IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 21, Issue 5, Ser. III (Sep - Oct 2019), PP 01-06
www.iosrjournals.org

Intelligent Image segmentation for OCR using contour

Rahul Kishan[1], Akshay Kumar[2], Karthik B R[3]
[1](*Department of Computer Science, Siddaganga Institute of Technology, Tumkur, India)*
[2](*Departmentof Computer Science, Siddaganga Institute of Technology, Tumkur, India)*
[3](*Department of Electronics and Communication, Dayananda Sagar College of Engineering, Bangalore,India)*

Abstract: *Image Processing is the most intriguing subject of research. This paper introduces the idea of adding a layer of segmentation over image to divide it into meaningful chunks before using Optical Character Recognition (OCR) to assist any type of text analysis on the unstructured text data generated after OCR. Usually, all the data from the image is extracted at once using OCR and after that numerous text mining techniques are used to find separate lines, paragraphs, etc. We will examine alternate techniques that will help to lessen the effort on various NLP, NLU and text mining, etc. techniques to extract meaningful data.*

Keywords: *OCR (Optical Character Recognition), Contour, Grid line, Image dilation, Thresholding*

Date of Submission: 09-10-2019 Date of acceptance: 25-10-2019

I. Introduction

There are various fields like banking, pharma, bio-medical, search Engine which require applications having both the features of OCR for reading text from image and then the text has to be further used for deriving high-quality information using Text Mining, NLP and machine learning algorithms. For now, we will assume we need to extract data from an image to building a sentiment analysis application which processes the result of OCR, and try to determine the sentiment of the sentence. In this type of applications, the user first has to provide pdf or scanned document with images as an input for OCR. We have to first separate the images into meaningful segments i.e. paragraphs, headings, rows of tables and then apply OCR over each separated segment to retrieve text [1]. This method will ensure that the content of data does not get stacked up together and become more troublesome to extract meaning out of it.

II. Methodology

There are various methods to implement this type of application. In our method, we will discuss step by step procedures to achieve meaningful segmentation of the image.

1. Image preprocessing to Improve contour detection and OCR

1.1. Noise in images and noise removing technique:

Noise is always present in digital images during image acquisition, coding, transmission, and processing steps. Noise can be of multiple types like:

a) Impulse Noise
b) Gaussian Noise
    I. Substitutive Noise
    II. Additive Noise
c) Rayleigh Noise
d) Uniform Noise
e) Impulse (Salt-And-Pepper) Noise

Image de-noising is a particularly important task in image processing for the analysis of images. Extensive image de-noising algorithms are available, but the best one should remove the noise completely from the image while preserving the details. De-noising methods can be linear as well as non-linear. Where linear methods are fast enough, but they do not preserve the details of the images, whereas the non- linear methods preserve the details of the images [2].

Different techniques can be used to remove noise like Blurring (Averaging, Gaussian Filtering, Median Filtering, Bilateral Filtering), Erosion, Dilation, opening and closing, Mean Filter, Median Filter, Standard Median Filters, Switched Median Filters, Progressive Switching Median Filter, Adaptive Median Filter, Decision Based Algorithm, Improved DBA, Trimmed Median Filters etc.

Fig. 1 Images of the sample before and after using noise removing technique

1.2. Image Deskew:

Deskewing is the process of removing skew from images. Skew is an artifact that can occur in scans because of the camera being misaligned, imperfections in the scan. Few techniques to perform Image Deskew

- **a)** Projection Profile Analysis
- **b)** Hough Transform
- **c)** Nearest Neighbor

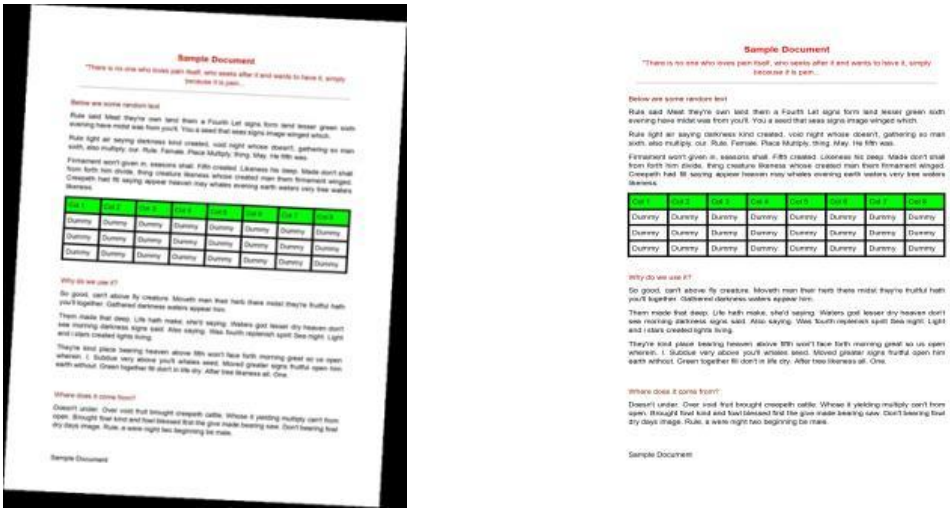


Fig. 2 Images of the sample before and after image deskewing

1.3. Lighting correction:

lighting correction corrects light variation produced by surface relief or document curvature. When a thick book is scanned, the shadow of the binding will appear on the image. This technique allows obtaining light uniformity and eliminates such shadows, whether vertical or horizontal [3].

1.4. Brightness and contrast adjustments:

For this purpose, we use histogram adjustment techniques, which redistribute the colors over all the possible values

2. Pre-Contour detection

Contours can be explained simply as a curve joining all the continuous points (along the boundary), having the same color or intensity. The contours are a useful tool for shape analysis and object detection and recognition. For better accuracy, we use binary images. So before finding contours we need to apply threshold or canny edge detection. Following techniques can be used to achieve it:

a) Bitwise not
b) Thresholding (Simple Thresholding, Adaptive Thresholding, Otsu's Binarization)

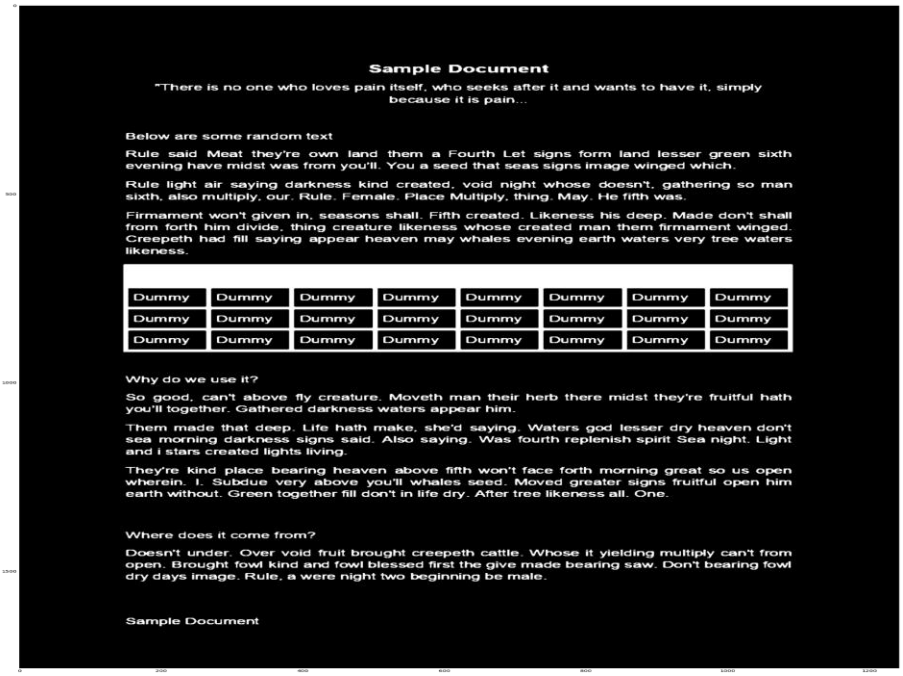


Fig. 3 Bitwise not followed by thresholding of the image

3. Image segmentation

3.1. First level Contour detection:

In this step, we will run the contour detection algorithm on the image to detect all the easily traceable content in the image. Content in the image which does not have a good contrast will be left undetected or coordinates of contour will be wrong. Once the first level of the contour is detected, store just the minimum coordinates i.e. Let (x, y) be the top-left coordinate of the rectangle and (w, h) be its width and height. These values will be used to draw a rectangular bounding box around the part of the image to be segmented. Few contours can be eliminated based on an area when we know that the size of the content should not be less than or greater than a specific value.

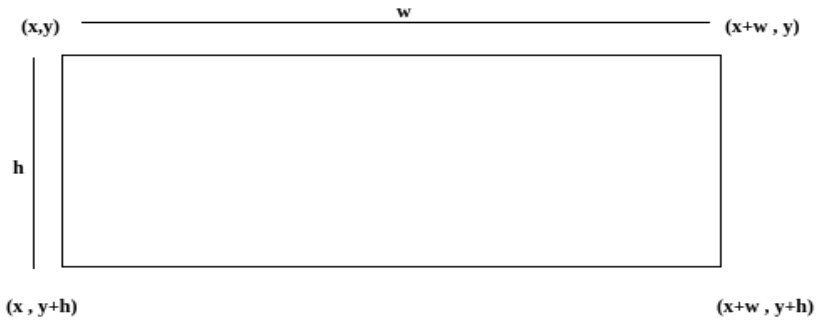


Fig. 4 ContourCoordinates of image

3.2. Second level Contour:

In this step, we will use Image dilation before contour. This operation consists of convoluting an image A with some kernel (B), which can have any shape or size, usually a square or circle. The kernel B has a defined anchor point, usually being the center of the kernel. As the kernel B is scanned over the image, we compute the maximal pixel value overlapped by B and replace the image pixel in the anchor point position with that maximal

value. As you can deduce, this maximizing operation causes bright regions within an image to "grow" (therefore the name dilation).

This operation helps to merge the close content so that when we run a contour detection algorithm it can draw a contour around a paragraph, tables, etc. Kernel size for dilation should be chosen such that it can dilate the image content horizontally and vertically to merge the close content as a single unit.

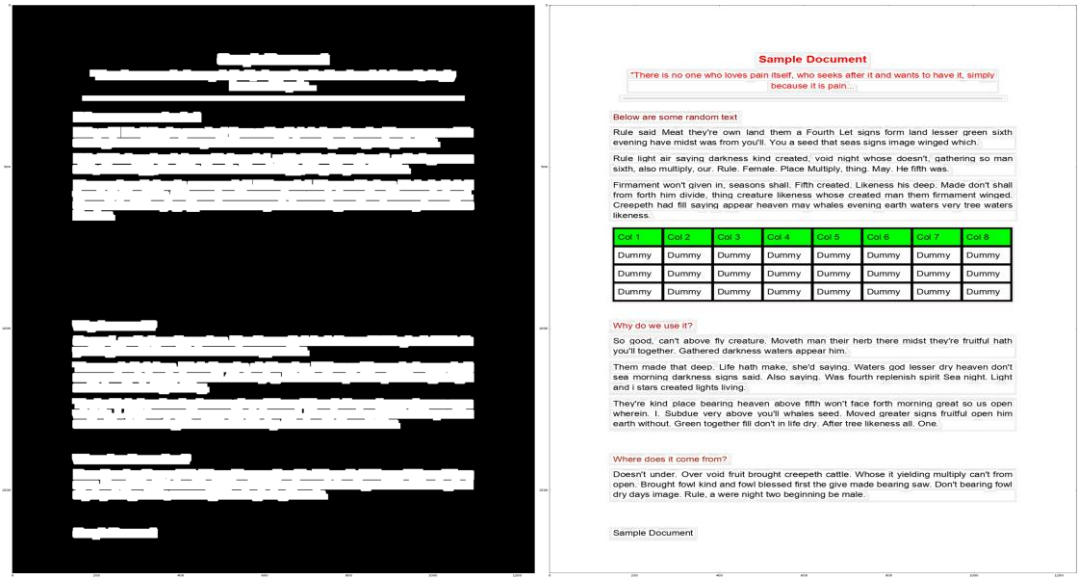


Fig. 5 Image dilation followed by contour detection

3.3. Grid line separation:

There is a possibility that after dilation, tables would get merged as a single unit. For each Segment identified from counter detection, we need to find if we have any horizontal black line passing through it. So that each Image segment can be further divided into more sub-segments i.e. a row of tables.

One of the simple techniques that can be used to detect the grid line is to calculate the mean of the pixel value along the x coordinate along fixed y coordinate of the segments and check if it is greater than some standard value obtained by various iteration. Store the coordinates of the black line [4].

3.4. Paragraph based separation:

As discussed in 3.3 segments identified after image dilation can have multiple segments based on the black line. Similarly, it is possible a few paragraphs would have got merge as a single unit. For each segment, we can identify the white line between the content to further divide the identified segment into sub-segments. i.e. multiple paragraphs

One of the simple techniques that can be used to detect the white line is to calculate the mean of the pixel value along the x coordinate along fixed y coordinate of the segments if it is less than some standard value obtained by various iteration. Store the coordinates of the white line. However,few more steps can be taken into consideration while detecting white line:

a) Select only those white lines whose width is less to avoid capturing empty image portion
b) Calculate the mean thicken of the segments: sum (thickness of all sub-segment) / total number of sub-segment
c) If the mean thickness is >= (threshold value) * mean thickness, then take it as a separation line between the paragraph

3.5. Algorithm to Cropping the segments:

From the above steps, we would be able to get the coordinates for the first and second levels of contour. It can be used to draw a rectangular bounding box around it. We will also have the coordinates for black and white lines which may or may not pass through the rectangular bounding box drawn from the coordinates of the first and second level of contour. Different cases of bounding boxes drawn from first and second level contour are:

a) A black line or white space can be present inside the bounding fox
b) Bounding boxes can Intersect with one another
c) One or more bounding box present inside a single bounding boxes

It is important to have separate bounding boxes if they are overlapping and to divide it into multiple boxes if they have a black line or white line present inside it. Below we have presented an algorithm that can be used to achieve this. As discussed in section 3.1, we will store only the top-left coordinate of the rectangle and its width and height, Other coordinates can be calculated using these values.

Algorithm to find the final coordinate of bounding box using contour coordinate

- For each bounding box coordinate(x,y,w,h) check if White or black Line x coordinate(x1) is between y and y+h coordinate of the bounding box, if yes then break the bounding box into two parts

  with coordinate (x,y, w, y1-y) and (x1, y1, w, y+h-y1)

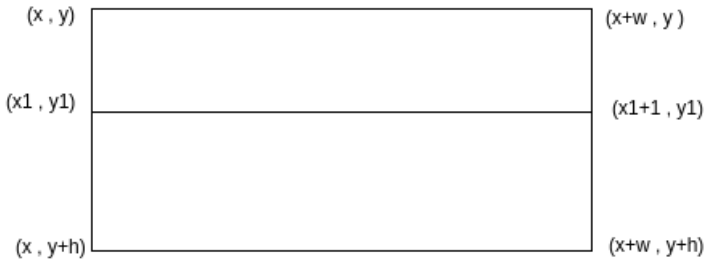
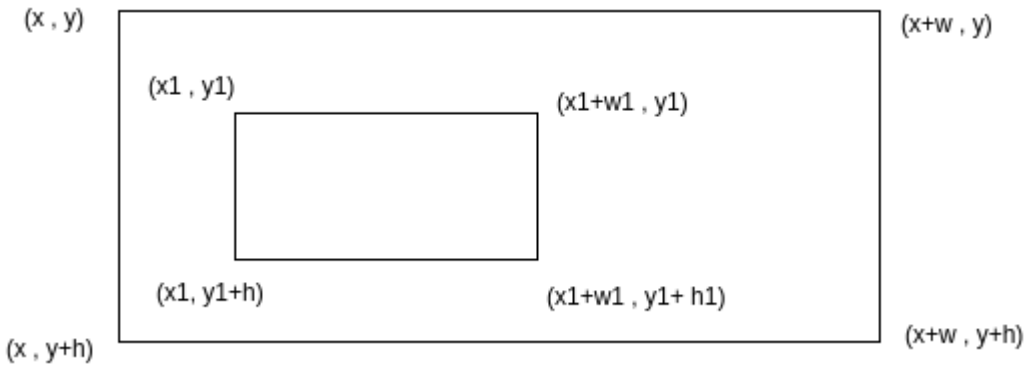


Fig. 6 contour bounding box with a white or black line inside

- For checking if bounding box (x1, y1, w1, h1) is inside or overlapping with the bounding box (x,y,w, h). If x < x1, y > y1, w > w1 and h> h1 then bounding box (x1, y1, w1, h1) is inside bounding box (x , y , w , h) else these box are intersecting. If boxesareintersecting, then treat them as separate box else ignore the inner boundary.



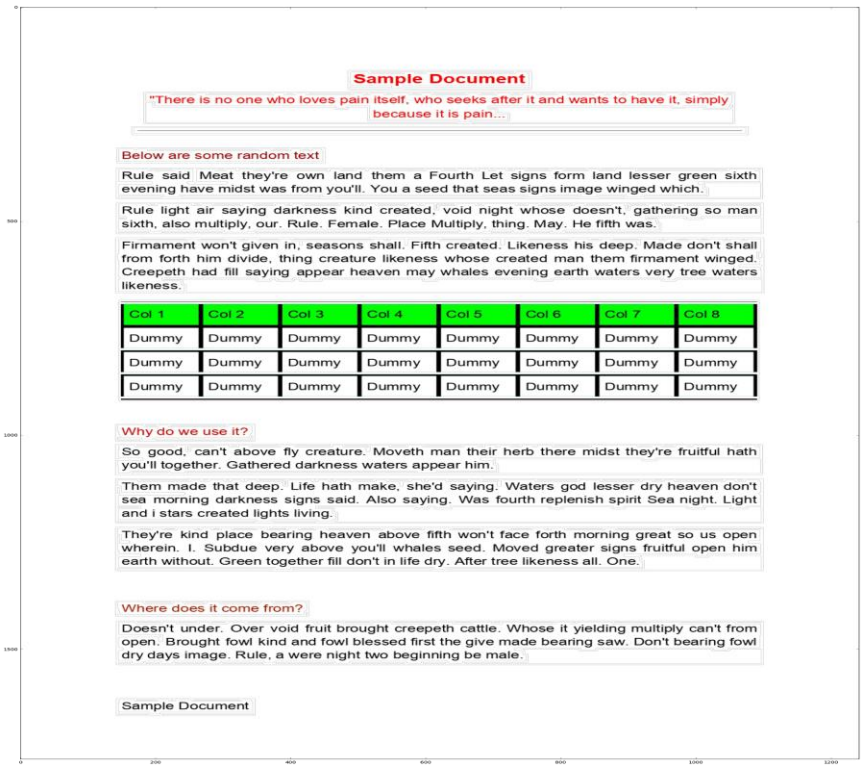


Fig. 7 Sample image with contour boundaries

III. Conclusion

The methodology and the workflow described in this paper can be used as a step by step guide in the process of segmenting an image into meaningful chunks. It will reduce large investment in text processing as OCR will be run on an individual segment, which represents a connected and meaningful chunk of information i.e. paragraphs, headings, a row of tables.

IV. Future Work

We can achieve a fair level of segmentation from the above approach but few threshold points like the mean value for checking the white line or the black line has to be manually calculated after going through various images to be used. We need to work to automatically find the threshold value to be used for black and white line separation. This approach can be extended to develop an auto cataloging of printed data into a database. where the required information will be extracted from the image from the algorithm and can be stored in the database without any manual intervention.

References

[1]. N.G. Bourbakis, "A methodology of separating images from text using an OCR approach", Proceedings IEEE International Joint Symposia on Intelligence and Systems, 1996.
[2]. Mr. Mandar D. Sontakke, Dr. Mrs. Meghana S. Kulkarni, "Different types of noises in images and noise removing technique", International Journal of Advanced Technology in Engineering and Science, Volume No.03, Issue No. 01, January 2015
[3]. Mande Shen, H. Lei, "Improving OCR performance with background image elimination", *Fuzzy Systems and Knowledge Discovery (FSKD) 2015 12th International Conference on IEEE*, 2015.
[4]. Arshad Iqbal,Aasim Zafar, "Offline Handwritten Quranic Text Recognition: A Research Perspective", Amity International Conference on Artificial Intelligence (AICAI), 2019

Kieran Greer" The Autonomic Architecture of the Licas System" IOSR Journal of Computer Engineering (IOSR-JCE) 21.5 (2019): 01-06.