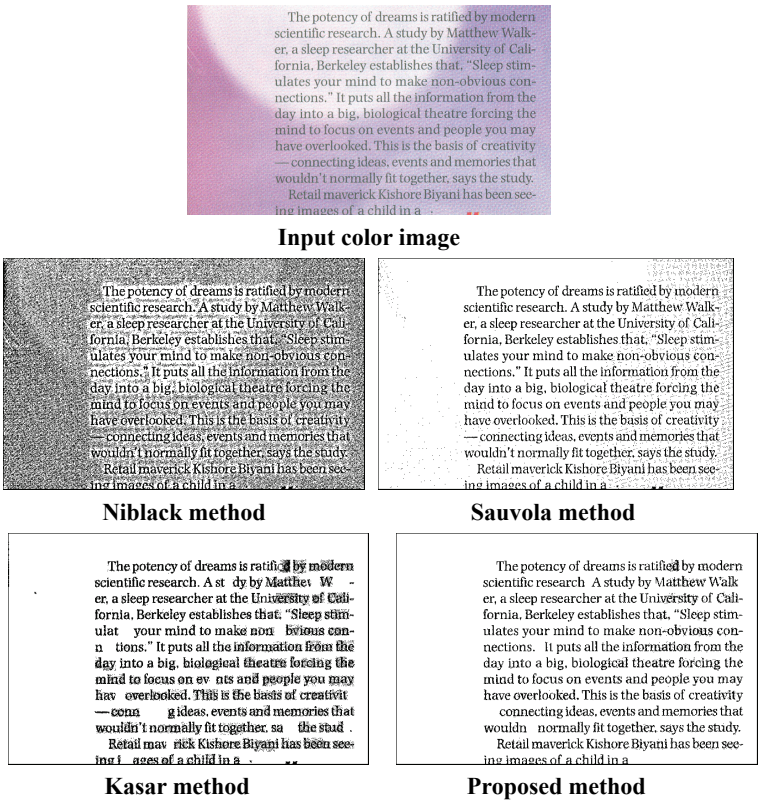


Figure 12(a). Comparison of results of foreground text separation from complex background in text rich document image.

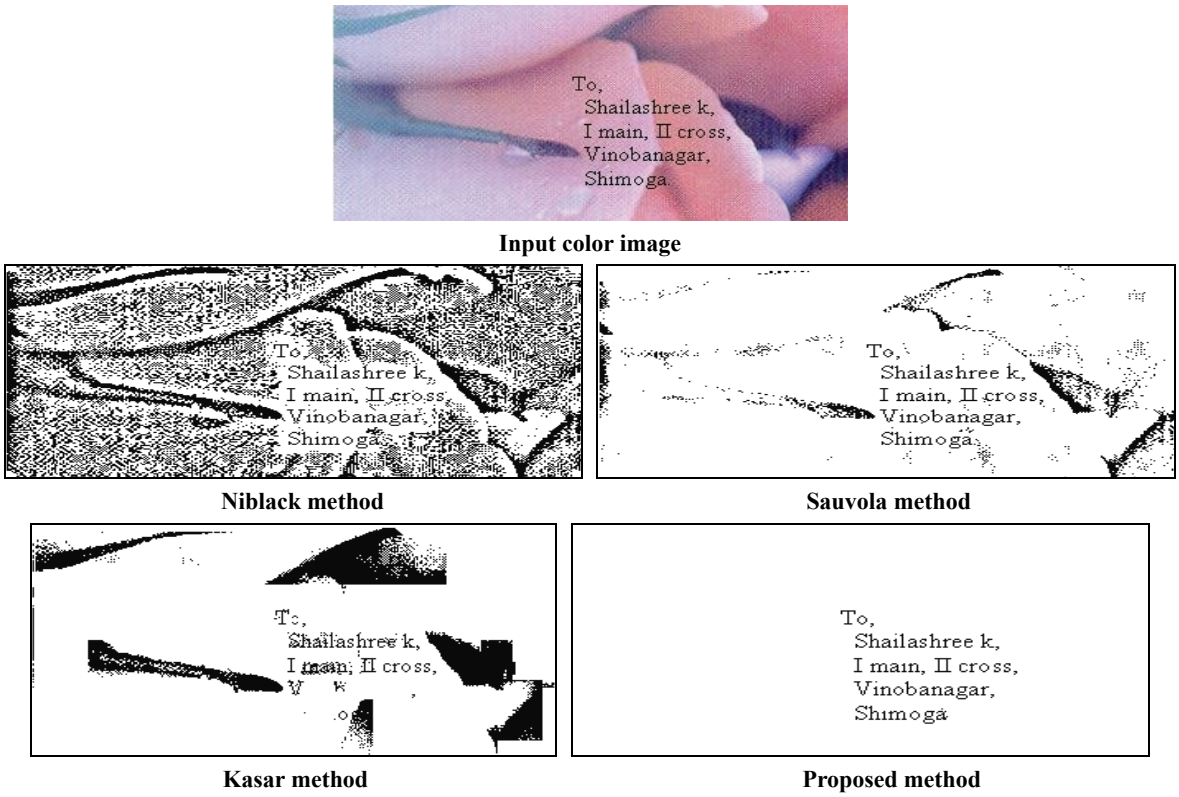


Figure 12(b). Comparison of results of foreground text separation from complex background in postal document image.

We created 20 ground truth images by selecting complex textures from Microsoft power point designs. On one set of 10 different backgrounds with varying complexities the same textual content of 540 characters is superimposed. Similarly on another set of 10 images created, each is superimposed with the same postal address shown in Figure 12(b). The outcome of the experiments on these ground truth images is shown in Table 6. As the output image produced by Niblack is too noisy the amount of characters recognized by OCR is very less. Kasar method fails to detect the foreground characters in document images having textured background which resulted in loss of text information present in the foreground of the input document image. This leads to very low character recognition accuracy by OCR. As the output images produced by Niblack, Sauvola and Kasar methods are noisier compared to the proposed method the amount of characters recognized by OCR is low which is evident from Table 6. These existing methods do not perform well when documents have textured/patterned background. This drawback is overcome by the proposed method. Yet in another experiment, a set of 10 typical document images from the corpus were tested with Niblack, Sauvola, Kasar and proposed method. The readability of extracted foreground text is evaluated on Readiris pro 10.04 OCR. The number of characters recognized by OCR is described in Table 7.

Our approach successfully separates the foreground in document images which are of low resolution and free from degradations such as blur, uneven lighting, and wavy patterned text. From Table 7 it is evident that our approach performs well for complex background color document images compared to the methods [6,9,13] and leads to higher character recognition accuracy through OCR.

One advantage of proposed method over the existing conventional approaches is it successfully extracts the foreground text without a prior knowledge of foreground and background polarities. Another advantage over existing methods is it is less expensive as it detects the image segments containing text and extracts the text from detected text segments without using the color information. The approach is independent of medium of foreground text as it works on edge information.

4. Time Complexity Analysis

Suppose size of the document image is MxN. Accordingly the size of RGB color image is 3xMxN. So the total number of pixels in input color image I is 3xMxN. The time complexity of the proposed algorithm in order notation is $O(N^2)$ if M=N. For the purpose of profiling

Table 6. Details of Foreground text extraction results on ground truth images by OCR.

Image type	Number of characters	Niblack method	Sauvola method	Kasar method	Proposed method
		CRR (%)	CRR (%)	CRR (%)	CRR (%)
Text rich document	540	27.89	89.63	76.33	98.53
Postal document	50	8.60	58.00	27.00	83.00

CRR–Average Character Recognition Rate when output image is OCRed.

Table 7. OCR based recognition of characters: Details for 10 test images with complex background.

Source of the document image	Number of characters	Niblack method	Sauvola method	Kasar method	Proposed method
		NCR (%)	NCR (%)	NCR (%)	NCR (%)
News paper	483	0	13.66	70.39	97.52
News paper	300	20	99.66	94.00	98.33
Magazine	144	0	0	0	92.36
Invitation card	748	0	98.93	95.45	95.32
Story book	300	0	96.66	44.66	98.66
Story book	440	0	99.09	51.59	99.55
Story book	139	0	97.12	73.38	100
Synthesized image	398	0	0	3.52	93.72
Postal doc.	47	0	0	51.06	100
Postal doc.	50	0	0	0	82
Average	**305**	**2**	**50.51**	**48.41**	**95.75**

NCR–Number of characters recognized by OCR.

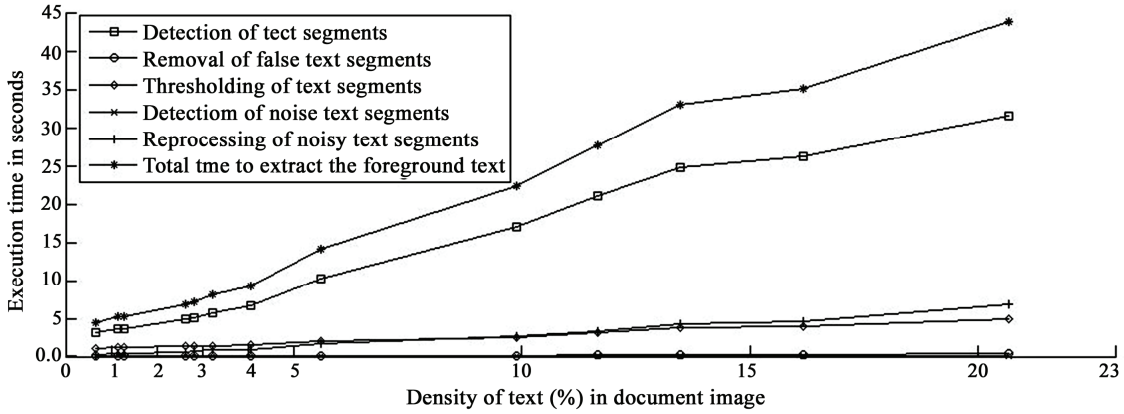


Figure 13. Plot showing execution time of each stage in proposed approach.

we have kept the size of all the test document images uniform (350x600 pixels). Postal document images contain text in sparse i.e., only printed postal address, on an average of 45 characters. Text rich document images contain text in dense, on an average of 200 characters. The algorithm was executed on a Intel(R) Core(TM) 2 Duo CPU, 2.20GHz, 1GB RAM. It is observed that total time needed to extract the foreground text is high for text rich document images compared to sparse text document images. Time needed to process the document depends on the amount of text in the foreground of the source document image. The total time of the entire process includes the time of I/O operations of document images also. Figure 13 shows the plot of execution time of each stage of the proposed approach for document images with varying density of textual information.

5. Conclusions and Future Work

In this paper a hybrid approach is presented for extraction of foreground text from complex color document images. The proposed approach combines connected component analysis and texture feature analysis to detect the segments of image containing text. An unsupervised local thresholding method is used to extract the foreground text in segments of image containing textual information. A simple and computationally less expensive method for texture analysis of image segments is proposed for reduction of false text regions. We have not used color information in extracting the text since in the first stage of our approach the evidence of all textual edges comes from intensity values in each color channel (RGB model) and this makes computations inexpensive. Threshold value to separate the foreground text is derived from the image data and does not need any manual tuning. The proposed algorithm detects on an average 97.12% of text regions in source document image. The shortcomings of the proposed approach are 1) it fails to separate the foreground text when the contrast between the foreground and background is very poor 2) it fails to detect single letter word which neither contains a hole nor creates a hole by the dilation.

The algorithm has so far been tested on text dominant documents only which are scanned from news papers, magazines, story books of children, postal envelopes. Also we tested the proposed approach on synthesized images. The behavior of the documents containing graphic objects in foreground is considered as future extension of the present work. Design of post processing steps to recover the missed single character words without holes is another future direction of the current study.

6. References

[1] A. K. Jain and S. K. Bhattacharjee, "Address block location on envelopes using Gabor filters," Pattern Recognition, Vol. 25, No 12, pp. 1459–1477, 1992.

[2] V. Wu, R. Manmatha, and E. M. Riseman, "Textfinder: An automatic system to detect and recognize text in images," IEEE PAMI, Vol. 21, No. 11, pp. 1224–1229, 1999.

[3] D. Chen, H. Bourland, and J. P. Thiran, "Text identification in complex background using SVM," Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 621–626, 2001.

[4] U. Garain, T. Paquet, and L. Heutte, "On foreground-background separation in low quality document images," International Journal of Document Analysis and Recognition, Vol. 8, No. 1, pp. 47–63, 2006.

[5] H. Hase, M. Yoneda, S. Tokai, J. Kato, and C. Y. Suen, "Color segmentation for text extraction," IJDAR, Vol. 6, No. 4, pp. 271–284, 2003.

[6] T. Kasar, J. Kumar, and A. G. Ramakrishnan, "Font and background color independent text binarization," Proceedings of 2nd International Workshop on Camera Based Document Analysis and Recognition, pp. 3–9, 2007.

[7] E. Kavallieratou and E. Stamatatos, "Improving the quality of degraded document images," Proceedings of 2nd International Conference on Document Image Analysis for Libraries, pp. 340–349, 2006.

[8] G. Leedham, Y. Chen, K. Takru, J. H. N. Tan, and L. Mian, "Comparison of some thresholding algorithms for text/background segmentation in difficult document images," Proceedings of 7th International Conference on Document Analysis and Recognition, pp. 859–864, 2003.

[9] W. Niblack, "An introduction to image processing," Prentice Hall, Englewood Cliffs, 1986.

[10] S. Nirmala, P. Nagabhushan, "Isolation of foreground-text in document images having known complex background," Proceedings of 2nd International Conference on Cognition and Recognition, pp. 99–106, 2008.

[11] N. Otsu, "A threshold selection method from gray level histograms," IEEE Transactions on Systems, Man & Cybernetics, Vol. 9, No. 1, pp. 62–66, 1979.

[12] M. Pietikäinen and O. Okun, "Text extraction from grey scale page images by simple edge detectors," Proceedings of the 12th Scandinavian Conference on Image Analysis (SCIA), pp. 628–635, 2001.

[13] J. Sauvola and M. Pietikäinen, "Adaptive document image binarization," Pattern Recognition, Vol. 33, No. 2, pp. 225–236, 2000.

[14] M. Sezgin and B. Sankur, "Survey over image thresholding techniques and quantitative performance evaluation," Journal of Electronic Imaging, Vol. 13, No. 1, pp. 146–165, 2004.

[15] K. Sobottka, H. Kronenberg, T. Perroud, and H. Bunke, "Text extraction from colored book and journal covers," IJDAR, Vol. 2, No. 4, pp. 163–176, 1999.

[16] C. L. Tan and Q. Yaun, "Text extraction from gray scale document image using edge information," Sixth International Conference on Document Analysis and Recognition, pp. 302–306, 2001.

[17] O. D. Trier and A. K. Jain, "Goal directed evaluation of binarization methods," IEEE PAMI, Vol. 17, No. 12, pp. 1191–1201, 1995.

[18] Y. Zhong, K. Karu, and A. K. Jain, "Locating text in complex color images," Pattern Recognition, Vol. 28, No. 10, pp. 1523–1536, 1995.

[19] K. I. Kim, K. Jung, and H. J. Kim, "Texture based approach for text detection in images using support vector machines and continuously adaptive mean shift algorithm," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 25, No. 12, pp. 1631–1639, 2003.

[20] Y. Liu, S. Goto, and T. Ikenaga, "A robust algorithm for text detection in color images," Proceedings of Eighth International Conference on Document Analysis and Recognition, pp. 399–403, 2005.

[21] Z. Wang, Q. Li, S. Zhong, and S. He, "Fast adaptive threshold for Canny edge detector," Proceedings of the SPIE, pp. 501–508, 2005.

