A NEW APPORACH TO OPTICAL CHARACTER RECOGNITION BASED ON TEXT RECOGNITION IN OCR

Dr. Rajnesh Kumar (rajnesh.gcnagina@gmail.com)
Ramniwash Khatkar(ramniwashkhatkar007@gmail.com)
[1,2]Teaching Associates in CDLU, Sirsa
Mandeep Kaur (Mann.Mandeep Kaur52@gmail.com)

Abstract Optical Character Recognition (OCR) is a technology that enable of you to convert different types of documents, such as scanned paper documents, either hand written or machine printed script, PDF files or images captured by a digital camera into editable and searchable data. Our intention is to build an automatic text localization and extraction system which is able to accept different types of still images (or video frames) possibly with a complex background. This paper investigates methods for building an efficient application system for detecting text of any grayscale values embedded in images.

Keywords: OCR, Proposed Text Detection Method, Results for gray scale images.

1.Introduction

Optical Character Recognition is a process of scanning printed pages as images on a flatbed scanner and then using OCR software to recognize the letters as ASCII text. The OCR software has tools for both acquiring the image from a scanner and recognizing the text (Jain A. and Yu. B., 1998). OCR is available to recognize printed texts in widely used languages such as English, Chinese, and Japanese. These systems can process documents that are typewritten, or printed. They can recognize characters with different fonts and sizes as well as different formats including intermixed text and graphics. Text characters embedded in images and video sequences represent a rich source of information for content-based indexing and retrieval applications. However, these text characters are difficult to be detected and recognized due to their various sizes, grayscale values and complex backgrounds (Kasturi R. and Trivedi M., 1990). Both empirical image processing methods and statistical machine learning and modeling approaches are studied in two sub-problems: text detection and text recognition. Applying machine learning methods for text detection encounters difficulties due to character size, grayscale variations and heavy computation cost. To overcome these problems, we propose a two-step localization/verification approach. The first step aims at quickly localizing candidate text lines, enabling the normalization of characters into a unique size. In the verification step, a trained support vector machine or multi-layer perceptrons is applied on background independent features to remove the false alarms. Text recognition, even from the detected text lines, remains a challenging problem due to the variety of fonts, colors, and the presence of complex backgrounds and the short length of the text strings (Liang J., Phillips I., and Haralick R., 2001). Two schemes are investigated addressing the text recognition problem: bi-modal enhancement scheme and multi-modal segmentation scheme. In the bi-modal scheme, we propose a set of filters to enhance the contrast of black and white characters and produce a better binarization before recognition (Phillips I., Chen S., and Haralick R., 1993). For more general cases, the text recognition is addressed by a text segmentation step followed by a traditional optical character recognition (OCR) algorithm within a multi-hypotheses framework. In the segmentation step, we model the distribution of grayscale values of pixels using a Gaussian mixture model or a Markov Random Field (Sobottka K., Kronenberg H., Perroud T., and Bunke H., 2000). The resulting multiple segmentation hypotheses are post-processed by a connected component analysis and a grayscale consistency constraint algorithm. Finally, they are processed by OCR software. A selection algorithm based on language modeling and OCR statistics chooses the text result from all the produced text strings.

Steps in OCR

OCR steps are divided into two part, preprocessing and recognition.

Preprocessing steps:

a) Binarization,
b) Noise removing,
c) Skew detection and correction,
d) Line segmentation, Word segmentation, Character segmentation

Recognition steps:

e) Feature extraction, selection
f) Classification

2.The Proposed Text Detection Method

In this section, the processing steps of the proposed text localization approach are presented. Our intention is to build an automatic text localization and extraction system which is able to accept different types of till images (or video frames) possibly with a complex background.

The system design is based on the following assumptions:

(a) The input to our system can be a grayscale image;

(b) The current version can only detect texts with a horizontal alignment, and

(c) Texts that are smaller than a certain (small) font size will not be detected.

(d) If there is less part of text as compare the image part in image, will not be detected.

Step 1: Image Preprocessing.

In the image data is not represented continue pixels space, and image part is represented continue pixels space . In contrast to the approaches presented in our system only uses the luminance data during further processing. After that, luminance value thresholding is applied to spread luminance values throughout the image and increase the contrast between the possibly interesting regions and the rest of the image.

Step 2: Edge Detection.

This step focuses the attention to areas where text may occur. We employ a simple method for converting the gray-level image into an edge image. Our algorithm is based on the fact that the character contours have high contrast to their local neighbors. As a result, all character pixels as well as some non-character pixels which also show high local color contrast are registered in the edge image. In this image, the value of each pixel of the original image is replaced by the largest difference between itself and its neighbors (in horizontal, vertical and diagonal direction). Despite its simplicity, this procedure is highly effective. Finally, the contrast between edges will be increased by means of a convolution with an appropriate mask.

Step 3: Detection of Text Regions

The horizontal projection profile of the edge image is analyzed in order to locate potential text areas. Since text regions show high contrast values, it is expected that they produce high peaks in horizontal projection. First, the histogram H is computed, maximum number of black pixel is the number of pixels in line. These values stores in 2D array. The value of array is show the how many number of pixel set are stored.

In subsequent processing, the local maxima are calculated by the histogram determined above. Two thresholds are employed to find the local maxima. A line of the image is accepted as a text line candidate if either it contains a sufficient number of edges width or the difference between the edge’s width in one line to its previous line is small bit difference Both thresholds are defined empirically and are fixed. In this way, a text region is isolated which may contain several texts aligned horizontally.

A set of lines generated so far may not include all characters in BW. These characters, called missed, can be both inside and outside a line. Outside each line, rectangular regions on both sides (left and right) of it from regions of interest. Each region has the height equal to the height of the current line and the width equal to the maximal width of the bounding box among all components in this line. These parameters are not fixed but changing after inclusion of a component in the line. Line expansion stops when no component to be included is found.

In a later step, we define the x-coordinates of the leftmost and rightmost, top and bottom point of the text region. Finally, the exact coordinates for each of the detected areas are used to create bounding boxes.

Step 4: Enhancement and Segmentation of Text Regions.

First, geometric properties of the text characters like the possible height, width, width to height ratio are used to discard those regions whose geometric features do not fall into the predefined ranges of values. All remaining text candidates undergo another treatment in order to generate the so called text image where detected text appears on a simplified background. The binary edge image is generated from the edge image, erasing all pixels outside the predefined text boxes and then binarizing it. This is followed by the process of gap filling. If one white pixel on the binary edge image is surrounded by two black pixels in horizontal, vertical or diagonal direction, then it is also filled with black.

The gap image is used as a reference image to refine the localization of the detected text candidates. Text segmentation is the next step to take place. It starts with extraction of text candidates from the gray image.

2.1 Properties of Text Characters

We relied on the following well-known properties:

Property 1 Characters are normally arranged either horizontally or vertically.

Property 2 Given a fixed font type, style and size, heights of the characters belonging to the same group (groups include ascenders, descanters, capitals, and lower-cases) are approximately constant.

Property 3 Characters form regular (or weakly periodic)structures in both horizontal and vertical directions.

Property 4 Given a fixed font type, style and size, characters are composed of strokes of approximately constant width.

An input image is binary and text can be black on white background and white on black background within the same image. According to Property 1, we assume either horizontal or vertical text. The origin of coordinates is at the upper-left image corner and X-axis (Y-axis) is directed to the right (downwards) from the origin. Text of both orientations is first detected on white (normal) background, followed by text detection of both orientations on black (inverse) background.

2.2 Flow Chart of Proposed Method

This operation includes order-statistic filtering, followed by removing isolated black and white pixels. The order-statistic filtering replaces each pixel in BW by the *rth* Pixel in the sorted set of its neighbors in a 3x3 neighborhood. The obtained image is ANDed with BW and remaining isolated black and white pixels are removed from the image. As a result, we reduce the number of noisy pixels, while preserving character shapes as much as possible.

2.4Algorithm for Text Region Extraction

The procedure for extracting a text region from an image can be broadly classified into three basic steps:

(1) Detection of the text region in the image,

(2) Localization of the region, and

(3) Creating the extracted output character image

Steps:-

1. Convert the input image to binary color space. The luminance(Y) value is used for further processing. The output is a gray image.
2. Convert the gray image to an edge image.
3. Compute the horizontal and vertical projection profiles of candidate text regions using a histogram with an appropriate threshold value.
4. Use geometric properties of text such as width of text line to eliminate possible non-text regions. Text line area are maximum as compare to image or graph area.
5. Binarize the edge image enhancing only the text regions against a plain black background.
6. Create the Gap Image (as explained in the next section) using the gap-filling process and us this as a reference to further eliminate non-text regions from the output.
7. After that creates image part as a new image and text part also creates as a new image .

2.4.1 Step I: Image file is used as input:-

Printed document to be scanned as image file. That image file will be used as input for proposed approach. The image is taken as grayscale image. The image has only one Hindi word (text). The text may be straight or curved or skewed text. In this step flowing step is flow.

2.4.1.1 Read the file size and type of file:-

The image file has two part, header part and body part. In header part have all header information like file type of file, width, height, grayscale intensity, maximum intensity, type of file version also. In body part have the all image data. In this step, calculate the size of image, maximum intensity, body part. The body part is stored in 2D array. The size of 2D array is according to the calculated values of header part.

2.4.2 Binarization and Image filtering :-

The scanned images are in gray tone. Binarization is a process in which the gray scale images are converted to binary images. Binarization separates the foreground (text) and background information. The most common method for binarization is to select a proper threshold for the intensity of the image and then convert all the intensity values above the threshold to one intensity value (for example “white), and all intensity values below the threshold to the other chosen intensity (“black). In this step the image data is converted into binary form. This process is called binarization of image file. The image data is converted into two values, one is used for black pixels and another is used for while pixels

Image filtering

This operation includes order-statistic filtering, followed by removing isolated black and white pixels. The order-statistic filtering replaces each pixel in BW by the *rth* Pixel in the sorted set of its neighbors in a 3x3 neighborhood. The obtained image is ANDed with BW and remaining isolated black and white pixels are removed from the image. As a result, we reduce the number of noisy pixels, while preserving character shapes as much as possible.

2.4.3 ‘Black’ connected component detection:-This operation include the continues black pixel in the image .These pixels values stores in the 2D array .This steps totally depends upon the black pixel. The text area is less continues black pixel as compare to image part. In the 2D array the reference of array is represents the value of set of pixel. It is just like 0,1,2,3,4,5,6…………..etc. the at these addresses shows the how many time repeat these value of black pixel in one row

2.4.4 Horizontal text detection

The flow chart for this step is shown in Fig. 4.1. Processing starts by creating the 2D array in the pre steps. These values show the direction of text horizontally.1st the image is count as horizontally image and processing starts taking the property of horizontally text. In this step creates the boundary box around the text. These boxes have a set of black pixel according to horizontally.

2.4.5 Vertical text detection

Detection of vertically oriented text is similar to that of horizontally oriented. It, however, does not analyze those components that were already assigned to horizontal lines. Main changes are chiefly because of the fact that features related to ‘height’ are now interchanged with those related to ‘width’.

2.4.6 ‘White’ connected component detection In this part consider the white space between the text lines and the image or graphs. In subsequent processing, the local maxima are calculated by the histogram determined above. Two thresholds are employed to find the local maxima. A line of the image is accepted as a text line candidate if either it contains a sufficient number of edges width or the difference between the edge’s width in one line to its previous line is small bit difference . Both thresholds are defined empirically and are fixed. In this way, a text region is isolated which may contain several texts aligned horizontally.

In a later step, we define the x-coordinates of the leftmost and rightmost, top and bottom point of the text region. Finally, the exact coordinates for each of the detected areas are used to create bounding boxes.

2.4.7 Text part detection:-

First, geometric properties of the text characters like the possible height, width, width to height ratio are used to discard those regions whose geometric features do not fall into the predefined ranges of values. All remaining text candidates undergo another treatment in order to generate the so called text image where detected text appears on a simplified background. The binary edge image is generated from the edge image, erasing all pixels outside the predefined text boxes and then binarizing it.

A line of the image is accepted as a text line candidate if either it contains a sufficient number of edges width or the difference between the edge’s width in one line to its previous line is small bit difference . Both thresholds are defined empirically and are fixed. In this way, a text region is isolated which may contain several texts aligned horizontally. At last text part is separated from the image and saves as a new image which have only text part.

2.4.8 Image part detection:-

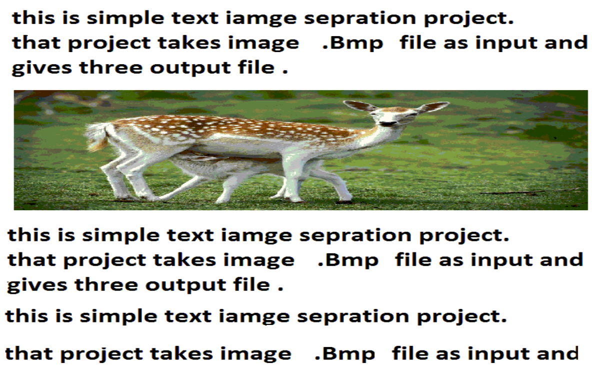
In the steps the image or graphically part is found out from image. In last steps creates the text part. Now to OCR the two image and get the image or graphic part.

3. Experimental Results for Gray Scale Images

The proposed approach has been evaluated using data sets containing different types of images. The whole test data consists of 326 images where 296 of them were extracted from various pdf files. These images can be further divided into two groups, based on the image:

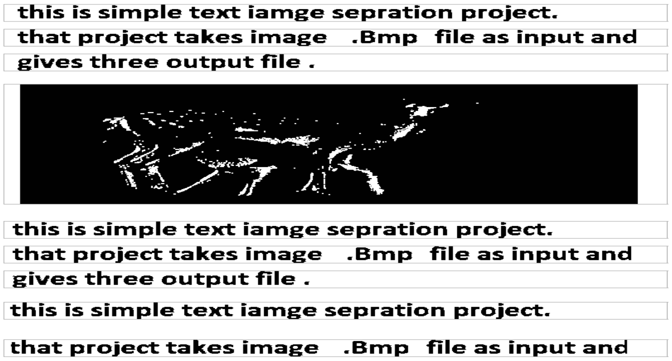
1. Sequences from different commercial advertisements (called the .commercials test set).

2. Sequences from different films, with a lot of text lines scrolling downwards, pre-title and credits title sequences (subsequently called the .credits test set).


this is simple text iamge sepration project.
that project takes image .Bmp file as input and
gives three output file .

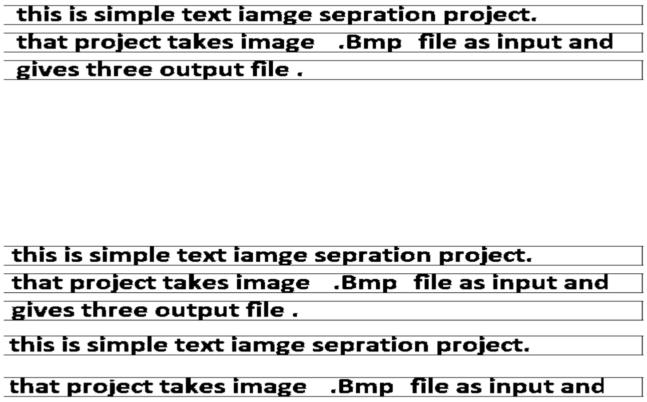
this is simple text iamge sepration project.
that project takes image .Bmp file as input and
gives three output file .
this is simple text iamge sepration project.
that project takes image .Bmp file as input and

Figure 2 Input Image


this is simple text iamge sepration project.
that project takes image .Bmp file as input and
gives three output file .

this is simple text iamge sepration project.
that project takes image .Bmp file as input and
gives three output file .
this is simple text iamge sepration project.
that project takes image .Bmp file as input and

Figure 3 Binarization and Boundary define Horizontal


this is simple text iamge sepration project.
that project takes image .Bmp file as input and
gives three output file .

this is simple text iamge sepration project.
that project takes image .Bmp file as input and
gives three output file .
this is simple text iamge sepration project.
that project takes image .Bmp file as input and

Figure 4 Only Text Part With Horizontally boundry

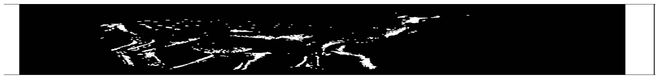
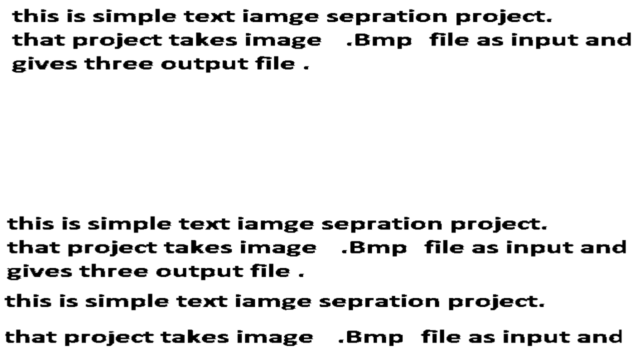


Figure 5 Only image Part With Horizontally boundry

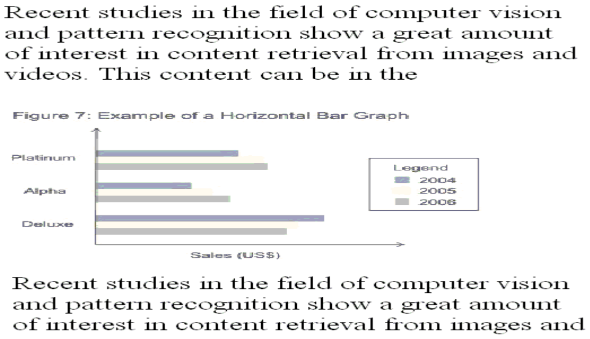
Image after Localization method

\


this is simple text iamge sepration project.
that project takes image .Bmp file as input and
gives three output file .

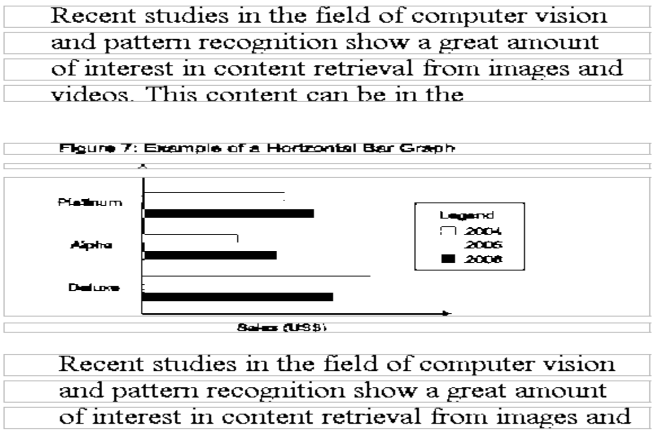
this is simple text iamge sepration project.
that project takes image .Bmp file as input and
gives three output file .
this is simple text iamge sepration project.
that project takes image .Bmp file as input and

Figure 6 Text Part Detection


Recent studies in the field of computer vision
and pattern recognition show a great amount
of interest in content retrieval from images and
videos. This content can be in the

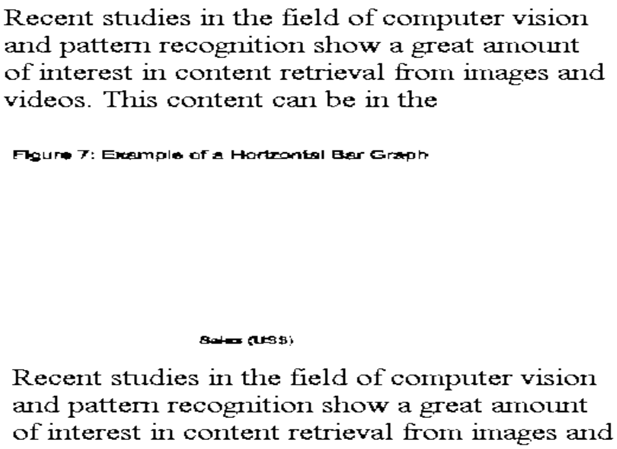
Recent studies in the field of computer vision
and pattern recognition show a great amount
of interest in content retrieval from images and

Figure 7 Input Image with Horizontal Bar Graph


Recent studies in the field of computer vision
and pattern recognition show a great amount
of interest in content retrieval from images and
videos. This content can be in the

Recent studies in the field of computer vision
and pattern recognition show a great amount
of interest in content retrieval from images and

Figure 8 Binarization and Boundary Define Horizontal



Recent studies in the field of computer vision
and pattern recognition show a great amount
of interest in content retrieval from images and
videos. This content can be in the

Figure 7: Example of a Horizontal Bar Graph

Sales (US$)

Recent studies in the field of computer vision
and pattern recognition show a great amount
of interest in content retrieval from images and

Figure 9 Text Part Detection

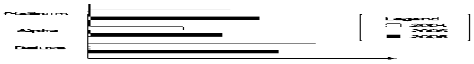


Figure 10 Image(Graph)Part Detection

4. Conclusion

- This approach is tested on 90 images(text and images parts). Out of 90 images, 84 images detected image and text part successfully. And 6 images give bad result.
- This approach is based on the number of lines of text part. If there is more number of lines then it give good result. If there is more number of images then number of text lines then results may be very poor.
- This approach works on all scripts successfully.

References

1. Jain A. and Yu. B. (1998), Automatic text location in images and video frames. *Pattern Recognition,* 31(12):2055–2076.

2. Kasturi R. and Trivedi M. (1990), *Image Analysis Applications.* New York: Marcel Dekker.

3. Liang J., Phillips I., and Haralick R. (2001), An optimization methodology for document structure extraction on Latin char- acter documents. *IEEE Trans. on Pattern Analysis and Ma- chine Intelligence*, 23(7):719–734.

4. Phillips I., Chen S., and Haralick R. (1993), CD-ROM document database standard. In *Proc. Of the 2nd Int. Conf. on Document Analysis and Recognition, Tsukuba, Japan,* pages 478–483.

5. Sobottka K., Kronenberg H., Perroud T., and Bunke H. (2000), Text extraction from colored book and journal covers. *Int. Journal on Document Analysis and Recognition*, 2(4):163–176.