Text Detection and Recognition Using Camera Based Images

H.Y. Darshan[1], M.T. Gopalkrishna[1], and M.C. Hanumantharaju[2]

[1] Department of Information Science and Engg,
Dayananda Sagar College of Engg, Bangalore, India
[2] Department of Electronics and Communication Engg,
Sri Bhusanayana Mukundadas Sreenivasaiah Institute of Technology, Bangalore, India
{repsol26.DHY,gopalmtm,mchanumantharaju}@gmail.com

Abstract. The increase in availability of high performance, low-priced, portable digital imaging devices has created an opportunity for supplementing traditional scanning for document image acquisition. Cameras attached to cellular phones, wearable computers, and standalone image or video devices are highly mobile and easy to use; they can capture images making them much more versatile than desktop scanners. Should gain solutions to the analysis of documents captured with such devices become available, there will clearly be a demand in many domains. Images captured from images can suffer from low resolution, perspective distortion, and blur, as well as a complex layout and interaction of the content and background.In this paper, we propose an efficient text detection method based on Maximally Stable Exterme Region (MSER) detector, saying that how to detect regions containing text in an image. It is a common task performed on unstructured scenes, for example when capturing video from a moving vehicle for the purpose of alerting a driver about a road sign . Segmenting out the text from a clutterd scene greatly helps with additional tasks such as optical charater recognition (OCR). The efficiency of any service or product, especially those related to medical field depends upon its applicability. The applicability for any service or products can b achieved by applying thr basic principles of Software Engineering.

Keywords: Text detection, maximally stable extremal re-gions, connected component analysis.

1 Introduction

With the increase in growth of camera-based applications available on smart phones and portable devices, understanding the pictures taken by these devices semantically has gained increasing scope from the computer vision community in these years. Among all the information contained in the image, text, which carries semantic information, could provide valuable clues about the content of the image and thus is very important for human as well as computer to understand the scenes. For character recognition in the scene, these methods

S.C. Satapathy et al. (eds.), *Proc. of the 3rd Int. Conf. on Front. of Intell. Comput. (FICTA) 2014 – Vol. 2,* Advances in Intelligent Systems and Computing 328, DOI: 10.1007/978-3-319-12012-6_63

directly extract features from the original image and uses various classifiers to the character to recognize. While for scene text recognition, since there are no Binarization and Segmentation stages, most methods that exist take up multi-scale sliding window strategy to get the character detection results. As sliding window strategy does not make use of the special structure information of each character, it will produce many false positives. Thus, these methods mainly depends on the post processing methods such as pictorial structures. As proved by Judd et al., given an image containing text and other objects, viewers tend to fixate on text, suggesting the importance of text to human.

In fact, text recognition is indispensable for a lot of applications such as automatic sign reading, language translation, navigation and so on. Thus, understanding scene text is more important than ever. The following sources of variability still need to be accounted for: (a) font style and thickness; (b) background as well as foreground color and texture; (c) camera position which can introduce geometric distortions; (d) illumination and (e) image resolution. All these factors combine to give the problem a flavor of object recognition. Many problems need to be solved in order to read text in camera based natural images including text localization, character and word segmentation, recognition, integration of language models and context, etc.

2 Review of Literature

Devvrat C. Nigam et al. [2] proposed character extraction and edge detection algorithm for training the neural network to classify and recognize the handwritten characters. In general, handwriting recognition is classified into two types as off-line and on-line handwriting recognition methods. The on-line Methods have been shown to be superior to their off-line counterparts in recognizing handwritten characters due to the temporal information available with the former. There are basically two main phases in our Paper: Pre-processing and Character Recognition. In the first phase, they are preprocessing the given scanned document for separating the Characters from it and normalizing each characters. Initially we specify an input image file, which is opened for reading and preprocessing. The image would be in RGB format (usually), so we convert it into binary format. To do this, it converts the input image to grayscale format (if it is not already an intensity image), and then uses threshold to convert this grayscale image to binary i.e. all the pixels above a certain threshold as 1 and below it as 0.

Mohanad Alata et al. [6] proposed method was based on a combination of an Adaptive Color Reduction (ACR) technique and a Page Layout Analysis (PLA) approach. K. Atul Negi, Nikhil Shanker and Chandra Kanth Chereddi [6] presented a system to locate, extract and recognize Telugu text. The circular nature of Telugu script was exploited for segmenting text regions using the Hough Transform.

Jerod J. Weinman et al. [5] propose a probabilistic graphical model for STR that brings both bottom-up and top-down information as well local and long-distance relationships into a single elegant framework. In addition to individual character appearance, our model integrates appearance similarity, one underused source of information, with local language statistics and a lexicon in a unified probabilistic framework to reduce false matches errors in which the different characters are given the same label by a factor of four and improve overall accuracy by greatly reducing word error. The model adapts to the data present in a small sample of text, as typically encountered when reading signs, while also using higher level knowledge to increase robustness.

Shangxuan Tian et al. [3] propose to recognize the scene text by using an extension of the HOG, namely, co-occurrence HOG (Co- HOG) [13] that captures gradient orientation of neighboring pixel pairs instead of a single image pixel. Co-HOG divides the image into blocks with no overlap which is more efficient than HOG with overlapped blocks [25]. This is essential in the real-time text recognition system. More Importantly, relative location and orientation are considered with each neighboring pixel, respectively, which is more precise to describe the character shape. In addition, Co-HOG Keeps the advantages of HOG, i.e., the robustness to varying illumination and local geometric transformations.

Jain et al. [14] perform a color space reduction followed by color segmentation and spatial regrouping to detect text. Although processing of touching characters is considered by the authors, the segmentation phase presents major problems in the case of low quality documents, especially video sequences.

3 Proposed Method

The proposed text detection and recognition algorithm consists of the following steps: MSER region detection, Edge detection, Filter character candidates, Determine bounding boxes, Perform optical character recognition(OCR)

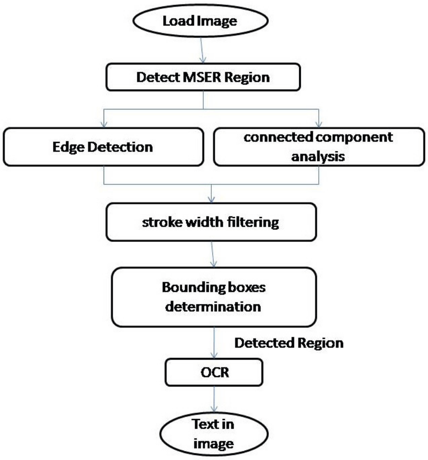


Fig. 1. Flow chart of the proposed method

3.1 Detect MSER Regions

Maximally Stable External Regions(MSER) are used as a method of blob detection in images. This method of extracting a comprehensive number of corressponding image elements contributes to the wid-baseline matching, and it has led to better stereo matching and object recognition algorithm. Becausee the regions are defined exclusively by the intensity functions in the region and the outer border, this leads to many key characteristics of the region which make them useful, and as follows: Invariance to affine transformation of image intensities, Covariance to adjacency preserving (continuous)transformation T : D –> D on the image domain, Stability, Multi-scale detection without any smoothing involved, The set of all external regions can be enumerated in worst-case O(n), where n is the number of pixels in the image. The algorithm -Start from a local intensity extremum point, -Go in every direction until the point of extremum of some function f. The curve connecting the points is the region boundary, -Compute geometric moments of orders up to 2 for this region, -Replace the region with ellipse.

3.2 Edge Detection

The Canny operator was designed to be an optimal edge detector (according to particular criteria — there are other detectors around that also claim to be optimal with respect to slightly different criteria). It takes as input a gray scale image, and produces as output an image showing the positions of tracked intensity discontinuities. The effect of the Canny operator is determined by three parameters — the width of the Gaussian kernel used in the smoothing phase, and the upper and lower thresholds used by the tracker. Increasing the width of the Gaussian kernel reduces the detector's sensitivity to noise, at the expense of losing some of the finer detail in the image. The localization error in the detected edges also increases slightly as the Gaussian width is increased. Usually, the upper tracking threshold can be set quite high, and the lower threshold quite low for good results. Setting the lower threshold too high will cause noisy edges to break up. Setting the upper threshold too low increases the number of spurious and undesirable edge fragments appearing in the output. One problem with the basic Canny operator is to do with Y-junctions i.e. places where three ridges meet in the gradient magnitude image. Such junctions can occur where an edge is partially occluded by another object. The tracker will treat two of the ridges as a single line segment, and the third one as a line that approaches, but doesn't quite connect to, that line segment.

3.3 Connected Component Analysis

First, we were instructed to write a function that would construct a histogram for a grayscale image. In order to do this, we must first be able to read in image data. In MATLAB, the imread() function will return a 3d array that looks like (rows,columns,color channels). Once we have our image read and stored in a 2d

Table 1. Results of Text detection

Algorithm	Precision	Recall
Ashida (ICDAR 2003) [9]	0.55	0.46
Shivakumara (TPAMI 2011) [1]	0.71	0.73
Our method	0.86	0.82

array, we can construct a histogram of relative frequencies. The histogram is an array with a user specified number of bins. The user also specifies the min and max value to use in creating the histogram. For example, if min and max were 100 and 200 respectively, every value lower than 100 would be placed in the smallest bin and every value above 200 would be placed in the largest bin. Finally, each bin is normalized by dividing by the number of pixels in the image. This makes the area under the histogram 1.0, and gives us the probability a pixel has a certain intensity if sampled randomly from the image.Finally we filter out the character candidates using the same.

3.4 Stroke Width Filtering

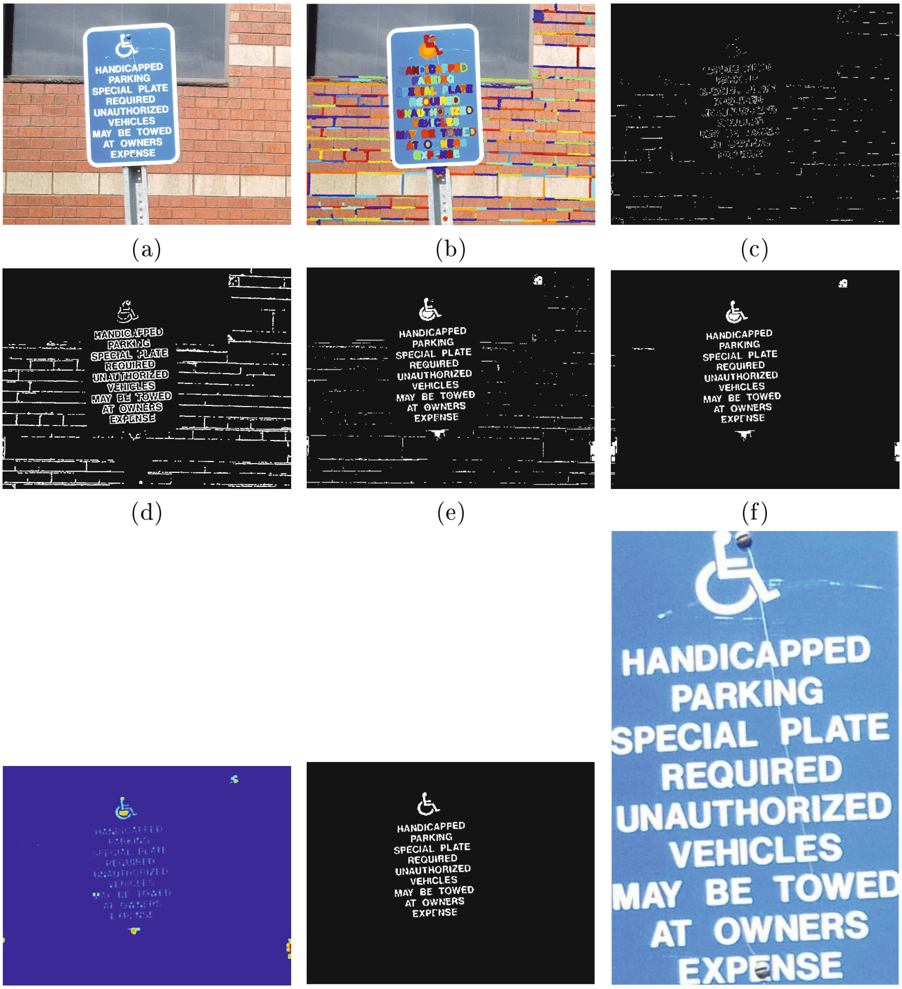
Another useful discriminator for text in images is the variation in stroke width within each text candidate. Characters in most languages have a similar stroke width or thickness throughout. It is therefore useful to remove regions where the stroke width exhibits too much variation [1]. The stroke width image below is computed using the helperStrokeWidth helper function.

3.5 Determining Bounding Boxes

To compute a bounding box of the text region, we will first merge the individual characters into a single connected component. This can be accomplished using morphological closing followed by opening to clean up any outliers.

4 Experimental Results

In this study, the images for experiments are from ICDAR database [13]. Even we have experimented on NEOCR (Natural Environment Optical Character Recognition), MSER datasets . The precision rate and recall rates are to be as shown in the above table (tab.1). The experimental results for image is as shown below where (a) is the original image taken and followed by the stages to extract text region in image as in (i). Compared with the best algorithm we have listed, the proposed method improved the precision rate by 15 percent and improved the recall rate by 9 percent.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Fig. 2. Results of text detection. (a) Original image, (b) MSER Regions, (c) Canny edge detector, (d) Grown edges, (e) Segmented MSER regions, (f) Region filtered image, (g) Visualization of text candidates using stroke width, (h) Text candidates after stroke width filtering, (i) Text region

5 Conclusion

In this paper, we present a new text detection approach.Our approach uses both MSER and edges information.We have presented a method for locating text within natural images. The algorithm relies on a fundamental feature of text: text is usually surrounded by a contrasting, uniform background. Our proposed method of text segmentation searches for the textâĂŹs background rather than the actual text. This allows for a large variation in the distribution of text features while requiring little computation. We propose a MSER region detector to find the common characteristics of the text in an image. Experimental results on our own database as well as ICDAR 2003 text locating dataset demonstrate that our approach is robust to the orientation, perspective, color, and lighting of the text object, and can detect most text objects successfully and efficiently.

References

1. Wang, K., Babenko, B., Belongie, S.: End-to-end scene text recognition. In: International Conference on Computer Vision, ICCV (2011)
2. Shahab, A., Shafait, F., Dengel: Reading text in scene images. In: International Conference on Document Analysis and Recognition (2011)
3. Felzenszwalb, P., Girshick, R., McAllester, D., Ramanan, D.: Object detection with discriminatively trained partbased models. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(3), 402–413 (2011)
4. de Campos, T., Babu, B., Varma, M.: Character recognition in natural images. In: VISAP (2009)
5. Boureau, Y., Bach, F., LeCun, Y., Ponce, J.: Learning mid-level features for recognition. In: Computer Vision and Pattern Recognition, vol. 2013, Article Id 716948, 8 pages. Hindwani Publication Corporation
6. Weinman, J.J.: Typographical features for scene text recognition. In: IAPR International Conference on Pattern Recognition (August 2010)
7. Sharma, N., Pal, U., Kimura, F.: Recognition of Handwritten Kannada Numerals. In: 9th International Conference on Information Technology (2010)
8. Mishra, A., Alahari, K., Jawahar, C.V.: Top-down and bottom-up cues for scene text recognition. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2012)
9. Newell, A., Griffin, L.: Multiscale histogram of oriented gradient descriptors for robust character recognition. In: International Conference on Document Analysis and Recognition, ICDAR (2011)
10. Mishra, A., Alahari, K., Jawahar, C.V.: Top-down and bottom-up cues for scene text recognition. In: CVPR (2012)
11. Wang, K., Babenko, B., Belongie, S.: End-to-end scene text recognition. In: International Conference on Computer Vision, ICCV (2011)
12. Watanabe, T., Ito, S., Yokoi, K.: Co-occurrence histograms of oriented gradients for human detection. Information and Media Technologies (2010)
13. Sin, B., Kim, S., Cho, B.: Locating characters in scene images using frequency features. In: Proc. IEEE Int. Conf. Pattern Recognition (2010)
14. Shivakumara, W.H.P., Tan, C.: An efficient edge based technique for text detection in video frames. In: Proc. 8th IAPR Workshop Document Analysis Systems (September 2008)