

RESEARCH PAPER

Available Online at www.jgrcs.info

DETECTION OF TEXT STRING FROM NATURAL SCENE IMAGES BY IMAGE PARTITION AND GROUPING

Prajakta Bastawade*1, Prof. Bharati Dixit 2

*¹ Department of Information Technology, MIT College of Engineering, Pune, Maharashtra, India
prajaktamagdum@gmail.com

2 Department of Information Technology, MIT College of Engineering, Pune, Maharashtra, India
dixit.bharati@gmail.com

Abstract: Text detection in natural scene images is important task for applications such as assistive navigation, auxiliary reading, image retrieval, scene understanding, etc. This paper explores a new framework to detect text strings in complex natural scene images which consists of two steps: A) image partition to find text character candidates based on local gradient features and color uniformity of character components and 2) character candidate grouping to detect text strings based on joint structural features of text characters. In Boundary Clustering a new bigram-color-uniformity-based method is developed to model both text and attachment surface, and cluster edge pixels based on color pairs and spatial positions into boundary layers which is given as input for the entire framework. The given framework outperforms the state-of-the-art results on the public RRD which contains text only in horizontal orientation, and more effective on MSRA-TD500 which contains text in arbitrary orientations.

Key words: Adjacent character grouping, Gradient Magnitude, Text string detection, Text line grouping, Hough transform

I. INTRODUCTION

Text detection in natural scenes becomes a critical yet challenging task. Text information provides brief and significant clues for many image based applications such as scene understanding, content-based image retrieval, assistive navigation and automatic geocoding.

Different from document images, in which text characters are arranged into elegant poses and proper resolutions, text in natural scene images embedded in arbitrary shapes, sizes, and orientations into complex background, as shown in Fig. 1.[5]



Fig. 1. Examples of text in natural scene images [5]

To extract text string information from camera-captured images many algorithms and commercial optical character recognition (OCR) systems have been developed [1], [6]. All of these algorithms share the same assumption that

locations of text characters are approximately predictable, and background interference does not resemble text characters. It is impossible to recognize text in natural scene images directly because the off-the-shelf OCR software cannot handle complex background interferences and nonorienting text lines. Thus, we need to detect image regions containing text strings and their corresponding orientations. Text detection and localization procedure described in the survey of text extraction algorithms [2]

The rest of the paper is organized as follows. Section II gives text detection approaches, Section III describes overview of framework. Section IV describes boundary clustering. Section V describes algorithms of image partition to extract text. Section VI describes two grouping methods to extract text strings. Experimentation are presented in Section VII. Conclusion of the paper is given in Section VII and Section VIII gives Future work.

II. TEXT DETECTION APPROACHES[6]

1. **Connected Component Based-**It groups neighbouring pixels of similar colors into connected components and grouping small components into successively larger components until all regions are identified in the image.

Drawback-

-Not robust because they are based on geometrical properties of components

2. **Edge-based methods** -The edges of the text boundary are identified and merged, and then several heuristics are used to filter out the non-text regions.

Drawback-

-It give more false positives when the complex background present

3. **Texture-based methods** -It use the observation that text in images have distinct textural properties that distinguish them from the background. can be used to detect of a text region in an image.

Drawback-

-May be unsuitable for small fonts and poor contrast text

The algorithms in this proposed framework belong to this category of partition and grouping in which image is partition into to blocks and then groups the blocks verified by the features of text characters. In we make an effort to build an effective and practical detection system for texts of arbitrary orientations in complex natural scenes

III. OVERVIEW OF OUR FRAMEWORK

The proposed framework consists of Three main steps, given here.

- Step 1) Boundary Clustering [4]
- Step 2) Image partition to find text character candidates [1]
- Step 3) Character candidate grouping to detect text strings based on joint structural features of text [1]

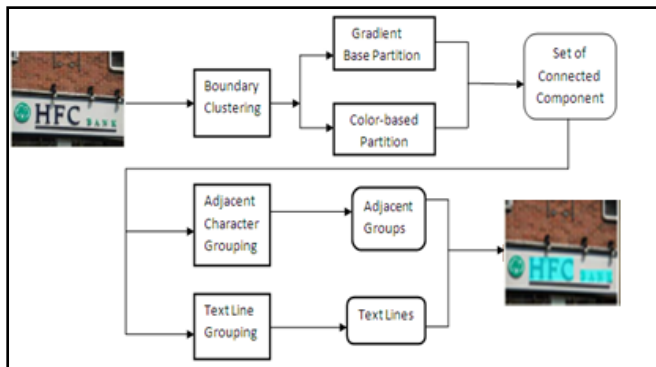


Fig .2 Flowchart of the framework of text string detection [1]

IV. BOUNDARY CLUSTERING

Boundary plays an important in role in the structural analysis and the geometrical model of text. In scene images, object boundary is derived from the color difference of two uniform regions: object and its surrounding backgrounds. Thus color uniformity and spatial

positions are employed to analyze the boundaries of text characters, and.

This paper uses a clustering algorithm to separate them from the boundaries of background outliers. . Fig. 3 illustrates three examples of boundary layers after EM-based clustering, in which, edge pixels with similar color pairs and spatial positions are grouped into the same layer. Boundaries at different veridical positions are assigned into different boundary layers because of y-coordinate spatial information in the clustering process

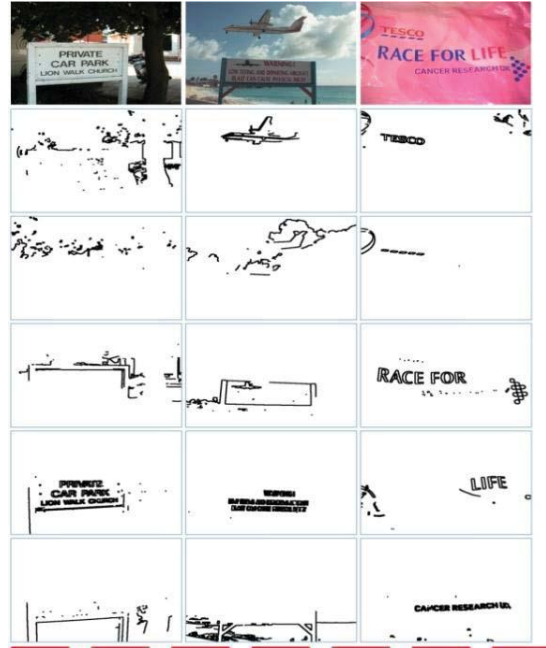


Fig. 3. Examples of boundary layers from scene images[4]

V. IMAGE PARTITION

To extract text information from a complex background, image partition is first performed to group together pixels that belong to the same text character, to get a binary map of candidate character components. Two algorithms have been designed gradient-based partition algorithm and color-based partition algorithm, respectively

A. Gradient-based partition by connecting paths of pixel couples

Although text characters and strings vary in font, size, color, and orientation, they are composed of strokes which are rectangle connected components with closed-width boundaries and uniform torso intensities as shown in Fig 4 [3]

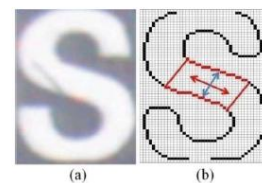


Fig 4 (a) Image patch of a text character (b) Stroke is marked by a red character boundary. [3]

On the gradient map we take an edge pixel from edge map as starting point and probe its partner along a path in gradient direction. If another edge pixel is reached with approximately equal gradient magnitudes and opposite directions .we obtain a pixel couple and its connecting path from to . This algorithm is applied to calculate connecting paths of all pixel couples shown in Fig. 5(b) marks all of the connecting paths shorter than 30 as white foreground.

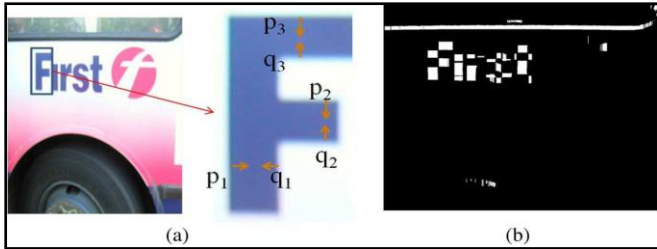


Fig 5 . a) Examples of pixel couples b) Connecting paths of all pixel couples [5]

The partition process is divided into two rounds. In the first round, the length range of connecting path is set as $0 < l < 30$ to describe stroke width. For each pixel couple whose connecting path falls on this length range, we establish an exponential distribution of gradient magnitudes of the pixels on its connecting path, and the λ decay rate is obtained

Thus, the connecting path with greater decay rate is marked as white foreground representing candidate character component, as shown in Fig. 6[1]

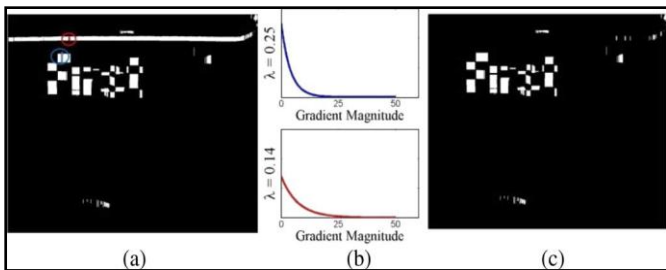


Fig 6 Two connecting paths of pixel couples and exponential distribution of gradient magnitudes [1]

To extract the complete stroke in rectangle shape, we start the second round use the concept of the aspect ratio of the rectangle stroke is no more than 6:1. At last, we perform morphological close and open as post processing to refine the extracted connected components as shown in Fig. 7[1]



Fig.7. The bottom row presents the partitioned components obtained from the first round and the second round, respectively[1]

B. Color-based partition by color reduction

we can locate text information by extracting pixels with similar colors.. Inspired by [4], we perform color reduction by using color histogram and weighted K-means clustering through the following steps.

First, a canny edge detector is performed to obtain edge image. Second, calculate color histograms of the original input image.. Third, after mapping all of the pixels from spatial domain to RGB color space, as shown in Fig. 8(b), weighted K-means clustering is performed to group together the pixels with similar colors.

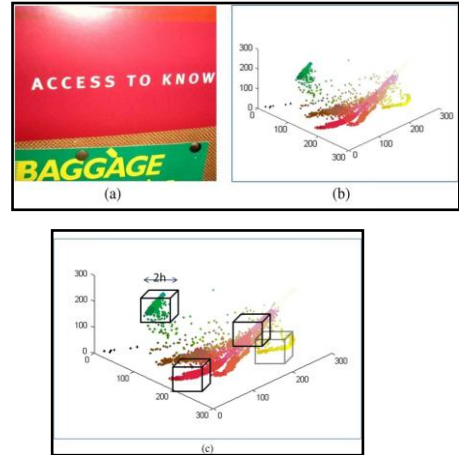


Fig. 8. (a) Scene image with multiple colors. (b) Color distribution in RGB space. (c) Four of the initial cube color clusters with radius h [1]



Fig.9. Some examples of color-based partition, where the left column shows original images and other columns contain the corresponding dominant color layers[1]

Each input image is partitioned to several color layers. A color layer that consists of only one foreground color on white background is a binary map of candidate character components. Then, connected component analysis is performed to label foreground regions of connected pixels.[6]

VI .CONNECTED COMPONENTS GROUPING

The image partition creates a set of connected components from an input image, including both text characters and unwanted noises. Observing that text information appears as one or more text strings in most natural scene images, we perform heuristic grouping and structural analysis of text strings to distinguish connected components representing text characters from those representing noises. Assuming that a text string has at least three characters in alignment, we develop two methods to locate regions containing text strings: adjacent character grouping and text line grouping, respectively.

A. Adjacent Character Grouping

Text strings in natural scene images usually appear in alignment, namely, each text character in a text string must possess character siblings at adjacent positions. Here, five constraints are defined to decide whether two connected components are siblings of each other.

- 1) Considering the capital and lowercase characters, the height ratio falls between $1/T1$ and $T1$
- 2) The distance between two connected components should not be greater than $T2$ times the width of the wider one.
- 3) For text strings aligned approximately horizontally, the difference between y -coordinates of the connected component centroids should not be greater than $T3$ times the height of the higher one.
- 4) Two adjacent characters usually appear in the same font size, thus their area ratio should be greater than $1/T4$ and less than $T4$.
- 5) If the connected components are obtained from gradient based partition as described in Section III-A, the color difference between them should be lower than a predefined threshold $T5$ because the characters in the same string have similar colors

According to the five constraints, a left/right sibling set FL/FR is defined for each connected component C as the set of sibling components located on the left/right of C

For two connected components C and C' , they can be grouped together as sibling components if the above five constraints are satisfied.



Fig. 10. (a) Sibling group of the connected component “r” where “B” comes from the left sibling set and “o” comes from the right sibling set. (b) Merge the sibling groups into an adjacent character group corresponding to the text string “Brolly?” (c) Two detected adjacent character groups marked in red and green, respectively.[5]

. As shown in Fig. 10[1] the resulting union of connected components is defined as adjacent character group denoted by AG , which is a subset of the set of connected components ..

B. Text Line Grouping

In order to locate text strings with arbitrary orientations, we develop text line grouping method. To

group together the connected components which correspond to text characters in the same string which is probably nonhorizontal, we use centroid as the descriptor of each connected component. Given a set of connected component centroids, groups of collinear character centroids are computed, as shown below

$$M = \{m | C \in S \text{ and } m = \text{centroid}(C)\}$$

$$L = \{G | G \subseteq M, |G| \geq 3, \forall m_i, m_j, m_k \in G,$$

where M denotes the set of centroids of all of the connected components obtained from image partition, and L denotes the set of text lines which are composed of text character centroids

They are character centroids and they are collinear where M denotes the set of centroids of all of the connected components obtained from image partition, and L denotes the set of text lines which are composed of text character centroids in alignment. Hough transform to describe the fitted line l_u by $\langle r_u, \theta_u \rangle$, resulting in ρ , where ρ is the equation of the fitted line in the Hough space. Thus, other collinear centroids along can be added into the end positions to form a complete text string increasingly. Fig. 11[1] illustrates the processing of fitted line refinement

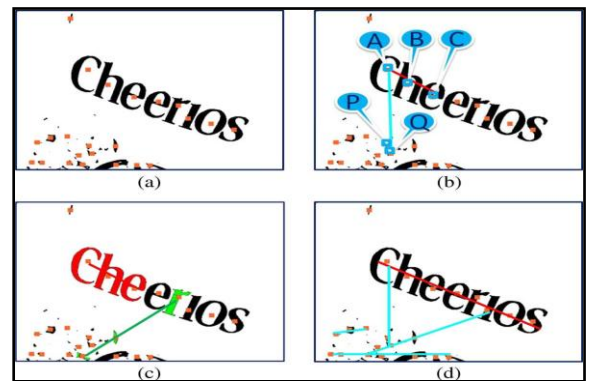


Fig. 11. Resulting Fitted Lines From Centroids Cascading. Red Line Corresponds To Text Region While Cyan Lines Are False Positives to Be Removed [1]

VII. EXPERIMENTATION

a) Datasets

Two datasets are employed to evaluate the proposed algorithms. The first is the Robust Reading Dataset RRD from ICDAR 2003[7],[10] which contain only horizontal text strings. The second dataset is MSRA Text Detection 500 Database (MSRA-TD500) [11] is collected and released publicly as a benchmark to evaluate text detection algorithms. The MSRA Text Detection 500 Database (MSRA-TD500) contains 500 natural images, which are taken from indoor (office and mall) and outdoor (street) scenes using a pocket camera .

b) Implementations

The experimental results are presented in following figures-

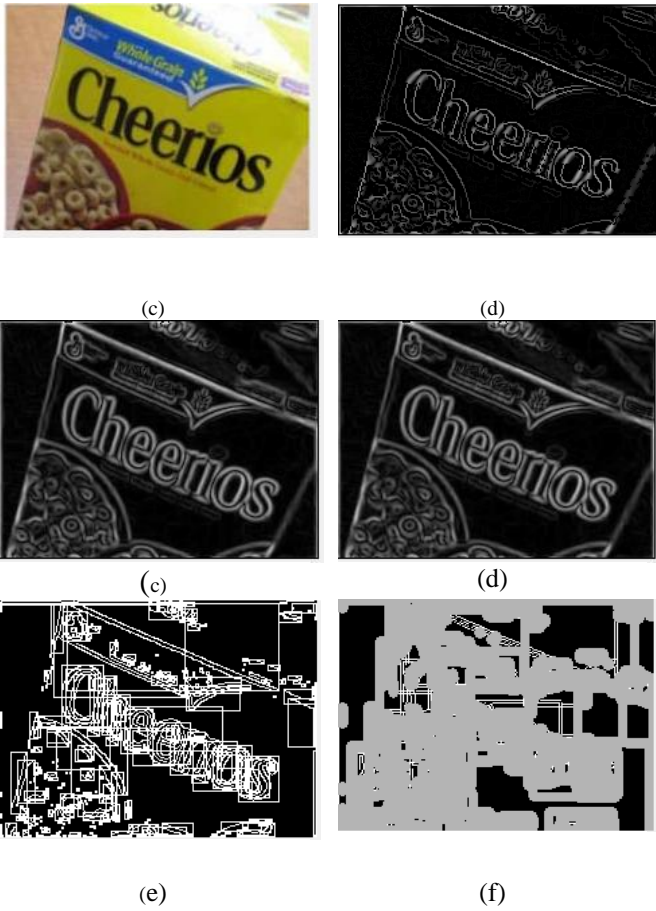


Table 2. Comparison of Precision, Recall ,FMeasure for JPG,GIF and BMP Images with and without Boundary Clustering(BC)

Image	Precision	Recall	Fmeasure	Time(s)
JPG_BC	0.76	0.9	0.83	8.82
JPG	0.71	0.77	0.74	8.93

JPG_BC: JPG with Boundary Clustering

JPG : JPG without Boundary Clustering

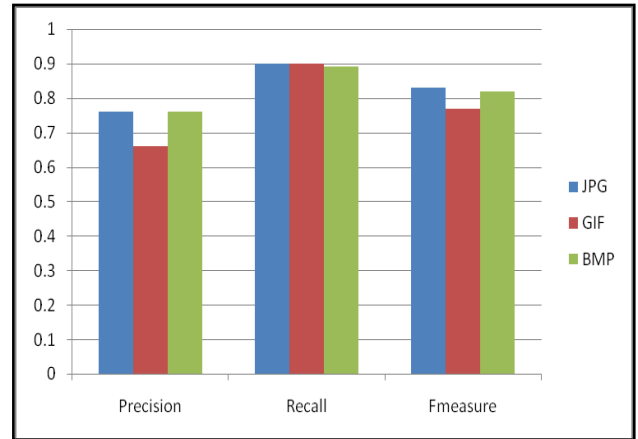


Fig 13. Variations in Precision, Recall and f-measure for different types of images.

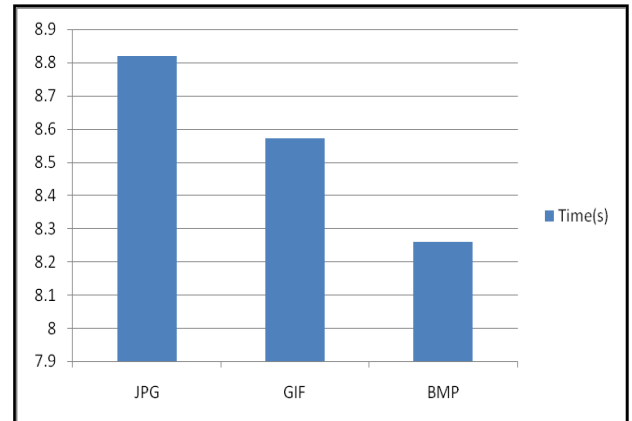


Fig 14. Time Analysis of Various Types of Images

Also the performances for JPG,BMP and GIF types of Images with and without boundary clustering are evaluated Results are presented in Table 6.

Fig 12 .(a)Original Image (b)Result of Boundary Clustering (c) Result of radient-based Partition Algorithm (d) Result of Color BasedPartition (e) Result of Adjacent Character Grouping (f) Result of Text Line Grouping

c) Results

To evaluate the performance, we calculate two metrics, precision and recall as in [7], [8]. Here, precision is the ratio of area of the successfully extracted text regions to area of the whole detected region, and recall is the ratio of area of the successfully extracted text regions to area of the ground truth regions. The area of a region is the number of pixels inside it.

The framework will better on JPG images .To show this some BMP and some JPG images are taken to evaluate the performance of Precision ,Recall ,f-measure and Time metrics. Results are presented in Table 5.

Table 1. Comparison of Precision, Recall ,FMeasure forJPG,GIF and BMP Images

Image	Precision	Recall	Fmeasure	Time(s)
JPG	0.76	0.9	0.83	8.82
GIF	0.66	0.9	0.77	8.57
BMP	0.76	0.89	0.82	8.26

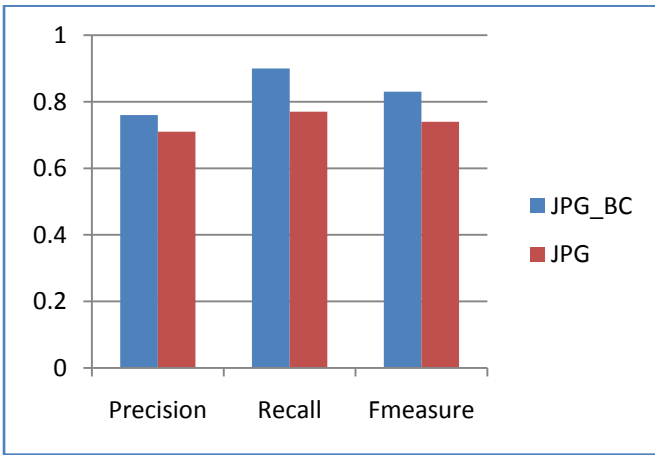


Fig 15. Variation in Precision ,Recall ,FMeasure with and without Boundary Clustering for JPG Images.

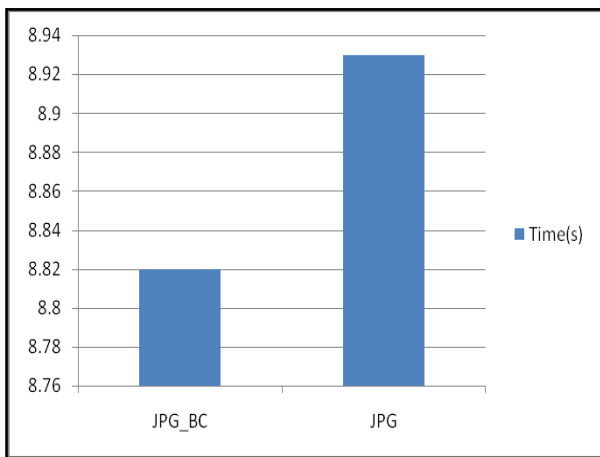


Fig 16. Time Analysis with and without Boundary Clustering for JPG Images

e) Result Analysis

The experimental results on different types of images in which some are taken from ICDAR 2003 and some are taken from MSRA Text Detection 500 Database (MSRA-TD500) are illustrated in Table 1.

To evaluate the performance of framework various types of images like BMP,GIF,and JPG are taken .As shown in Fig 13 Precision ,Recall and f-measure are comparable for BMP and JPG images. If we see Fig 14. Time taken to extract the text data for JPG images is more than GIF and BMP images.

Also the experimental results on the JPG images which are taken from same dataset are with and without boundary clustering are shown in Table2 .Also as shown in Fig 15 value of Precision, Recall and f-measure are better for JPG with boundary clustering algorithm than without boundary clustering algorithm .If we see fig 16. Time taken extract text data from JPG images with clustering algorithm is relatively less than Time taken to extract text data from JPG images without clustering algorithm.

VIII .CONCLUSION

Due to the unpredictable text appearances and complex backgrounds, text detection in natural scene images is still challenging. To locate text regions embedded in those images, a new framework is proposed based on boundary clustering, image partition and connected components grouping. Structural analysis is performed from text characters to text strings. Boundary clustering is performed to cluster edge pixels based on color pairs and spatial positions into boundary layers. Then the candidate text characters from connected components are chosen by gradient feature and color feature. And finally character grouping is performed to combine the candidate text characters into text .The text line grouping is able to extract text strings with arbitrary orientations

VIII. FUTURE WORK

The framework of this paper will not work on blur images and also on non English language text data. Future work will focus on developing the method for text extraction from complex backgrounds for blur images by performing some preprocessing operations and also developing the method for text extraction on other languages than English.

REFERENCES

- [1] C. Yi and Y. Tian “Text string detection from natural scenes by structure-based partition and grouping” IEEE Trans. IP, 2011.
- [2] “Text Information Extraction in Images and Video” A Survey Keechul Jung, Kwang In Kim, Anil K. Jain
- [3] Chucai Yi; YingLi Tian, "Text Detection in Natural Scene Images by Stroke Gabor Words," Document

- Analysis and Recognition (ICDAR), 2011 International Conference on , vol., no., pp.177,181, 18-21 Sept. 2011
doi: 10.1109/ICDAR.2011.44
- [4] Chucai Yi; YingLi Tian, "Localizing Text in Scene Images by Boundary Clustering, Stroke Segmentation, and String Fragment Classification," *Image Processing, IEEE Transactions on* , vol.21, no.9, pp.4256,4268, Sept. 2012
- [5] Epshtein, B.; Ofek, E.; Wexler, Y., "Detecting text in natural scenes with stroke width transform," *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on* , vol., no., pp.2963,2970, 13-18 June 2010
- [6] Shivakumara, P.; Trung Quy Phan; Tan, C.L., "A Laplacian Approach to Multi-Oriented Text Detection in Video," *Pattern Analysis and Machine Intelligence, IEEE Transactions on* , vol.33, no.2, pp.412,419, Feb. 2011
- [7] S. M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, and R. Young, "ICDAR 2003 robust reading competitions," in *Proc. 7th Int. Conf. Document Anal. Recognit.*, 2003, pp. 682–687.
- [8] S. M. Lucas, "ICDAR 2005 text locating competition results," in *Proc. Int. Conf. Document Anal. Recognit.*, 2005, vol. 1, pp. 80–84.
- [9] John Canny. A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-8(6):679–698, Nov. 1986.(canny)
- [10] <http://algoal.essex.ac.uk/icdar/Datasets.html>
- [11] [http://MSRA Text Detection 500 Database \(MSRA-TD500\) - TC11.html](http://MSRA Text Detection 500 Database (MSRA-TD500) - TC11.html)