International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 3, Issue 11, November 2014

Automatic Text Detection from Images

Jignesh N Solanki[#1], Viral D Sanghvi[*2]

[#]*Computer Engineering Department, C U Shah College of Engineering and Technology Wadhwan*

Abstract— **This paper deals with problem of finding a text in image. The proposed method deal with edge-based text extraction algorithm which can automatically detect and extract text from image. It is robust with respect to the font size, style, color, orientation, and alignment of text and can be used in a large variety of application fields, such as vehicle license detection and recognition, object identification, document retrieving, page segmentation, etc. The combination result in an automatic system for text detection localization and extraction that does not require any user interface at all.**

Keywords **- Candidate Text Region Detection, Text Region Localization, Text Extraction.**

I. INTRODUCTION

Text embedded in images provides brief and important information for summarization and indexing. In recent years a lot of research works focused on detecting and localizing text in images. Many characteristics of text regions have been summarized and characterized by effective features.

Text in various forms is frequently embedded into images to provide important information about the scene like names of people, titles, locations or date of an event in news video sequences, etc. Therefore, text should be detected for semantic understanding and image indexation. In the literature, text detection, localization, and extraction are often used interchangeably. This paper is about the problem of detection and localization. Text detection refers to the determination of the presence of text in a given image and text localization is the process of determining the location of text in the image and generating bounding boxes around the text.

For text detection we need to define what text is. A text is an alignment of characters", characters being letters or symbols from a set of signs which we do not specify in advance. In images, text can be characterized by a region of elongated shape band containing a large number of small strokes. The style and the size of characters can vary greatly from one text to another. In images of written documents background as well as text color are nearly uniform, the detection of text can easily be performed by thresholding the grayscale image. However, the task of automatic text detection in natural images or video frames is more difficult due to the variety in Size, orientation, color, and background complexity. A generic system for text extraction has to cope with these problems.

II. METHODS OF TEXT DETECTION IN IMAGES

In this paper,[1] a fast and effective text detection approach is proposed. The devised features based on the stroke maps in four directions are able to better represent the intrinsic characteristic of text. The machine learning based method can construct text detector of high performance more easily. Combination of the stroke-related feature and machine learning modeling methods is a promising solution to the issue of text detection in images and videos. In this paper [2] The proposed algorithm basic used the 8-connected component to binarize the image, then to find the characters in the image and recognize them. In this paper [3] As the four connected-component-based methods for text detection have been implemented and evaluated. The most effective proves to be the sequence: Sobel edge detection, Otsu binarization, connected-component extraction and rule-based connected-component selection. A high recall rate can be achieved by collecting all the candidate text areas proposed by the four individual methods. In this paper, Inspired by [5], we utilize the stroke filter to obtain the low-level representation of image content, and then derive more compact classification feature to create a fast and effective text detector in a machine learning way. Compared with [4], our algorithm investigates to use more simple and distinctive features, so it has higher computation efficiency. Different from Liu et al. [5] which used the constraint-based way (connected component analysis) to detect text, our algorithm employs the SVM to identify text regions. In this paper[6] an efficient algorithm which can automatically detect, localize and extract horizontally aligned text in images (and digital videos) with complex backgrounds is presented. The proposed approach is based on the application of a color reduction technique, a method for edge detection, and the localization of text regions using projection pro_le analyses and geometrical properties. The output of the algorithm is text boxes with a simplified background, ready to be fed into an OCR engine for subsequent character recognition. In this Paper [7], A robust system is proposed to automatically detect and extract text in images from different sources including video, newspapers, advertisements, stock certificates, photographs and checks. Text is first detected using multiscale texture segmentation and spatial cohesion

ISSN: 2278 – 1323

constraint then cleaned up and extracted using a histogram based binarization algorithm.

III. PROPOSED SYSTEM

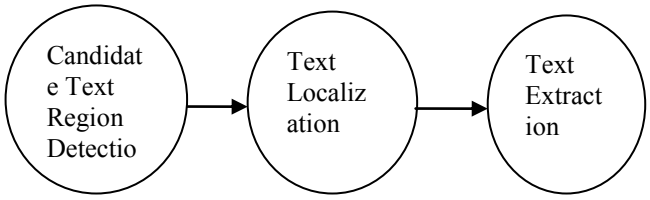


Fig. 1. Block Diagram of Proposed System

A. Candidate Text Region Detection

This stage aims to build a feature map by using three important properties of edges: edge strength, density and variance of orientations. The feature map is a gray-scale image with the same size of the input image, where the pixel intensity represents the possibility of text. In this proposed method, use magnitude of the second derivative of intensity as a measurement of edge strength as this allows better detection of intensity peaks that normally characterize text in images. The edge density is calculated based on the average edge strength within a window. Considering effectiveness and efficiency, four orientations (0^o, 45^o, 90^o, 135^o) are used to evaluate the variance of orientations, where 0^o denotes horizontal direction, 90^o denotes vertical direction, and 45^o and 135^o are the two diagonal directions, respectively.

B. Text Region Localization

Normally, text embedded in an image appears in clusters, it is arranged compactly. Thus, characteristics of clustering can be used to localize text regions. A morphological dilation operator can easily connect the very close regions together while leaving those whose position is far away to each other isolated. In this proposed method, a morphological dilation operator with a 7×7 square structuring element to the previous obtained binary image to get joint areas referred to as text blobs. Two constraints are used to filter out those blobs which do not contain text, where the first constraint is used to filter out all the very small isolated blobs whereas the second constraint filters out those blobs whose widths are much smaller than corresponding heights.

C. Text Extraction

The purpose of this stage is to extract accurate binary characters from the localized text regions for recognition.

D. Algorithm Development of Text Detection

Input: Image having text data in "*.jpg;*.png;*.bmp format

Output: detect the text part on the Image having text.

Algorithm Development:

1. *Declare the path for the Image file*
2. *Read the image and store it into variable I*
3. *Show/ Display the Image*
4. *Compare the size of the Image*
   a. *if size(I,3) == 3*
   b. *Convert the Image from RGB to gray*
   c. *Update I*
   d. *I ← rgb2gray(I)*
   e. *End*
5. *Declare the variable m for different Orientation*
   m0 = [-1 -1 -1; 2 2 2;-1 -1 -1]
   m45 = [-1 -1 2;-1 2 -1; 2 -1 -1]
   m90 = [-1 2 -1;-1 2 -1;-1 2 -1]
   m135 = [2 -1 -1; -1 2 -1;-1 -1 2]
6. *Declare the Scale value*
7. *Take the values of e for different orientations*
   a. *e = Feature_map(I,m0,m45,m90,m135)*
8. *Update I1*
   a. *I1← I*
9. *Foe all scale value calculate the Edge-map*
10. *For i= 2: Scale*
11. *Update I with the resized image*
    a. *I ←imresize(I,0.5)*
12. *Update e1*
    a. *e1 = Feature_map(I,m0,m45,m90,m135*
13. *Resize the matrix to add up*
14. *e2 ← imresize(e1,size(e))*
15. *Update e*
    a. *e ← e + e2*
16. *Convert the image to Binary*
    a. *edge_map = e>60*scale*

ISSN: 2278 – 1323

17. *Show the image*
    a. *imshow(e,[])*
18. *Apply Multy edge for clearing the text*
19. *Update Text with Multi edge*
20. *Text ← multiedge(I1,2)*
21. *Show image*
22. *Delete the text*
    a. *bw = bwmorph(edge_map,'dilate')*
23. *Remove smaller areas*
    a. *bw = bwareaopen(bw,round(0.04*numel(I)))*
24. *Show image with bw*
25. *Take the segmented part of the image or the one by one area*
    a. *[L count] = bwlabel(bw)*
26. *Generate the empty Matrix at the Text Zone*
27. *TEXT_region = zeros(size(bw,1),size(bw,2))*
28. *For every image*
29. *Update bw1*
    a. *bw1 = L==ii*
30. *Update A*
    a. *A = sum(bw1(:))*
31. *Generate the Matrix*
    a. *[xi yi] = find(bw1)*
32. *H = max(xi)-min(xi)*
33. *W = max(yi)-min(yi)*
34. *Calculate A*
    a. *A/(H*W)*
35. *if A/(H*W) >0.4*
36. *Update Text region*
37. *TEXT_region(min(xi):max(xi),min(yi):max(yi)) ←Text(min(xi):max(xi),min(yi):max(yi))*
38. *End*
39. *Show figure with text image.*

E. Algorithm Development of Feature map

Input: u, m0, m45, m90, m135 u : Gray Image
m0, m45, m90, m135: Convolution Operator

Output: mapping the feature of the image with many orientations.

Algorithm Development:

1. *Perform Convolution operation of the mask with the complete Image data*
2. *Initialize variables for storing the convoluted output for different orientations*
    a. *a1t ← abs(conv2(double(u),m0, 'same'))*
    b. *a2t ←abs(conv2(double(u),m45, 'same'))*
    c. *a3t ←abs(conv2(double(u),m90, 'same'))*
    d. *a4t ←abs(conv2(double(u),m135, 'same'))*
3. *Remove extra lines created by Convolution process.*
4. *Initialize variables for storing the convoluted output with different orientations.*
5. *Update the variables*
    a. *a1←a1t*
    b. *a2←a2t*
    c. *a3←a3t*
    d. *a4←a4t*
6. *Fuze both Horizontal and vertical edges and threshold values*
    a. *e ← a1 + a2 + a3 + a4*
7. *Assign the Normalized Value*
    a. *N = 4*
8. *Update the value of e*
    a. *e ← e./N*

F. Algorithm Development of Multiedge

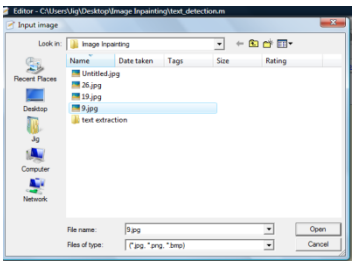
Algorithm Development:

1. *Declare the variable h1, h2, v, a1, a2, ns*
2. *Assign the values*
    a. *h1= [1 0 -1; 2 0 -2; 1 0 -1]*
    b. *h2= [1 2 1; 0 0 0;-1 -2 -1]*
    c. *v= [1 2 1; 2 4 2; 1 2 1]*
    d. *a1=filter2(h1,a)*
    e. *a2=filter2(h2,a)*
    f. *ns =h1*
3. *foreach Image*
    a. *ii=1:s*
4. *nsh←conv2(ns,v)*

ISSN: 2278 – 1323

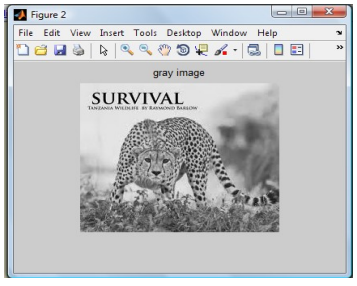
5. *calculate the value of p*
6. *p←max(max(ns))/max(max(nsh))*
7. *Increase the scale of the wavelet*
8. *Calculate ah1 and ah2*
9. *ah1=filter2(v,a1)*p;*
10. *ah2=filter2(v,a2)*p;*
11. *Update ns with nsh*
12. *ns← nsh*
13. *Distinguish Noise and the original image*
    a. *Update a1 and a2*
    b. *a1←a1.*(abs(a1)<=abs(ah1))+ah1.*(abs(a1)>abs(ah1))*
14. *a2←a2.*(abs (a2)<=abs(ah2))+ah2.*(abs(a2)>abs(ah2))*
    a. *end*
15. *Update e*
    a. *E←(a1.^2+a2.^2)*
16. *Update the Threshold*
    a. *Y←(e>filter2(ones(10)/100,e)).*(e>mean2(e))*

IV. Result Analysis

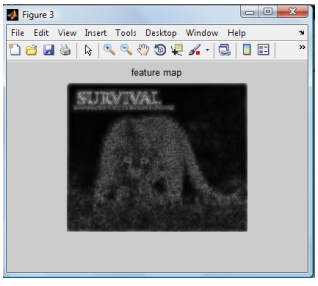


(a) Upload Image

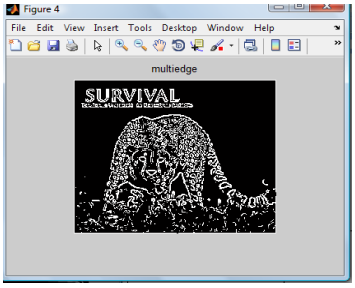
(b) Color Image



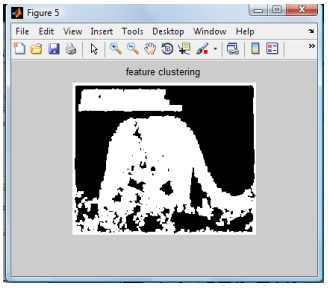
(c) Gray Image



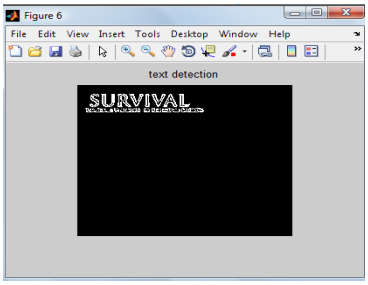
(d) Feature Map



(e) Multi Edge Detection



(f) Feature Clustering



(g) Text Extraction

Fig.2 Text Detection from Images.

V. Conclusions

The proposed method is not sensitive to image color/intensity and Robust with respect to aspect to font sizes and styles, orientations, alignment/layout.

References

[1] *Xiaojun Li*1, *Weiqiang Wang, Shuqiang Jiang,Qingming Huang*1,2, *Wen Gao*, fast and effective text detection
[2] *Mohanada Alata mohamad Al-Shabi "Text Detection and character Recognition using fuzzy image processing "*
[3] Nobuo Ezaki, Marius Bulacu Text Detection from Natural Scene Images: Towards a System for Visually Impaired Person.
[4] Q.Ye, Q.Huang, W.Gao and D. Zhao, "Fast and robust text detection in images and video frames," *Image Vis. Comput.* vol. 23, no. 6, pp. 565-576, Jun. 2005.
[5] Q. Liu, C. Jung, S. Kim, Y. Moon and J.Kim," Stroke filter for text localization in video images," in *Proc. Int. Conf. Image Process.,* Atalanta, GA, USA, Oct. 2006, pp. 1473-1476.
[6] Julinda Gllavata1, Ralph Ewerth1 and Bernd Freisleben A Robust Algorithm for Text Detection in Images
[7] Victor Wu, Raghavan Manmatha and Edward M Riseman TextFinder: An automatic system to detect and recognize text in Images.

ISSN: 2278 – 1323