TEXT EXTRACTION FROM AN IMAGE BY USING DIGITAL IMAGE PROCESSING

Praveen Choudhary
Assistant Professor, S.S. Jain Subodh P.G. College, Jaipur, India
Prateeshvip79@gmail.com

Dr. Vipin Kumar Jain
Assistant Professor, S.S. Jain Subodh P.G. College, Jaipur, India

Manuscript History
Number: IRJCS/RS/Vol.05/Issue07/JYCS10081
Received: 02, July 2018
Final Correction: 07, July 2018
Final Accepted: 12, July 2018
Published: July 2018

doi://10.26562/IRJCS.2018.JYCS10081

Editor: Dr.A.Arul L.S, Chief Editor, IRJCS, AM Publications, India
Copyright: ©2018 This is an open access article distributed under the terms of the Creative Commons Attribution License, Which Permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited

Abstract - Text in images contain important contents for information indexing and retrieval, automatic annotation and structuring of images. The text extraction from an image involves several steps including preprocessing, segmentation, feature extraction, classification, post processing. Preprocessing covers basic operation on input image like binarization which convert gray Scale image into Binary Image, noise reduction which remove the noisy signal from image. The segmentation stage for segment the given image into line by line and segment each character from segmented line. The Feature extraction defines the characteristics of character. A classification contains the database and does the comparison. It is very useful in today's word, a many time we have to recognized characters, written on sign boards with an big image, road side signals, number plates etc.

Keywords- Preprocessing; binarization; segmentation; classification;

I. INTRODUCTION

The image content is classified into two categories: perceptual content and semantic content [1]. Perceptual contents include colors, shapes, textures, intensities, and their temporal changes while semantic contents include objects, events, and their relations. Text content contains high level of semantic information as compared to visual information. Therefore text extraction from images is very significant in content analysis. It has many useful applications such as automatic bank check processing [2], vehicle license plate recognition [3], document analysis and page segmentation [4], signboard detection and translation [5], content based image indexing, assistance to visually impaired persons, text translation system for foreigners etc. Text appearing in images is classified into three categories: document text, caption text, and scene text [6]. In contrast to caption text, scene text can have any orientation and may be distorted by the perspective projection therefore it is more difficult to detect scene text.

• Document text: A document image usually contains text and few graphic components. It is acquired by scanning journal, printed document, handwritten historical document, and book cover etc.[7]

• Caption text: It is also known as overlay text or artificial text. It is artificially superimposed on the image at the time of editing, like subtitles and it usually describes the subject of the image content [7].

• Scene text: It occurs naturally as a part of the scene image and contain important semantic information such as advertisements, names of streets, institutes, shops, road signs, traffic information, board signs, nameplates, food containers, street signs, bill boards, banners, and text on vehicle etc.[7]
II. PROPERTIES OF TEXT IN IMAGES

Texts usually have different appearance due to changes in font, size, style, orientation, alignment, texture, color, contrast, and background. These changes will make the problem of automatic text extraction complicated and difficult. Text in images exhibit variations due to the difference in the following properties [8]:

- **Size**: The size of text may vary a lot.
- **Alignment**: Scene text may be aligned in any direction and have geometric distortions while caption text usually aligned horizontally and sometimes may appear as non-planar text.
- **Color**: The characters tend to have same or similar color but low contrast between text and background makes text extraction difficult.
- **Edge**: Most caption and scene texts are designed to be easily read, hence resulting in strong edges at the boundaries of text and background.
- **Compression**: Many images are recorded, transferred, and processed in compressed format. Thus, a faster text extraction system can be achieved if one can extract text without decompression.
- **Distortion**: Due to changes in camera angles, some text may carry perspective distortions that affect extraction performance.

III. TEXT EXTRACTION TECHNIQUES

The various techniques of text extraction are as follow:

3.1. Region based Method:
Region-based method uses the properties of the color or gray scale in the text region or their differences to the corresponding properties of the background. They are based on the fact that there is very little variation of color within text and this color is sufficiently distinct from text's immediate background [21]. Text can be obtained by thresholding the image at intensity level in between the text color and that of its immediate background. This method is not robust to complex background. This method is further divided into two sub-approaches: connected component (CC) and edge based.

3.1.1 CC based Method:
CC-based methods use a bottom-up approach by grouping small components into successively larger components until all regions are identified in the image [10-13]. A geometrical analysis is required to merge the text components using the spatial arrangement of those components so as to filter out non-text components and the boundaries of the text regions are marked. This method locates text quickly but fails for complex background.

3.1.2. Edge based Method:
Edges are a reliable feature of text regardless of color/intensity, layout, orientations, etc. Edge based method is focused on high contrast between the text and the background [9,14-16]. The three distinguishing characteristics of text embedded in images that can be used for detecting text are edge strength, density, and the orientation variance. Edge based text extraction algorithm is a general-purpose method, which can quickly and effectively localize and extract the text from both document and indoor/ outdoor images. This method is not robust for handling large size text.

3.2. Texture based Method:
This method uses the fact that text in images has discrete textural properties that distinguish them from the background. The techniques based on Gabor filters, Wavelet, Fast fourier transform (FFT), spatial variance, etc are used to detect the textual properties of the text region in the image [17-20]. This method is able to detect the text in the complex background. The only drawback of this method is large computational complexity in texture classification stage.

3.3. Morphological based Method:
Mathematical morphology is a topological and geometrical based method for image analysis [17,18,21]. Morphological feature extraction techniques have been efficiently applied to character recognition and document analysis. It is used to extract important text contrast features from the processed images. These features are invariant against various geometrical image changes like translation, rotation, and scaling. Even after the lightning condition or text color is changed, the feature still can be maintained. This method works robustly under different image alterations.

IV. STEPS OF METHODOLOGY

4.1 Input Image:
Image may be acquired by scanning or through the camera. This image contains text and objects. Our aim separate the text and object.

4.2 Pre-processing:
In this stage image is converted in grayscale and then into the binary format. Hence we get image which is suitable for further processing.
4.3 Segmentation:
After pre-processing the noise free image is passed to the segmentation, now image is decomposed into individual characters. Here histograms are used to detect the width of the words. Then the words are decomposed into characters.

![Text with an image](image)

Fig.: 1. Text with an image

4.4 Feature Extraction:
Here individual image glyph is considered and extracted for features. First a character glyph is defined by the following attributes like height of the character, width of the character.

```
Input Image
  ↓
Pre-Processing
  ↓
Segmentation
  ↓
Feature Extraction
  ↓
Classification
  ↓
Post Processing
```

4.5 Classification:
The features are analysed using the set of rules and labeled as belonging to different classes. The height of character and the width of the character, various distance metrics are chosen as the candidate for classification when conflict occurs. The classifications are written for other character sets. When a new glyph is given to this classifier block it extracts the features and compares the features as per the rules and then recognizes the character and labels it.

4.6 Post Processing
After the classification the whole data is checked with the database where we have already saves the classification of characters, numbers and symbols. Here our method checks the pattern of inputted character with database.
V. ALGORITHM

1. Start
2. Scan the textual image.
3. Convert color image into gray image and then binary image.
4. Do preprocessing like noise removal, skew correction etc.
5. Load the DATABASE.
6. Do segmentation by separating lines from textual image.

VI. MATLAB CODE FOR CHARACTER EXTRACTION

1. Reading the Image
   imagen=imread('image_a.jpg');
2. Showing the image on screen
   figure(1)
   imshow(imagen);
3. Convert to gray scale
   if size(imagen,3)==3 % RGB image
      imagen=rgb2gray(imagen);
   end
4. Convert to binary image
   threshold = graythresh(imagen);
   imagen =~im2bw(imagen,threshold);
5. Remove all object containing fewer than 30 pixels
   imagen = bwareaopen(imagen,30);
   pause(1)
6. Show image binary image
   figure(2)
   imshow(~imagen);
7. Label connected components
   [L Ne]=bwlabel(imagen);
8. Measure properties of image regions
   propied=regionprops(L,'BoundingBox');
   hold on
9. Plot Bounding Box
   for n=1:size(propied,1)
      rectangle('Position',propied(n).BoundingBox,'EdgeColor','g','LineWidth',2)
   end
   hold off
   pause (1)
10. Objects extraction
    figure
    for n=1:Ne
       [r,c] = find(L==n);
       n1=imagen(min(r):max(r),min(c):max(c));
       imshow(~n1);
       pause(0.5)
    end
VIII. CONCLUSION

In this paper we proposed algorithm for solving the problem of text extraction from an image. We had given the input in the form of images. The algorithm was trained on the training data that was initially present in the database. We have done preprocessing and segmentation and detect the line.

REFERENCES


