International Journal of Scientific & Engineering Research Volume 10, Issue 6, June-2019
ISSN 2229-5518

Extracting Text Information from Digital Images

Md. Mijanur Rahman, Mahnuma Rahman Rinty and Fouzia Risdin

Abstract— Extracting text from an image is a difficult task. The main purpose of this project work is to develop a system that extracts text and its related information from digital images. Text on images contains useful data or information for annotation, indexing, and structuring of images. However, the problem of text extraction extremely challenging for variations of text on images. Various systems have been proposed in the past for detection and extraction of text from images. The main aim is to extract the character data from the digital images. This paper proposed effective algorithms that have been used to extract characters from printed paper or scanned document. The proposed system scans the text by evaluating each and every line. The system achieved more accurate results for normal documented image as well as with the scanned pages and has been implemented using C/C++ programming paradigm in Linux platform.

Index Terms— Adaptive Thresholding, Canny Edge-Detection, Morphological Image Processing, OCR, Text Extraction.

1 Introduction

Today the most information is available either on paper or in the form of photographs or videos. Large information is stored in images. The major feature of this system is to extract and display text with its information in text format from the image format. There are various approaches for extraction. These approaches have been proposed for specific application. Various sources of image with text are available like: image with caption text, image with shaded background, some images with text are also different with their color, alignment size etc. [1]. This variation makes the problem very difficult to draw automatically.

In this paper an improved system is proposed which extracts text in images. The system takes colored images as input. The main objective of the system is to automatic the task of text extraction and displays its information. So it is useful to detect, extract and recognize text from digital images and automates the task of extracting text document, or metadata from digital images, to make the proper use of image database retrieval. The Key Features of the Proposed System are Grayscale conversion, Canny-edge detection, Adaptive thresholding, Morphological Operation, Character (also word and line) block detection, Optical Character Recognition.

2 Previous works

Various methods have been proposed in the past for text extraction from images. Some of these include page segmentation, text color extraction, video frame text detection and content-based image or video indexing [1].Various methodologies that have been used for extracting text from images are: converting colored image to grayscale, binarization, connected components, horizontal and vertical projections and reconstruction [2]. On another paper, image segmentation and text extraction system included methodologies are frame extraction, image segmentation, image classification, text extraction and OCR [1]. Effective algorithms have been used to implement the methods on the field of text extraction. Another effective method is A Robust Algorithm for Text Detection in Images which can automatically detect, localize and extract horizontally aligned text in images (and digital videos) with complex backgrounds is presented [3]. Text information extraction from images and video in [4] includes methodologies like detection, localization, tracking, extraction, enhancement, and recognition of the text from a given image. The developed system in [4] is useful for describing the contents of an image, it can be easily extracted compared to other semantic contents, and it enables applications such as keyword-based image search, automatic video logging, and text based image indexing. So, taking into consideration of the previous works we have tried to design our system in a new platform and simple manner.

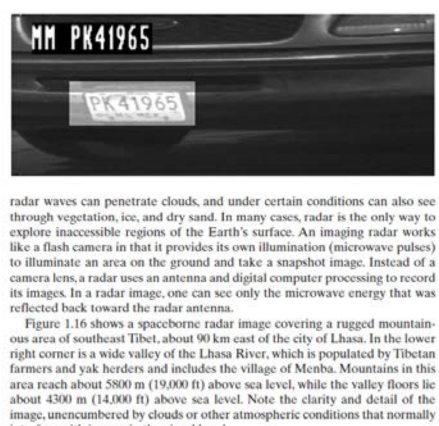
- *Corresponding Author: Md. Mijanur Rahman, Ph.D. Professor, Dept. of Computer Science and Engineering, Jatiya Kabi Kazi Nazrul Islam University, Bangladesh. Phone: +8801712594569. Email: mijanjkkniu@gmail.com*
- *Mahnuma Rahman Rinty is currently pursuing masters degree program in Computer Science and Engineering in Jatiya Kabi Kazi Nazrul Islam University, Bangladesh. E-mail: mahnumarinty@gmail.com*
- *Fouzia Risdin is currently pursuing masters degree program in Computer Science and Engineering in Jatiya Kabi Kazi Nazrul Islam University, Bangladesh. E-mail: fouzia.risdin@gmail.com*

3 Methodologies

3.1 Image Acquisition

Image acquisition is a process where an image is given that is already in digital form. This is the stage where the image under consideration is taken.

Image acquisition contains variation in the intensity levels along the image. So many noises are also added to the image. So the first task is required to denoise the image. It's called pre-processing [5]. The proposed system used documented image as input. Such type of sample image is shown in Fig.1.



radar waves can penetrate clouds, and under certain conditions can also see through vegetation, ice, and dry sand. In many cases, radar is the only way to explore inaccessible regions of the Earth's surface. An imaging radar works like a flash camera in that it provides its own illumination (microwave pulses) to illuminate an area on the ground and take a snapshot image. Instead of a camera lens, a radar uses an antenna and digital computer processing to record its images. In a radar image, one can see only the microwave energy that was reflected back toward the radar antenna.

Figure 1.16 shows a spaceborne radar image covering a rugged mountainous area of southeast Tibet, about 90 km east of the city of Lhasa. In the lower right corner is a wide valley of the Lhasa River, which is populated by Tibetan farmers and yak herders and includes the village of Menba. Mountains in this area reach about 5800 m (19,000 ft) above sea level, while the valley floors lie about 4300 m (14,000 ft) above sea level. Note the clarity and detail of the image, unencumbered by clouds or other atmospheric conditions that normally interfere with images in the visual band.

Fig. 1. Sample Documented Image

ISSN 2229-5518

3.2 Grayscale Operation

A digital color image includes color information for each pixel. There are various color models which are used to represent a color image. These are RGB color model, in which red, green and blue light is added together in various ways to reproduce a broad array of colors. The other models are CMY color model which uses cyan, magenta and yellow light and HSI model which uses hue, saturation and intensity variations.
Grayscale images contain many shades of gray without any observable color. [6]. In an 8 bit image, 256 shades are possible. The darkest possible shade black is represented as 00000000 and lightest possible shade white is represented as 11111111.

3.3 Canny Edge-Detection

Edge detection is the previous and major step before objects detection and segmentation. It is a necessary step to find the boundaries of the objects [7].

Canny Algorithm

It is a multi-step algorithm that can detect edges with noise suppressed at the same time [8]. It follows the following steps:

Step-1. Smooth the image with a Gaussian filter to reduce noise and unwanted details and textures.

$$g(m,n) = G\sigma(m,n) * f(m,n) \quad (1)$$

Where

$$G_\sigma(x,y) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp[-\frac{x^2+y^2}{2\sigma^2}] \quad (2)$$

Step-2. Compute gradient of g(m , n) using any of the gradient operators (Roberts, Sobel, Prewitt, etc) to get:

$$M(n,n) = \sqrt{g_m^2(m,n + g_n^2(m,n)}$$

and

$$\theta(m,n) = tan^{-1}[\frac{g_n(m,n)}{g_m(m,n)}]$$

Step-3. Threshold M:

$$M_T(m,n) = \begin{cases} M & if\ M(m,n)>T \\ 0 & otherwise \end{cases}$$

Where T is chosen that all edge elements are kept while most of the noise is suppressed.

Step-4. Suppressed non-maxima pixels in the edges in M_T obtained above to thin the edge ridges. To do so, check to see whether each non-zero $M_{T(m,n)}$ is greater than its two neighbors along the gradient direction $\theta_{(m,n)}$. If so, keep $M_{T(m,n)}$ unchanged, otherwise, set it to 0.

Step-5. Threshold the previous result by two different thresholds T_1 and T_2 (where $T_1 > T_2$) to obtain two binary images T_1 *and* T_2.

Step-6. Link edge segments in T_2 to form continuous edges. To do so, trace each segment in T_2 to its end and then search its neighbor in T_1 to bridge the gap until reaching another edge segment in T_2 .

3.4 Adaptive Thresholding

Thresholding is required for image segmentation. It distinguishes the image regions as objects or the background. Although the detected edges are consist of text edges and non-text edges in every detail component sub-band, it distinguishes them due to the fact that the intensity of the text edges is higher than that of the non-text edges.
An appropriate threshold value is selected and preliminarily removed the non-text edges in the detail component sub-bands. Adaptive thresholding employs to calculate the target threshold value T [9].
The target threshold value is obtained by performing an equation on each pixel with its neighboring pixels. Then it needs to employ two mask operators to obtain such an equation and then calculate the threshold value for each pixel in the detail sub-bands. Basically, the dynamic thresholding method obtains different target threshold values for different images. Each detail component sub-band is then compared with T to obtain a binary image. The threshold T is determined as follows:

$$\boldsymbol{T} = \frac{\sum(\boldsymbol{e\,s(i,j) \times s(i,j)})}{\sum \boldsymbol{s(i,j)}} \quad (3)$$

$$s(i,j) = Max(|g_1 ** es\{i,j\}|, |g_2 ** es\{i,j\}|) \quad (4)$$

$$g_1 = [-1\ \ 0\ \ 1],\ g_2 = [-1\ \ 0\ \ 1]^2 \quad (5)$$

In Eq. (4), denote two-dimensional liner convolution. After that, we can apply Eq. (3) to compute T and the binary edge image (e) is then given by the following equation.

$$e(i,j) = \begin{cases} 255, & if\ es(i,j)>T \\ 0\ \ , & otherwise \end{cases} \quad (6)$$

3.5 Long-Line Finding

Long line finding is also a part of image segmentation. The edge-detection block finds the edges in the intensity image. This process is very efficient to the Hough Lines block by reducing the image area over which the block searches for long lines [7].
Hough Transformation for finding straight lines algorithm finds all long-lines in the digital images [8]. The algorithm is described below.

Accumulate the straight line segments in gray-tone image **S** to accumulator A.

S[R,C] is the input gray-stone image.

NLINES is the number of rows in the image.

NPIXELS is the number of pixels per row.

A[DQ,THETAQ] is the accumulator array.

DQ is the quantized distance from a line to the origin.

THETAQ is the quantized angle of the normal to the line.

ISSN 2229-5518

```
procedure accumulator_lines(S,A)
{
 A := 0;
 PTLIST := NIL;
 for R := 1 to NLINES
  for C := 1 to NPIXELS   {
   DR := row_gradient(S,R,C);
   DC := col_gradient(S,R,C);
   GMAG := gradient(DR, DC);
   if  GMAG > gradient_threshold  {
    THETA := atan2(DR, DC);
    THETAQ := quantize_angle(THETA);
    D:= abs(C*cos(THETA)– R*sin(THETAQ));
    DQ := quantize_distance(D);
    A[DQ,THETAQ]:=A[DQ,THETAQ]+GMAG;
   PTLIST(DQ,THETAQ):=append(PTLIST(DQ,
                              THETAQ),[R,C]) ;
   }
  }
}
```

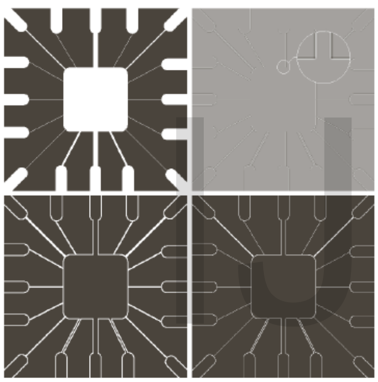


Fig. 2. Line Detection

3.6 Morphological Operation

Morphological image processing is a collection of non-linear operations related to the shape or morphology of features in an image. Morphological operation includes: erosion, dilation, opening and closing. Dilation operation is normally performed for enhancing image area that containing text.

Process of performing dilation operation

There are three kinds of edge regions intermixed with the text regions of grayscale image [9]. Overlapping occurs a lot after the morphological dilation operation. On the contrary, only one kind of edge region or two kinds of edge regions exist separately in the non-text regions and hence there is no overlapping even after the dilation. So, the bitwise AND operator causes to get the candidate text regions. There are also some non-text component regions which are too large or too small. By limiting the block size, we obtain the final text regions. Each text region has a moderate size w × h (pixels) in a candidate text region image.

Fig. 3. Dilated image of three binary regions.

(a) (b) (c)

Fig. 4. (a) Horizontal, (b) Diagonal and (c) Vertical edges dilation operators.

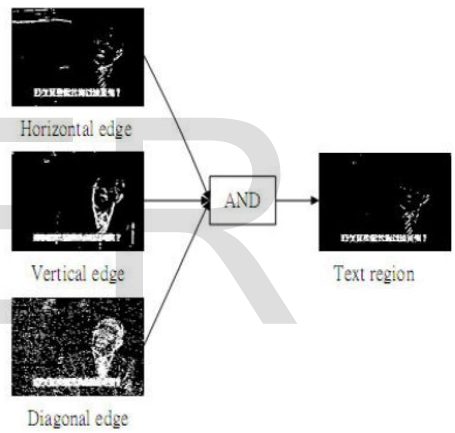


Fig. 5. Text extraction by using the bitwise AND operator.

The minimum text block size is determined as follows:
width>100pixels, height>35(pixel)
Removing the candidate text regions smaller than this limit.

Fig. 6. (a) The candidate text region (b) the extracted real text region.

3.7 Finding Line Block

It is the process of detecting each character block, word block, line block and paragraph block of text in the digital images. Each block is like a shape of a rectangle. This operation is per-

ISSN 2229-5518

formed by the Best Fitting Block algorithm.
The decomposition of an image into rectangular block is found first. To start with, the "best fitting block" foreground pixel is found first. This process is done by using a top-down scan. Then, a search is conducted for all "best fitting blocks" at all foreground pixels which belong to the "surrounding lines" of already detected blocks, i.e. are immediately adjacent to the perimeter of such blocks and belong to their exterior. This process is repeated iteratively, transforming the pixels of all detected blocks to background pixels, until no foreground pixel remains on the surrounding lines of all blocks of the current connected image component. There follows a complete description of the following algorithm [10]:

Step-1: Perform a top-down scan of the image to find a pixel at coordinates (x_0,y_0): $I(x_0,y_0) = 1$.

Step-2:
(a) Find the opposite vertex (x_{op},y_{op}) of the "best fitting block" at (x_0,y_0).
(b) Set BlockNum =1.
(c) Set $XF[1] = x_0$, $XL[1] = x_{op}$, $YF[1] = y_0$ $YL[1] = y_{op}$.
(d) Set $I(x, y) = 0 \; \forall \; x \in [XF[1] \ldots XL[1]] \wedge y \in [YF[1] \ldots YL[1]]$.

Step-3: Set iter =1.

Step-4: For every (x, y) belonging to the surrounding line of the block with opposite vertices at (X F(iter), Y F(iter)) and (X L(iter), Y L(iter) and obeying $l(x, y) = 1$:
a) Find the opposite vertex (x_{op},y_{op}) of the "best fitting block at (x, y).
b) Set BlockNum = BlockNum + 1.
c) Set XF[BlockNum] = x, XL[BlockNum] = x_{op}, YF[BlockNum = y, YL[BlockNum] = y_{op}.
d) Set $I(x,y)=0 \forall x \in [XF[BlockNum] \ldots XL[BlockNum]] \wedge y \in [YF[BlockNum] \ldots YL[BlockNum]]$.

Step-5: If iter < BlockNum then iter = iter + 1 and go to step 4.

Once the process has been completed, the coordinates of all blocks that form the first object of the image are stored in the matrices XF[i], X¸[i], YF[i], YL[i], for i = 1…BlockNum. The above process is repeated until all image objects have been segmented.

Fig. 7. Line segmented image of printed text.

3.8 Cropped Image

Cropping is the removal of the outer parts of an image to improve framing, accentuate subject matter or change aspect ratio [11].

Syntax of Cropping Image:

J = imcrop
J = imcrop(I)
J = imcrop(X,cmap)
J = imcrop(h)
J = imcrop(I,rect)
J = imcrop(X,cmap,rect)
J = imcrop(x,y,__)
[J,rect2 = imcrop(__)
[x2,y2,__] = imcrop(__)

Description:

- J = imcrop creates an interactive Crop image tool associated with the input image.
- J = imcrop(I) displays the image I in figure window and creates an interactive crop image tool.
- J = imcrop(X,cmap) displays the indexed image X in a figure using the colormap cmap and creates an interactive crop image tool.
- J = imcrop(h) creates an interactive crop image tool associated with the image specified by the handle h.
- J = imcrop(I,rect) crops the image I according to rect, which specifies the size and position of the crop rectangle [xmin ymin width height], in terms of spatial coordinates.
- J = imcrop(X,cmap,rect) crops the indexed image X with colormap cmap according to the vector rect, which specifies the size and position of the rectangle.
- J = imcrop(x,y,__) crops the image using a nondefault coordinate system, where x and y specify the image limits in the world coordinate system.
- [J,rect2 = imcrop(__) also returns the position of the cropping rectangle in rect2.
- [x2,y2,__] = imcrop(__) also returns the image limits in x2 and y2.

3.9 Optical Character Recognition (OCR)

Optical character recognition also called optical character reader (OCR) is the mechanical or electronic conversion of images of typed handwritten or printed text into machine encoded text, whether from scanned document.
In this project work, Tesseract OCR engine has been used for extracting text information. It is an open-source OCR engine. Usually OCR engine can only return characters, font, word, line or region information. We just need to add many other modules and tune them to get the best results.
No own page layout analysis is needed for Tesserat. It assumes its input is a binary image with optional polygonal text regions defined. The process of OCR has a step-by-step pipeline, which includes Line and Word finding, Word Recognition, Static Character Classifier, Linguistic Analysis and Adaptive Classifier.

4 Experimental Results and Disscussion

The proposed approach has been evaluated using digital image containing different types of text format. So before getting the final text from the image, some resulting images of different stages have been built and these results are shown in the following figures.

Fig. 8. Input image

Fig. 9. Canny Image

Canny Edge-Detection Result

As Canny algorithms gives most accurate result for edge detection so that we have used it. The following result in Fig.9 shows each object is segmented from the input image for closed region boundaries.

Fig. 10. Thresholded Image

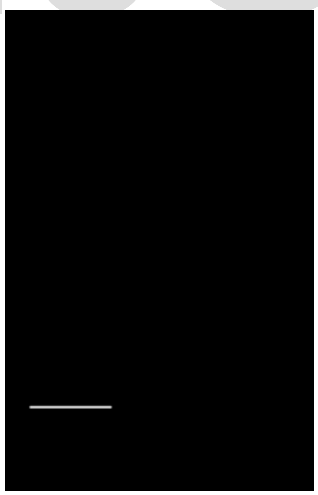


Fig.11. Long-line Found in the Image

.Adaptive Thresholding Result

Thresholding process is applied to discard unwanted pixel regions that are less than a calculated threshold value. The resulting thresholded image is shown in Fig.10.

Long-Line Finding Result

To remove unnecessary long line from the image long-line finding has been done by using Hough Transformation algorithm and the result is shown in Fig.11.

Non-edge Removing Result

After applying edge detection algorithm, it is necessary to remove the non-edges for extracting only texts from the image. The non-edge removing has been performed by Morphological (dilation) operation.

Fig. 12. Non-edge removed image

Fig.13. Block-extracted image

Block-Extraction Result

A rectangular block for each character is found at this stage. These blocks are symbolized by the teal color. The result is shown in Figure-13.

Finding Line-Block

After finding character blocks the line blocks are found by using "Best Fitting algorithm" where all rectangular character blocks in one line is determined as a single line block.

Finding Paragraph Block

After finding all line blocks, by adding all line-blocks when the system finds a full-stop or gap then it counts all lines as a para-block. The para-block image is shown in Fig.15.

Fig.14. Line-block found image

Fig.15.Para-blocks found image

Final Text Extraction Result

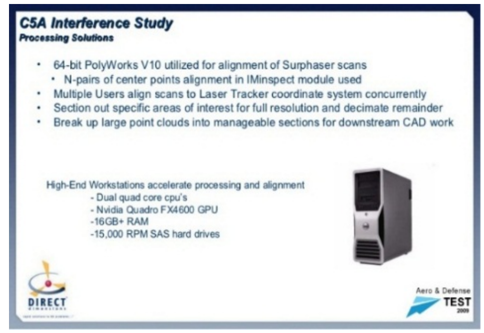
For a digital scanned copy image, the result is mostly accurate and all the texts have been extracted with very little noise. The final extracted text of the input image (see Figure-8) has been saved as text file and shown in Fig.-16.

Chapter 4
Rapid Serial Visual Presentation: Bilingual
Lexical and Attentional Processing
Jennifer M. Martin and Jeanette Altarrilm
This chapter examines the use of Rapid Serial Visual Prexemmion
Abstract
(RSVP) as a research method for studying reading and attention in bilinguals
Theoretical background and methodological considerations are provided for the
most common ways in which RSVP is used: lexical pmcessing, repetition blimlness
(R8), the attentional blink (AB). and executive control. The authors also describe
and discuss relevant studies that have used bilingual participants, whether exclu
sively or in comparison to monolinguals. To date, there has been relatively little
use of RSVP in bilingual research. However, this chapter provides rationale for its
use as a well-controlled experimental method that is especially well-suited for use
with bilinguals (whose reading speeds tend to vary a great deal). Suggestions for
future research are also provided
Introduction
9‘?
Throughout this volume. the various methods used to study bilingual reading and
related processes are presented along with discussions on how these methods inform
theory and research in bilingualism. The focus of this chapter is on rapid serial
visual presentation (RSVP). RSVP is a method in which letters, digits, or words are
presented one at a time for a designated brief period of time. For example, Fig. 4.1, below,
shows a sample experiment that uses an RSVP stream of eight items presented
serially for 100 milliseconds (ms) with no pause in between (0 ms interstimulus
interval; 181). The boxes in the figu reindica tewh att heparticipa ntwou ld beviewi ng
on the computer screen. proceeding chronologically from the upper left corner to
the lower right comer. Stimuli may comprise a list of items or a full phrase or sen
tence. The greatest strength of this method is flexibili tya ndcontr ol inmanipulati ng
J. Altarriba. PhD.
J.M. Marlin, M.A. (ET-S)
Dcpanment of Psychology, University at Albany. State University of New York
Social Science Building-Room 399 1400 Washington] Avenue. Albany. NY l2222, USA
e-mail: jmaninS @albany.cdu: .lAltartiba@albany.edu
Q‘ .0
:l.
E) Springer Science+Business Media New York 2016
RR. Hercdia cl al. (eds). Methods in Bilingual Reading7 Comprehension
Research. The Bilingual Mind and Brain Book Series I
DOI [0.1007/978-1«1939-29934 4
6|

Fig. 16. Extacted Text File (almost text data)

For a slide documented image (see Fig.17), the result is also quite accurate with less unwanted characters. The input image contains some figures and sysmbols.

Input Image



Extracted Text File

05A Interference Study
Processing Solutions
64-bit PolyWorks V10 utilized for alignment of Surphaser scans
N-pairs of center points alignment in lMinspect module used
Multiple Users align scans to Laser Tracker coordinate system concurrently
Section out specific areas of interest for full resolution and decimate remainder
Break up large point clouds into manageable sections for downstream CAD work
__—l
High-End Workstations accelerate processing and alignment
l
Dual quad core cpu‘s
Nvidia Quadro FX4600 GPU
1668+ RAM
15.000 RPM SAS hard drives
__4
a |
0 v;
Aero & Defense
TEST
DIRECT
linonctonl
nova tum». u ll noon- '
"UNI:

Fig.17. Text extraction result, including text and unwanted characters.

The proposed method is not suitable for all types of digital image like Fig.18. Because that image contains figure or picture like scenes more than text. The extraction results from such type of image files includes major unwanted characters or symbols. Therefore, the proposed system is best suite for scanned-copy and documented image files that contain text mostly as document.

Input Image

Extracted Text File

WAN
Wide Area Network
r—m
v ‘__._.’—— T
:1":
LI—
'13‘
'LII‘
l-LIZH
I-FIZH
rrfi
L1“
III III
14
MAN
Metropolitan
Area Network
LAN
Local Area Network
nmunffil‘iilil'I—“ffff.
mm ”H EW
‘4
E i
F.
JIIIIIMI

Fig. 18. Fault text extracted result, contains unwanted characters.

5 Conclusion

In this paper, the main focus is on extraction of text from digital images and displaying its information. The proposed system used documented images or scanned text book pages as inputs and produced better results. Current technologies do not work well for digital images that containing bent or arched text or image that contains pictures more than text. It also unable to remove undesirable text from images. In contrast, the result is more accurate with small number of unwanted characters for normal documented image as well as with the scanned pages. This system also introduced a new platform for text information extraction in Linux operating system. The future plan is to implement an intelligent information extraction system that will automatically skip all unwanted characters or symbols.

Acknowledgment

The authors wish to thank JKKNIU and UGC research cell for their helpful guidances and useful supports.

References

[1] Danial Md Nor, Rosli Omar, M. Zarar M.Jenu, and Jean-Marc Ogier, *"Image Segmentation and Text Extraction: Application to the Extraction of Textual Information In Scene Images"*, International Seminar on Application of Science Mathematics 2011, ISASM2011, University of La Rochelle, France.

[2] Victor Wu, R. Manmatha, Edward M. Riseman, *"Text Finder: An Automatic System to Detect and Recognise Text in Images"* , IEEE Transactions on Pattern Analysis and Machine Intelligence 21(11):1224 – 1229, December 1999.

[3] J Gllavata, R Ewerth and B Freisleben, *"A Robust Algorithm for Text Detection in Images"*, Proceedings of the 3rd International Symposium on Image and Signal Processing and Analysis 2003 (ISPA2003), Italy.

[4] Keechul Junga1, Kwang In Kimb2, Anil K. Jainc3, *"Text information extraction in images and video: a survey"* The Journal of the Pattern Recognition Society, Pattern Recognition 37: 977 – 997, 2004.

[5] Santosh, Dr. Jenila Livingston L.M., *"Text Detection From Documented Image Using Image Segmentation"*, International Journal of Technology Enhancements and Emerging Engineering Research, Vol 1, Issue 4, 2013.

[6] Satish Kumar, Sunil Kumar, Dr. S. Gopinath, *"Text Extraction From Images"*, International Journal of Advanced Research in Computer Engineering & Technology, Vol 1, Issue 4, June 2012.

[7] R.C. Gonzales and R.E. Woods, Digital Image Processing, Addison-Wesley, Reading, 1992.

[8] A. Borji, and M. Hamidi, *"Evolving a Fuzzy Rule-Base for Image Segmentation"*, International Journal of Intelligent Systems and Technologies, World Academy of Science, Engineering and Technology, Vol 1, No 4, 2007.

[9] Adesh Kumar, Pankil Ahuja and Rohit Seth, *"Text Extraction and Recognition from an Image"*, Conference on Advances in Communication and Control Systems 2013 CAC2S2013.

[10] Stavros J. Perantonis, Basilios Gatos, Nikos Papamarkos, *"Block decomposition and segmentation for fast Hough transform evaluation"*, Pattern Recognition 32: 811-824, 1999.

[11] Zhang, M., Zhang, L., Sun, Y., Feng, L., Ma, W, *"Auto Cropping for Digital Photographs"* IEEE Conf. on Multimedia and Expo 2005, July 2005.

[12] Ray Smith, *"An Overview of the Tesseract OCR Engine"*, IEEE Ninth International Conference on Document Analysis and Recognition (ICDAR 2007), Parana, Brazil.