ISSN No. 0976-5697

Volume 6, No. 6, June - July 2015

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Empirical Performance Evaluation Methodology and its Application to Page Segmentation Algorithms

Pinky Gather
Department of Computer Science and Applications
Ch. Devi Lal University
Sirsa(Hry), India.

Avininder Singh
Department of Computer Science and Applications
Ch. Devi Lal University
Sirsa(Hry), India.

Abstract: Text extraction in an image is a challenging task in the computer vision. Text extraction plays an important role in providing useful and valuable information. Text line segmentation is a major component for document image analysis. Text in documents depend upon various factors such as language, styles, font, sizes, color, background, orientation, fluctuating text lines, crossing or touching text lines. The ascending approach to segmentation of scanned documents in the area of background, text, and photographs is considered. Such classification algorithms can also be used in the printing industry for selective or enhanced scanning and object-oriented rendering. We propose a page-layout-segmentation technique to extract text from scanned documents and also extract each character from the image.

Keywords: character extraction, document image, Image segmentation, page segmentation, X-Y Cut Technique text extraction.

I. INTRODUCTION

Document image segmentation to text lines and words is a critical stage towards unconstrained handwritten document recognition[1]. With the drastic advancement in Computer Technology & communication technology, the modern society is entering to the information edge. In change in the traditional document system (paper etc), people now follow electronic document system (PDF Format) for communication and storage which is currently imperative[2].

But on complex matters, the document image is difficult to accurately identify the information directly out of the need. On such cases preprocessing the document is done before its entry. Image segmentation theory, as digital image processing has become an important part of people active research. Image processing document image segmentation theory is an important research topic in the process it is mainly between the document image pre-processing and advanced character recognition an important link between. The relatively effective and commonly used for document image segmentation and classification methods include threshold, and geometric analysis and other categories[2].

After segmenting, Text part is detected and extracted for further process, earlier, text extraction techniques have been developed only on monochrome documents. These techniques can be classified as bottom-up, top-down and hybrid[2]. Here, we address the problem of locating the textual data in an image. Further, we have extended text extraction scheme for the segmentation of document images. Our text extraction scheme can identify and isolate textual regions in these kind of images. Such a system finds applications in image and text database retrieval, automated processing and reading of documents, and storing the documents in digitized form[3].

Segmentation accuracy determines the eventual success or failure of computerized analysis procedures[5]. The text character contain in the document image can be any gray scale value, low resolutions, variable size and embedded in complex background. Many problems encountered in the segmentation, these includes the difference in the skew angle between lines, characters or even along the same text line, adjacent text line, overlapping words and touching characters[5].

Page segmentation is the process to identify the areas of interest in the image of a document page. For a conventional document page with material printed in dark ink on a light colored paper, the areas of interest in the (binary) image will be neighbourhoods of black pixels. Page segmentation produces a description of the geometrical aspects of the areas of interest. The most common aspects are spatial extent and position on the page. Page segmentation can be thought of as a mapping from the pixel-based image data to a description of the areas of interest[6].

A. Image Segmentation:

It is the process of partitioning a digital image into multiple segments. The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze. Image segmentation is typically used to locate objects and boundaries in images. More precisely, image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share certain characteristics[7].

B. Current Image Segmentation Technique:

In recent years, a lot of research is done in the field of image segmentation process. There are currently thousands of algorithm, each doing the segmentation process slightly different from another, but still there is no particular algorithm that is applicable for all types of digital image, fulfilling every objective. Thus, algorithm developed for a group of images may not always apply to images of another class. Currently image segmentation approach, based on two properties of an image, is divided into two categories:

a. ***Discontinuities based:*** In this category, subdivision of images are carried out on the basis of abrupt changes in the intensity of grey levels of an image. It is based on identification of isolated points, lines and edges. This include image segmentation algorithms like edge detection.

b. ***Similarities based:*** In this category, subdivision of images are carried out on the basis of similarities in intensity or grey levels of an image. It is based on identification of similar points, lines and edges. This includes image segmentation algorithms like thresholding, region growing, region splitting and merging[8].

Segmentation techniques can be classified into three main types which are: pixel based, edge based, and region based. The choice of a specific technique related to the complexion of the required application and the surrounding environments. A brief explanation of these techniques is presented in the following subsections.

a) ***Pixel based Segmentation:*** The simplest segmentations methods is the pixel based known also as point based or thresholding method. In this method image pixels are classified according to specific threshold values. Various algorithms are proposed for skin color detection , such as piecewise linear classifiers, Bayesian classifier, histogram, fuzzy clustering , Gaussian classifiers, histogram based thresholding, Neural Networks NNs.

b) ***Edge based Segmentation:*** Edge based or known as boundary-based segmentation methods indicate the segmenting of the image depending on the edges between the segmented regions through finding the connection between edge pixels to form a contour. This can be performed either automatically using some edge detection filters, or manually by dragging the mouse to create boundaries between the segmented regions. Examples of edge detection filters are Prewitt's filter, Laplacian of Gaussian filter, watershed segmentation algorithm, and canny edge detector .

c) ***\Region based Segmentation****:* In this method the image is divided into groups of similar pixels that share some amenities. Region-based methods follow the principle of similarity values of the neighbouring pixels in the same region by comparing each pixel with neighbouring pixels to determine the region it belongs to according to similarity conditions. The segmentation process in region-based methods uses the feature image rather than the original image, where the feature image is represented by the regions classified by the segmentation process, however these methods are affected with the noise. Some region based skin classifications are: region growing, region merging, and region-splitting algorithms [9].

II. PROPOSED ALGORITHM

A. *Image Acquisition :*

In first step, where the image is taken as input. In the case of online recognition system a specialized hardware is implemented as explained earlier whereas for offline systems, the images are obtained either through a scanner or a camera. Whenever an image is acquired, there will be some variations in the intensity levels along the image. Also noise gets added to the image. Hence preprocessing is required for adjusting the intensity levels and to de noise the image.

B. *Preprocessing:*

In the second step, we perform the reprocessing that is the most important part of a better performing recognition system. Here, the acquired image is processed to remove any noise that may have incurred into the image during the time of acquisition or during the time of transmission. A colored image then it will be converted to a gray image before proceeding with the noise removal procedure. The de noised image is then converted to a binary image with suitable threshold.

C. *Segmentation :*

In the third step, we perform the segmentation that refers to a process of partitioning an image into groups of pixels which are homogeneous with respect to some criterion. Segmentation algorithms are area oriented instead of pixel oriented. The result of segmentation is the splitting up of the image into connected areas. Thus segmentation is concerned with dividing an image into meaningful regions. Image segmentation can be broadly classified into two types:

a. ***Local Segmentation*****:** It deals with the segmenting sub images which are small windows on a whole image.

b. ***Global segmentation:*** It deals with the images consisting of relatively large number of pixels and makes estimated parameter values for global segments more robust.

D. *Text Extraction:*

Text extraction refers to the extraction of text present in the scanned document. After the segmentation we can detect the textual part in the scanned document and extract the text which is display in the command window. In this step it can remove non textual data and only display the textual data.

E. *Character Extraction:*

Character extraction refers to extract the character in the scanned document. In this step each textual part in the image is detected and shown in green colour that shows that it contain only text and all other noise is removed and it can also remove small objects i.e. less than 30 pixel which is not relevant to the text its just a random data. In this connected component is extracted as a single character. After that all character is extracted and make the individual image for each character which is thousand in counting.

After implementation, we will be comparing our results along with the results given in base paper .We will consider using the precision rate before and after preprocessing for all the images involved.

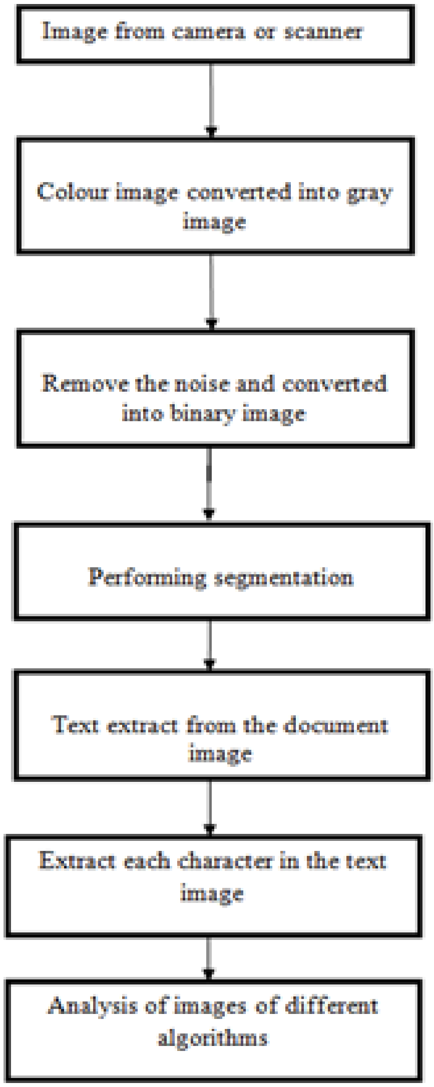
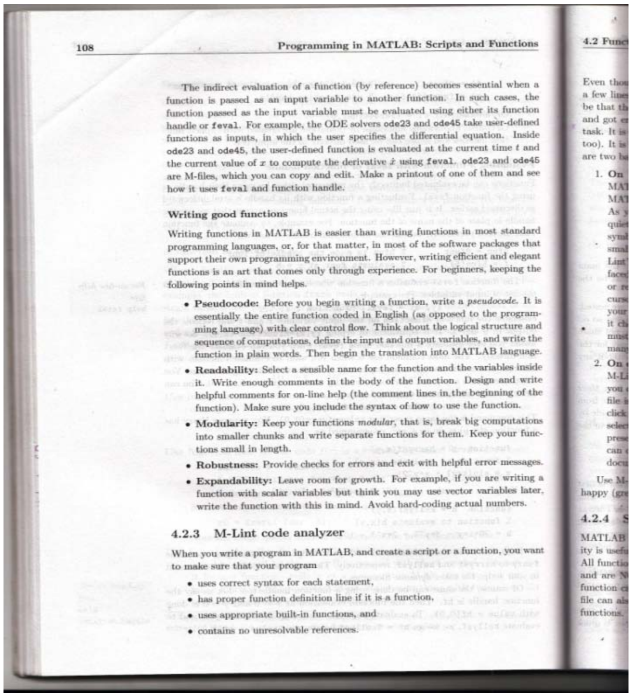


Image from camera or scanner
Colour image converted into gray image
Remove the noise and converted into binary image
Performing segmentation
Text extract from the document image
Extract each character in the text image
Analysis of images of different algorithms

Figure.1 Research methodology

III. RESULT AND DISCUSSION

We first input an image like below. This image has got a lot of noise associated with it. Our job would be to detect all this noise, rectify it and then present the same image noise free. After that we perform the paragraph segmentation.



108 Programming in MATLAB: Scripts and Functions

The indirect evaluation of a function (by reference) becomes essential when a function is passed as an input variable to another function. In such cases, the function passed as the input variable must be evaluated using either its function handle or `feval`. For example, the ODE solvers `ode23` and `ode45` take user-defined functions as inputs, in which the user specifies the differential equation. Inside `ode23` and `ode45`, the user-defined function is evaluated at the current time t and the current value of x to compute the derivative $\dot{x}$ using `feval`. `ode23` and `ode45` are M-files, which you can copy and edit. Make a printout of one of them and see how it uses `feval` and function handle.

Writing good functions

Writing functions in MATLAB is easier than writing functions in most standard programming languages, or, for that matter, in most of the software packages that support their own programming environment. However, writing efficient and elegant functions is an art that comes only through experience. For beginners, keeping the following points in mind helps.

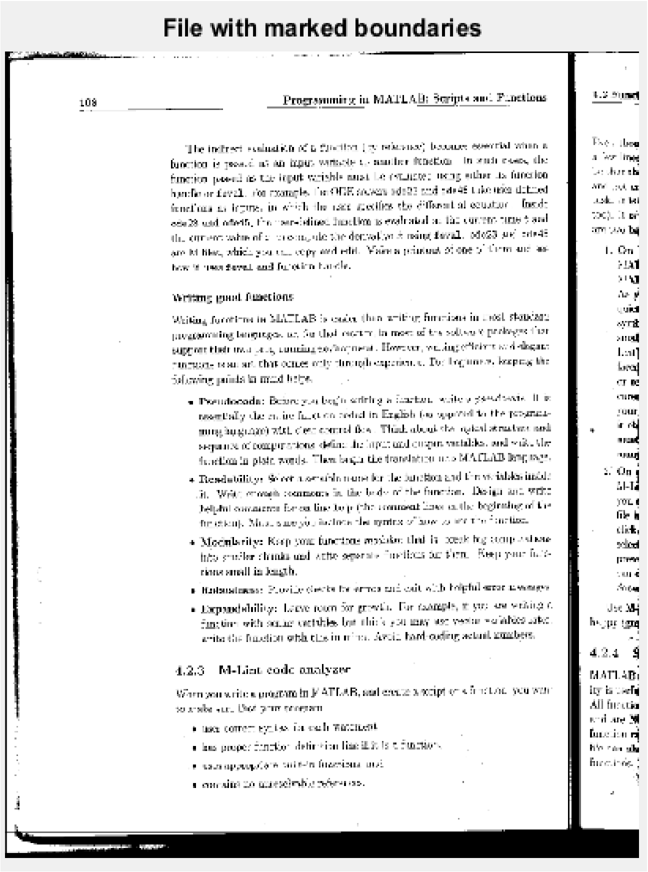
- **Pseudocode:** Before you begin writing a function, write a *pseudocode*. It is essentially the entire function coded in English (as opposed to the programming language) with clear control flow. Think about the logical structure and sequence of computations, define the input and output variables, and write the function in plain words. Then begin the translation into MATLAB language.
- **Readability:** Select a sensible name for the function and the variables inside it. Write enough comments in the body of the function. Design and write helpful comments for on-line help (the comment lines in the beginning of the function). Make sure you include the syntax of how to use the function.
- **Modularity:** Keep your functions *modular*, that is, break big computations into smaller chunks and write separate functions for them. Keep your functions small in length.
- **Robustness:** Provide checks for errors and exit with helpful error messages.
- **Expandability:** Leave room for growth. For example, if you are writing a function with scalar variables but think you may use vector variables later, write the function with this in mind. Avoid hard-coding actual numbers.

4.2.3 M-Lint code analyzer

When you write a program in MATLAB, and create a script or a function, you want to make sure that your program

- uses correct syntax for each statement,
- has proper function definition line if it is a function,
- uses appropriate built-in functions, and
- contains no unresolvable references.

Figure.2 Original Image



108 Programming in MATLAB: Scripts and Functions

The indirect evaluation of a function (by reference) becomes essential when a function is passed as an input variable to another function. In such cases, the function passed as the input variable must be evaluated using either its function handle or feval. For example, the ODE solvers ode23 and ode45 take user-defined functions as inputs, in which the user specifies the differential equation. Inside ode23 and ode45, the user-defined function is evaluated at the current time t and the current value of x to compute the derivative ẋ using feval. ode23 and ode45 are M-files, which you can copy and edit. Make a printout of one of them and see how it uses feval and function handle.

Writing good functions

Writing functions in MATLAB is easier than writing functions in most standard programming languages, or, for that matter, in most of the software packages that support their own programming environment. However, writing efficient and elegant functions is an art that comes only through experience. For beginners, keeping the following points in mind helps.

- Pseudocode: Before you begin writing a function, write a pseudocode. It is essentially the entire function coded in English (as opposed to the programming language) with clear control flow. Think about the logical structure and sequence of computations, define the input and output variables, and write the function in plain words. Then begin the translation into MATLAB language.
- Readability: Select a sensible name for the function and the variables inside it. Write enough comments in the body of the function. Design and write helpful comments for on-line help (the comment lines in the beginning of the function). Make sure you include the syntax of how to use the function.
- Modularity: Keep your functions modular, that is, break big computations into smaller chunks and write separate functions for them. Keep your functions small in length.
- Robustness: Provide checks for errors and exit with helpful error messages.
- Expandability: Leave room for growth. For example, if you are writing a function with scalar variables but think you may use vector variables later, write the function with this in mind. Avoid hard-coding actual numbers.

4.2.3 M-Lint code analyzer

When you write a program in MATLAB, and create a script or a function, you want to make sure that your program

- uses correct syntax for each statement,
- has proper function definition line if it is a function,
- uses appropriate built-in functions, and
- contains no unresolvable references.

Figure.3 Document Image with Mark Boundaries

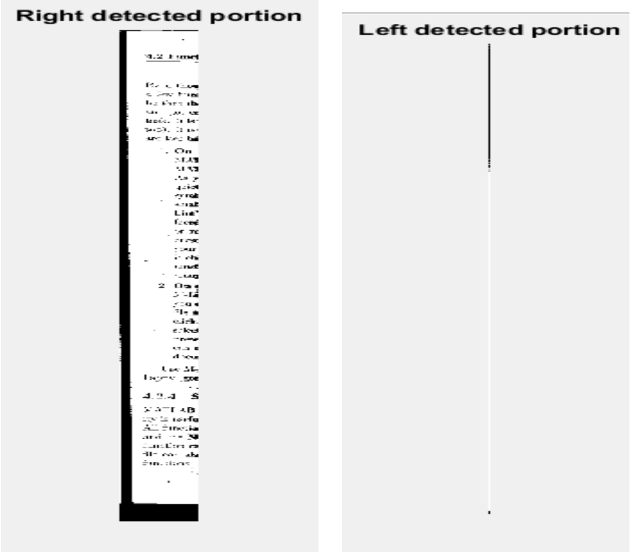


Figure.4 Noise detection at Left and Right portion

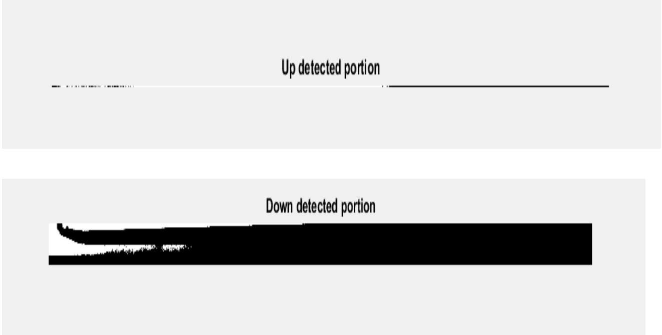


Figure.5 Noise detection at upper and lower portion

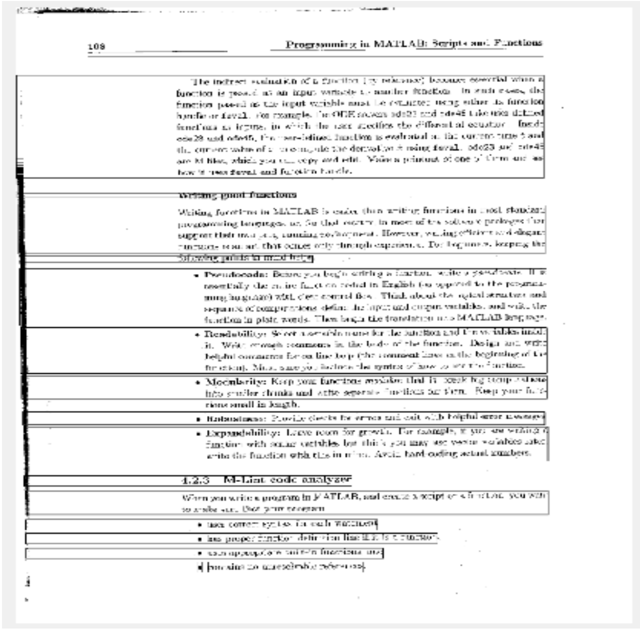
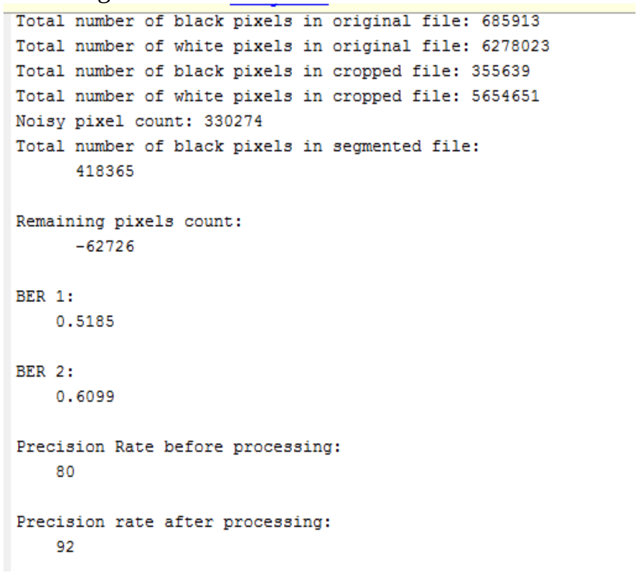

108
Programming in MATLAB: Scripts and Functions
Writing good functions
1.2.3 M-Lint code analyzer

Figure.6 Paragraph Segmentation

After segmentation, we calculate the total number of black pixel in image before removing the noise and the total number of pixel after removing the image ,which shows the decreasing of black pixel in the image effectively. Then, we calculate the bit error rate in cropped image and segmented image. After that, compare the precision rate before processing and after processing the image.

Following are the numerical results:



```
Total number of black pixels in original file: 685913
Total number of white pixels in original file: 6278023
Total number of black pixels in cropped file: 355639
Total number of white pixels in cropped file: 5654651
Noisy pixel count: 330274
Total number of black pixels in segmented file:
      418365

Remaining pixels count:
      -62726

BER 1:
    0.5185

BER 2:
    0.6099

Precision Rate before processing:
    80

Precision rate after processing:
    92
```

Figure.7 Numerical result of above document image

We can see that decrease in the number of black pixels in the output image is less than the original image. This also proves that noise was removed from the image successfully.

Summarize Data for different images that describe their precision rate before and after preprocessing.

Here we process the different scanned document images and calculate the precision rate before and after the preprocessing are given below:

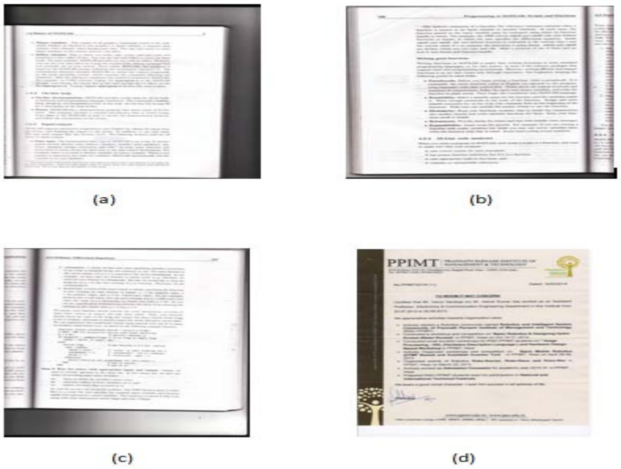


Figure.8 Different document images

Table I. Precision rate before and after preprocessing

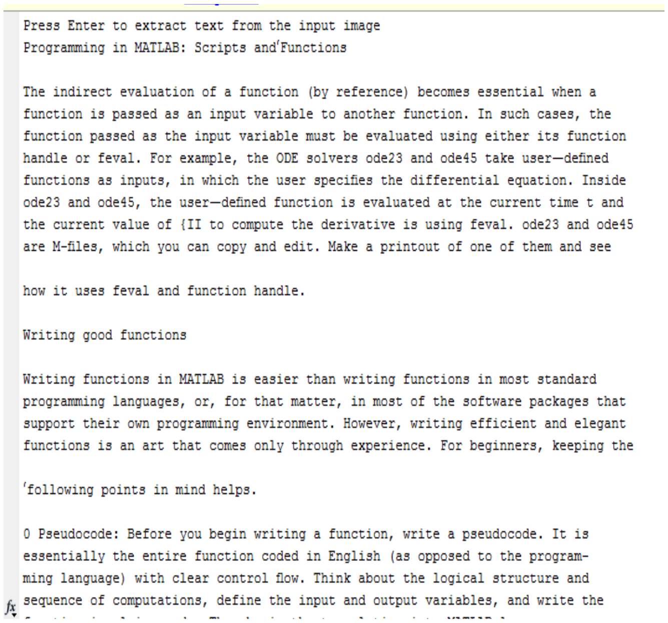
Image Number	Before Preprocessing	After preprocessing
Figure (a)	84%	92%
Figure (b)	80%	92%
Figure (c)	80%	92%
Figure (d)	80%	92%

A. Applications To Page Segmentation :

Now we implement the applications of page segmentation. By using the segmentation we can extract the text from scanned document image.

a. Text extraction:

In the text extraction, we have firstly detect the text from scanned document image and then the textual part is extracted from the document image and display at the command window.



```
Press Enter to extract text from the input image
Programming in MATLAB: Scripts and'Functions

The indirect evaluation of a function (by reference) becomes essential when a
function is passed as an input variable to another function. In such cases, the
function passed as the input variable must be evaluated using either its function
handle or feval. For example, the ODE solvers ode23 and ode45 take user—defined
functions as inputs, in which the user specifies the differential equation. Inside
ode23 and ode45, the user—defined function is evaluated at the current time t and
the current value of {II to compute the derivative is using feval. ode23 and ode45
are M-files, which you can copy and edit. Make a printout of one of them and see

how it uses feval and function handle.

Writing good functions

Writing functions in MATLAB is easier than writing functions in most standard
programming languages, or, for that matter, in most of the software packages that
support their own programming environment. However, writing efficient and elegant
functions is an art that comes only through experience. For beginners, keeping the

'following points in mind helps.

0 Pseudocode: Before you begin writing a function, write a pseudocode. It is
essentially the entire function coded in English (as opposed to the program-
ming language) with clear control flow. Think about the logical structure and
sequence of computations, define the input and output variables, and write the
```

Figure.9 Extracted text

b. Extraction of each Character:

Here, we have to implement the extraction for each character successfully. In this, we have to extract each character from the document image, firstly we detect the textual part of an image than show individual character that are present in text of scanned document. In the results, regions painted **green** indicate the text area

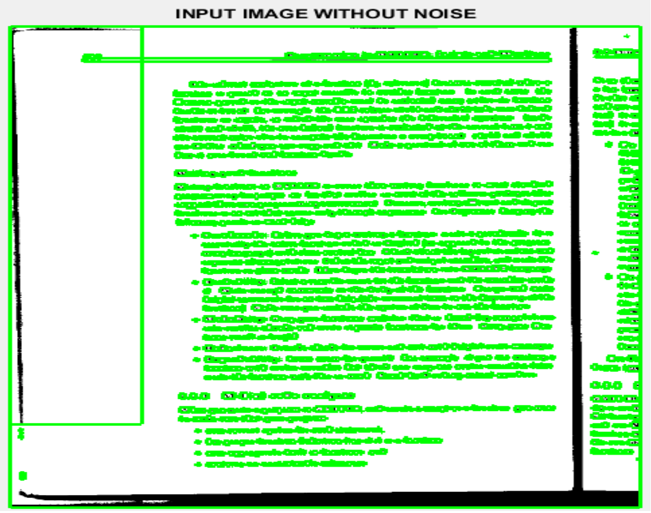


Figure.10 Detect the textual Part

Individual character that is present in document image which is nearly in thousands display in the specified folder. The characters are shown below:

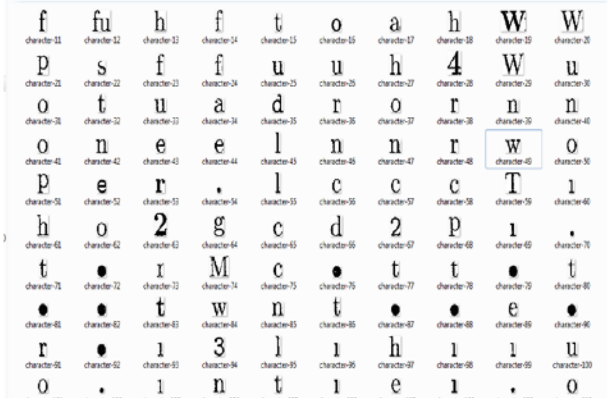


Figure.11 Show extraction of each character

IV. CONCLUSION

We present a relatively simple and effective algorithm for text detection and extraction. This new text extraction algorithm automatically detects and extract text from complex background images. From the above results, it can be concluded that we were successfully able to clear out noise from the scanned document. Also, numerical results show that percentage accuracy is greater than the base paper. This technique is very fast since only border pixels are accessed. The filtering and the feature extraction operations account for most of the required computations however, our method is very simple, computationally less expensive and efficient. Compared to other existing methods the dimensionality, and, so, the computation of the feature space, is considerably reduced. We have applied our algorithm on several structured and highly unstructured images with complex backgrounds and obtained encouraging results. We improved the speed of this technique with fast automatically identification of the connected point using window directions.

V. REFERENCES

[1]. Vassili Papavassiliou, Themos Stafylakis, Vassilis Katsourosa and George Carayannis, “Handwritten document image segmentation into text lines and words”, National Technical University of Athens, School of Electrical and Computer Engineers, pp.369-377,2010.

[2]. D.Sasirekha and Dr.E.Chandra, “Enhanced Techniques for PDF Image Segmentation and Text Extraction”, International Journal of Computer Science and Information Security,vol.10, no. 9,september 2012.

[3]. Sunil Kumar, Rajat Gupta and Nitin Khanna,” Text Extraction and Document Image Segmentation Using Matched Wavelets and MRF Model”, IEEE Transactions on Image Processing,pp.2117-2128 vol. 16, no. 8, august 2007.

[4]. Ankush Gautam,” Segmentation of Text From Image Document”, International Journal of Computer Science and Information Technologies, pp. 538-540,vol. 4 (3), 2013.

[5]. Santosh and Dr. Jenila Livingston L.M,”Text Detection From Documented Image Using Image Segmentation”, International Journal of Technology Enhancements and Emerging Engineering Research, pp.144-148,vol.1,2013.

[6]. Boontee Kruatrachue, Narongchai Moongfangklang and Kritawan Siriboon, “Fast Document Segmentation Using Contour and X-Y Cut Technique “,World Academy of Science, Engineering and Technology,pp.27-29,2007.

[7]. https://www1.imperial.ac.uk

[8]. Rohan Kandwal,Ashok Kumar and Sanjay Bhargava,“ Existing Image Segmentation Techniques”,International Journal of Advanced Research in Computer Science & Software Engineering,pp.153-156,vol.4,2014.

[9]. Mokhtar M. Hasan and Pramod K. Mishra, “Novel Algorithm for Skin Color based Segmentation using Mixture of GMMs”, Signal & Image Processing : An International Journal (SIPIJ), vol.4, no.4, pp. 139-148, 2013.

[10]. H.P B Narkhede,“ Review of Image Segmentation Techniques”,International Journal of Science & Modern Engineering,vol.1,pp.54-61,2013.

[11]. Mahesh T R, Prabhanjan S and M Vinayababu, “Noise Reduction By Using Fuzzy Image Filtering”, Journal of Theoretical and Applied Information Technology**,** Vol.15, No.2,2010.

[12]. Manpreet Kaur, Chirag Sharma, “Improved Method for Segmentation of Real-time Image of Printed Documents”, International Journal of Soft Computing and Engineering, Vol.4,pp.135-138, 2014.