Asian Journal of Information Technology 15 (21): 4318-4324, 2016
ISSN: 1682-3915

Text Detection and Recognition Using Stroke Width Transform (SWT)

[1]S. Anila and [2]N. Devarajan
[1]Department of Electronics and Communication Engineering,
Sri Ramakrishna Institute of Technology, Pachapalayam, Perur Chettipalayam,
641010 Coimbatore, India
[2]Department of Electrical and Electronics Engineering, Government College of Technology,
641013 Coimbatore, India

Abstract: A technique is proposed to detect and recognize the texts. Localizing and reading text in uncontrolled environments remain extremely challenging, due to various interference factors. An integrated framework for text detection and recognition of varying natural images such as in vision applications, for instance, image understanding, image indexing, video search, geolocation and automatic navigation is proposed. The technique involves: Text detection and recognition using texts of varying orientation with dictionary search. Thus in this proposed system, a novel image database with text of different scales s colors, orientations is designed which can be detected and recognized synchronously. Low quality noisy images can also be included in the database.

Key words: Component, formatting, style, styling, insert

INTRODUCTION

Detecting text in natural images, as opposed to scans of printed pages, faxes and business cards, is an important step for a number of Computer Vision applications, such as computerized aid for visually impaired, automatic geo coding of businesses and robotic navigation in urban environments. Retrieving texts in both indoor and outdoor environments provides contextual clues for a wide variety of vision tasks. Some of the sample texts from natural scene of image is shown in Fig. 1.

Text data present in images and video contain useful information for automatic annotation, indexing and structuring of images. Extraction of this information involves:

Detection, localization, tracking, extraction, enhancement and recognition. Recognition of texts from natural images can be performed based on features, the shape based features. Geometric Blur and Shape Context, consistently outperformed Scale Invariant Feature Transform (SIFT) as well as the appearance based features (Campos *et al*., 2009).

Significance of text in an image: Text within an image is of particular interest as:

Fig. 1: Sample text in natural scene of image

- It is very useful for describing the contents of an image
- It can be easily extracted compared to other semantic contents
- It enables applications such as keyword-based image search, automatic video logging and text-based image indexing

Text string detection consists of two steps; image partition to find text character candidates based on local gradient features and color uniformity of character components.

Character candidate grouping to detect text strings based on joint structural features of text characters in

Corresponding Author: S. Anila, Department of Electronics and Communication Engineering,
Sri Ramakrishna Institute of Technology, Pachapalayam, Perur Chettipalayam, 641010 Coimbatore, India

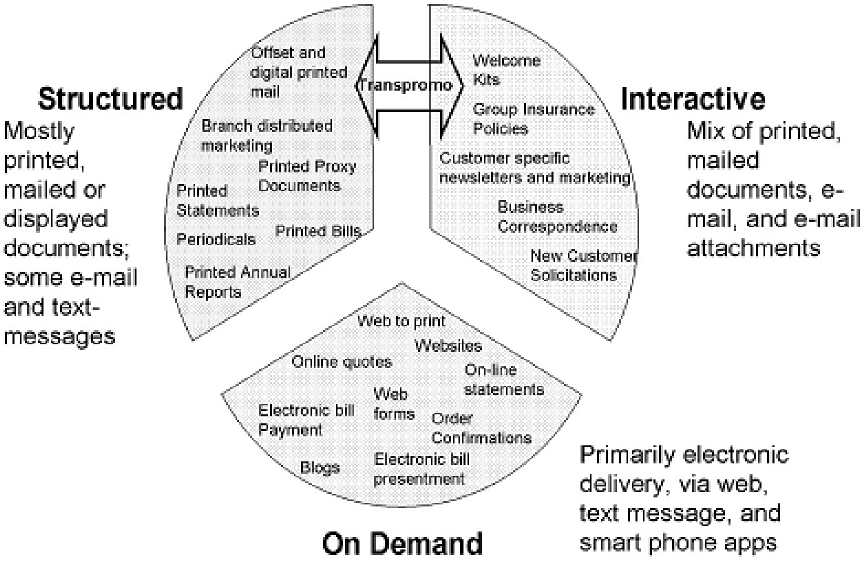

Structured
Mostly printed, mailed or displayed documents; some e-mail and text-messages
Offset and digital printed mail
Branch distributed marketing
Printed Proxy Documents
Printed Statements
Periodicals
Printed Bills
Printed Annual Reports
Transpromo
Welcome Kits
Group Insurance Policies
Customer specific newsletters and marketing
Business Correspondence
New Customer Solicitations
Interactive
Mix of printed, mailed documents, e-mail, and e-mail attachments
Web to print
Websites
Online quotes
On-line statements
Web forms
Electronic bill Payment
Order Confirmations
Blogs
Electronic bill presentment
On Demand
Primarily electronic delivery, via web, text message, and smart phone apps

Fig. 2: Document text image

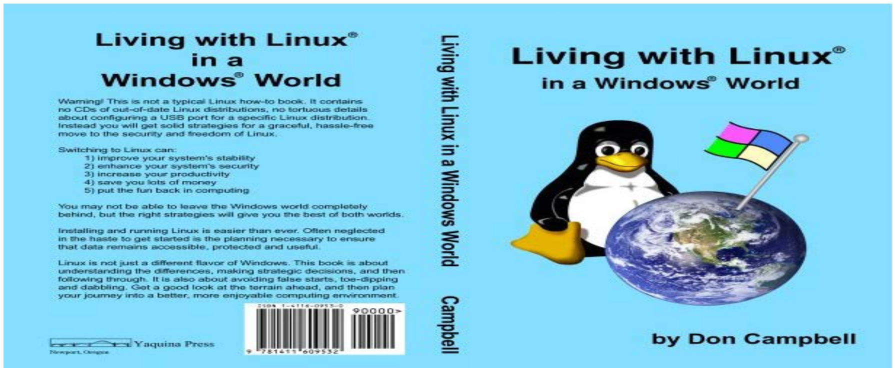
Fig. 3: Multi-color document images: each text line may or may not be of the same color

each text string such as character size differences, distances between neighbouring characters and character alignment. Visual recognition is a process os authenticating a true identity.

Primary property of scene text: The primary property of scene text such as, high contrast against background, uniform colors are difficult to preserve in real application. When the system scans whole image for texts, text pixels with low contrast and non uniform lighting could be confused as background due to similar colors. Indexing images or videos requires information about their content. Some of the document text image and colored text image are shown in Fig. 2-4. This content is often strongly related to the textual information appearing in them which can be divided into two groups:

- Text appearing accidentally in an image that usually does not represent anything important related to the content of the image. Such texts are referred to as scene text
- Text produced separately from the image is in general a very good key to understand the image and is called artificial text

In contrast to scene text, artificial text is not only an important source of information but also a significant entity for indexing and retrieval purposes. Localization of



Living with Linux®
in a
Windows® World

Warning! This is not a typical Linux how-to book. It contains no CDs of out-of-date Linux distributions, no tortuous details about configuring a USB port for a specific Linux distribution. Instead you will get solid strategies for a graceful, hassle-free move to the security and freedom of Linux.

Switching to Linux can:
1) improve your system's stability
2) enhance your system's security
3) increase your productivity
4) save you lots of money
5) put the fun back in computing

You may not be able to leave the Windows world completely behind, but the right strategies will give you the best of both worlds.

Installing and running Linux is easier than ever. Often neglected in the haste to get started is the planning necessary to ensure that data remains accessible, protected and useful.

Linux is not just a different flavor of Windows. This book is about understanding the differences, making strategic decisions, and then following through. It is also about avoiding false starts, toe-dipping and dabbling. Get a good look at the terrain ahead, and then plan your journey into a better, more enjoyable computing environment.

ISBN 1-4116-0953-0
90000>
9 781411 609532

Yaquina Press
Newport, Oregon

Living with Linux in a Windows World Campbell

Living with Linux®
in a Windows® World

by Don Campbell

Fig. 4: Colored text image

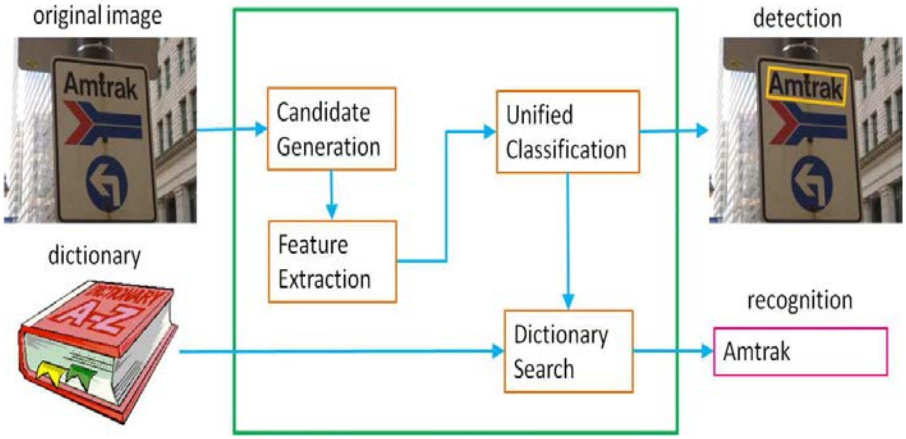


Fig. 5: Block diagram of the proposed system

text and simplification of the background in images is the main objective of automatic text detection approaches.

MATERIALS AND METHODS

Block diagram of the proposed system is shown in Fig. 5 and the various steps involved in the system are as follows:

Candidate generation: The candidates characters are initially generated from the image using Stroke Width Transform(SWT), (Epshtein *et al*., 2010). The extracted images are partitioned. Then clustering is done using clustering algorithm. A novel image operator that seeks to find the value of stroke width for each image pixel and demonstrate its use on the task of text detection in natural images can be used based on the operator that is local and data dependent, which makes it fast and robust enough to eliminate the need for multi-scale computation or scanning windows (Epshtein *et al*., 2010).

Feature extraction: Feature extraction starts from an initial set of measured data and builds derived values (features) intended to be informative, non redundant, facilitating the subsequent learning and generalization steps, in some cases leading to better human interpretations. Feature extraction is related to dimensionality reduction. When the input data to an algorithm is too large to be processed and it is suspected to be redundant (e.g., the same measurement in both feet and meters, or the repetitiveness of images presented as pixels), then it can be transformed into a reduced set of features. This process is called feature extraction. The extracted features are expected to contain the relevant information from the input data, so that the desired task can be performed by using this reduced representation instead of the complete initial data. Feature extraction involves reducing the amount of resources required to describe a large set of data. When performing analysis of complex data one of the major problems stems from the number of variables involved. Analysis with a large number of variables generally requires a large amount of memory and a classification algorithm which overfits the training sample and generalizes poorly to new samples. Feature extraction is a general term for methods of constructing combinations of the variables to get around these problems while still describing the data with sufficient accuracy. Redundant pixels of the image are removed.

Unified classification: The proposed algorithm uses component level classifier which consists of four stages:

Component extraction: At this stage, edge detection is performed on the original image and the edge map is fed to the SWT module to produce an SWT image. Neighbouring pixels in the SWT image are grouped together recursively to form connected components using a simple association rule. To extract connected components from the image, SWT is adopted for its effectiveness and efficiency. In addition, it provides a way to discover connected components from edge map directly, which makes it unnecessary to consider the factors of scale and direction.

Component analysis: Many components extracted at the component extraction stage are not parts of texts. The component analysis stage is aimed to identify and filter out those non-text components. First, the components are filtered using a set of heuristic rules that can distinguish between obvious spurious text regions and true text regions. Next, a component level classifier is applied to prune the non-text components that are hard for the simple filter. The purpose of component analysis is to identify and eliminate the connected components that are unlikely parts of texts (Yi and Tian, 2011).

Candidate linking: The remaining components are taken as character candidates. The first step of the candidate linking stage is to link the character candidates into pairs. Two adjacent candidates are grouped into a pair if they have similar geometric properties and colors. At the next step, the candidate pairs are aggregated into chains in a recursive manner. The character candidates are aggregated into chains at this stage. This stage also serves as a filtering step because the candidate characters cannot be linked into chains and are taken as components accidentally formed by noises or background clutters and thus are discarded. First the character candidates are linked into pairs. Whether two candidates can be linked into a pair is determined based on the heights and widths of their bounding boxes. However, bounding boxes are not rotation invariant, so their characteristic scales can be used. If two candidates have similar stroke widths (ratio between the mean stroke widths is <2.0), similar sizes (ratio between their characteristic scales does not exceed 2.5), similar colors and are close enough (distance between them is less than two times the sum of their characteristic scales), they are labelled as a pair.

Chain analysis: At the chain analysis stage, the chains determined at the former stage are verified by a chain level classifier. The chains with low classification scores (probabilities) are discarded. The chains may be in any direction, so a candidate might belong to multiple chains; the interpretation step is aimed to dismiss this ambiguity. The chains that pass this stage are the final detected texts. The candidate chains formed at the previous stage might include false positives that are random combinations of scattered background clutters (such as leaves and grasses) and repeated patterns (such as bricks and windows). To eliminate these false positives, a chain level classifier is trained using the chain level features. The chains whose total probabilities are lower than a threshold T are discarded.

Various steps involved in the proposed system: Flow chart of the proposed system is shown in Fig. 6.

Step 1: load image: The text can be rotated in plane, but significant out of plane rotations may require additional pre-processing.

Step 2: detect mser regions: Since text characters usually have consistent color, it should begin by finding regions of similar intensities in the image using the Maximally Stable Extremal Regions (MSER) region detector.

Step 3: use canny edge detecter to further segment the text: Since written text is typically placed on clear background, it tends to produce high response to edge detection. Furthermore, an intersection of MSER regions with the edges is going to produce regions that are even more likely to belong to text.

Step 4: filter character candidates using connected component analysis: Some of the remaining connected components can now be removed by using their region properties. The thresholds used below may vary for different s, image sizes or languages.

Step 5: Filter character candidates using the stroke width image: Another useful discriminator for text in images is the variation in stroke width within each text candidate.

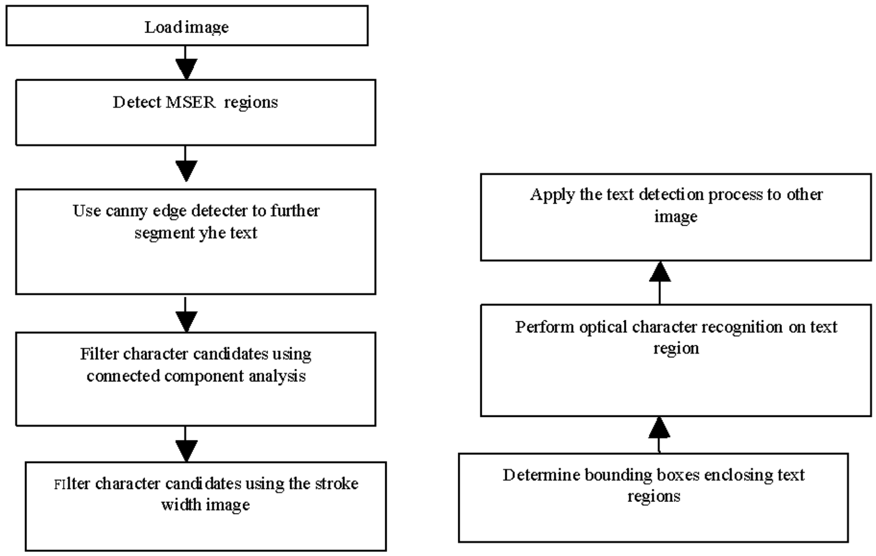

Load image
Detect MSER regions
Use canny edge detecter to further segment yhe text
Filter character candidates using connected component analysis
FIlter character candidates using the stroke width image
Determine bounding boxes enclosing text regions
Perform optical character recognition on text region
Apply the text detection process to other image

Fig. 6: Flow chart of the proposed system

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. The stroke width image below is computed using the helperStrokeWidth helper function.
Steps for SWT are as follows:

- Initially edges in the image are computed using canny edge detector.
- Gradient direction dp of each edge pixel ‘p’ is to be considered
- If ‘p’ lies on a stroke boundary, then dp must be perpendicular to the orientation of the stroke.
- The ray is found using, r = p+n.dp, n>0
- Now consider gradient direction dq which is opposite to dp
- Finally, if matching pixel ‘q’ is not found(or) if dq is not opposite to dp, the ray is discarded

Step 6: determine bounding boxes enclosing text regions: To compute a bounding box of the text region, the individual characters are merged into a single connected component. This can be accomplished using morphological closing followed by opening to clean up any outliers.

Step 7: perform optical character recognition(ocr) on text region: The segmentation of text from a cluttered scene can greatly improve OCR results. Since our algorithm already produced a well segmented text region, we can use the binary text mask to improve the accuracy of the recognition results. The goal of Optical Character Recognition (OCR) is to classify optical patterns (often contained in a digital image) corresponding to alpha numeric or other characters (Tsai *et al*., 2011). The process of OCR involves several steps including segmentation, feature extraction and classification. Each of these steps is a field unto itself and is described briefly here in the context of a Matlab implementation of OCR. A few examples of OCR applications are listed here. The most common for use OCR is the first item; people often wish to convert text documents to some sort of digital representation.

- People wish to scan in a document and have the text of that document available in a word processor
- Recognizing license plate numbers
- Post office needs to recognize zip-codes

Step 8: apply the text detection process to other images: To highlight flexibility of this approach, apply the entire algorithm to other images using the helperDetectText function.

RESULTS AND DISCUSSION

The images from ICDAR 2011 are used for testing the algorithm (Shahab *et al*., 2011). The various results obtained for the input image in Figure 7 using the above steps is shown in Fig. 8-13.

The algorithm is tested with 50 images each from the two databases (ICDAR in 2011 database and real time created database). The detection rate is found to be good.

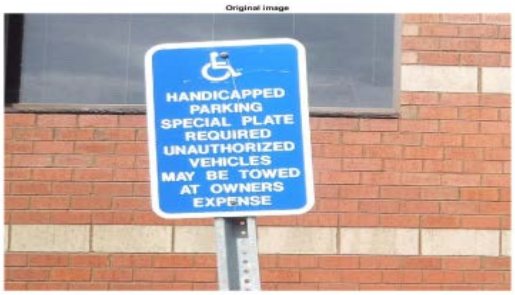


Fig. 7: Input image

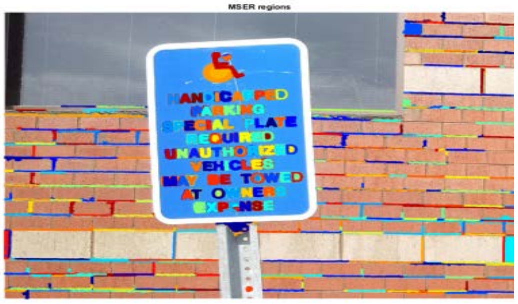


Fig. 8: MSER regions

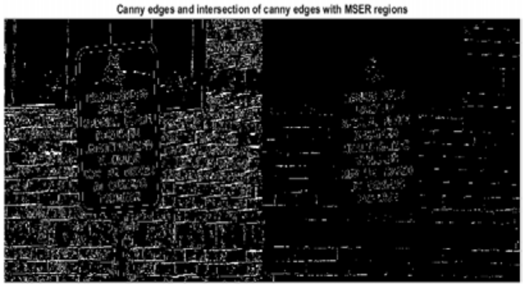

Canny edges and intersection of canny edges with MSER regions

Fig. 9: Canny edges and intersection of canny edges using MSER regions

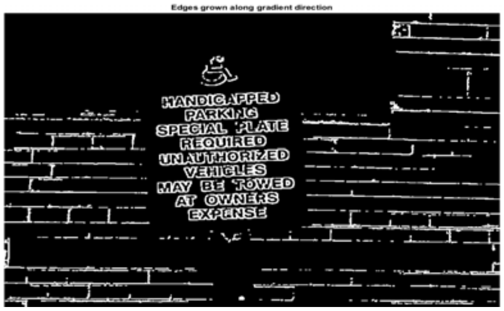


Fig. 10: Edges grown along gradient direction

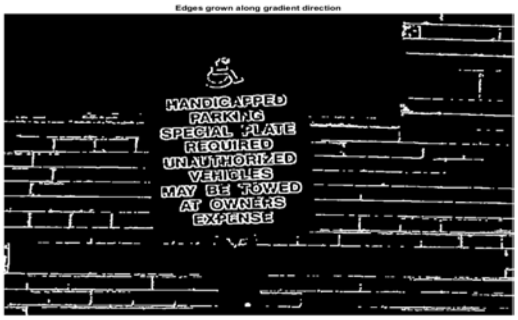


Fig. 11: Edges grown along gradient direction

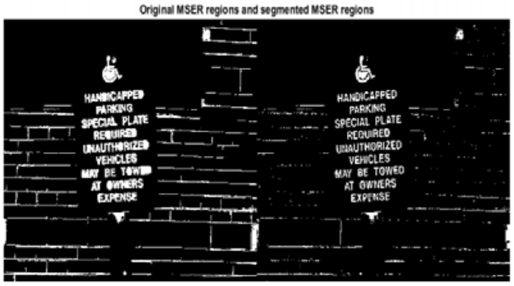


Fig. 12: Original MSER regions and segmented MSER regions

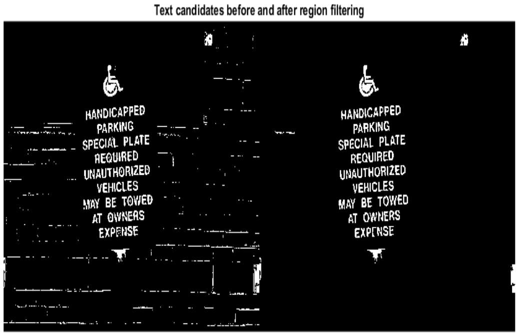


Fig. 13: Text candidates before and after region filtering

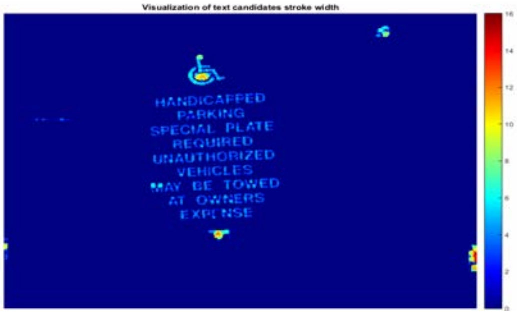
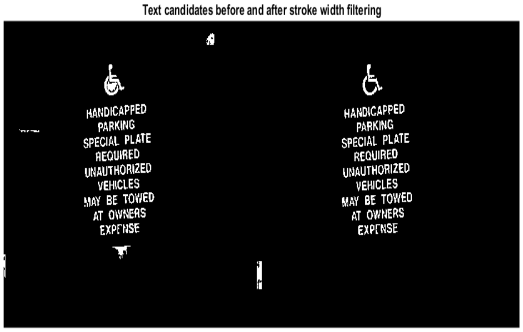


Fig. 14: Vishualization of text candidates stroke width



Text candidates before and after stroke width filtering

Fig. 15: Text candidates before and after stroke width filtering

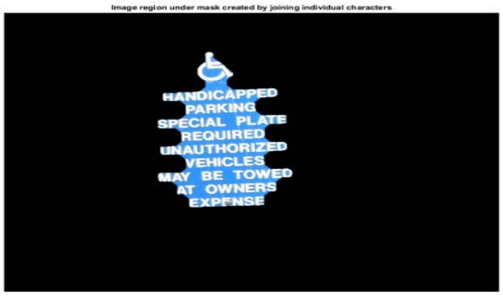
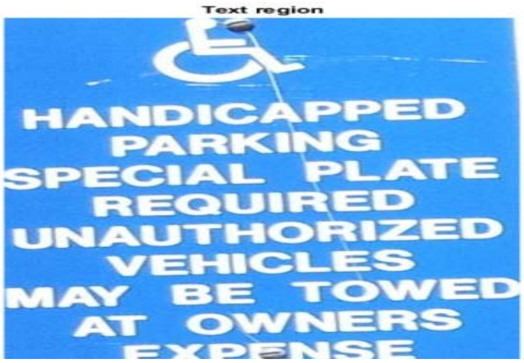


Image region under mask created by joining individual characters

Fig. 16: Image region under mask created by joining individual characters



Text region

Fig. 17: Text region

CONCLUSION

A text detection system that detects texts of arbitrary directions in complex natural scenes are detected and recognized. The algorithm has been tested for various types of images which has different scales, orientations, color and size. The testing has been proceeded for several images which are stored in the database. Furthermore, we have proposed a database with horizontal as well as non-horizontal texts. The proposed algorithm can assist numerous applications that require text information extraction from images or videos, such as video search, target geolocation and automatic navigation.

REFERENCES

Campos, D.T.E., B.R. Babu and M. Varma, 2009. Character in natural images. V.I.S.A.P.P., 2009: 273-280.

Epshtein, B., E. Ofek and Y. Wexler, 2010. Detecting text in natural scenes with stroke width transform. Proceeding of the 2010 IEEE Conference on Computer Vision and Pattern Recognition, June 13-18, 2010, IEEE, Redmond, Washington, ISBN: 978-1-4244-6984-0, pp: 2963-2970.

Shahab, A., F. Shafait and A. Dengel, 2011. ICDAR 2011 robust reading competition challenge 2: Reading text in scene images. Proceeding of the 2011 International Conference on Document Analysis and Recognition, September 18-21, 2011, IEEE, Kaiserslautern, Germany, ISSN: 1520-5363, pp: 1491-1496.

Tsai, S.S., H. Chen, D. Chen, G. Schroth and R. Grzeszczuk et al., 2011. Mobile visual search on printed documents using text and low bit-rate features. Proceeding of the 2011 18th IEEE International Conference on Image Processing, September 11-14, 2011, IEEE, New York, USA., ISBN:978-1-4577-1303-3, pp: 2601-2604.

Yi, C. and Y.L. Tian, 2011. Text string detection from natural scenes by structure-based partition and grouping. IEEE Trans. Image Processing, 20: 2594-2605.