IMPROVED SCENE TEXT DETECTION FOR NONTEXT FILTERING BASED CONNECTED COMPONENT CLUSTERING

[1]S.Senthamilsakthivel, [2]K.Prabhakaran
[1]PG Scholar, Department of ECE, SNS College of Technology, Coimbatore
[2]Assistant Professor, Department of ECE, SNS College of Technology, Coimbatore
[1]sakthi07tamil@gmail.com, [2]prabaakar.t@gmail.com

Abstract: In this paper, we present a new scene text detection algorithm based on two machine learning classifiers: one allows us to generate candidate word regions and the other filters out non-text ones. To be precise, we extract connected components (CCs) in images by using the maximally stable external region algorithm. These extracted CCs are partitioned into cluster so that we can generate candidate regions. Unlike convention almethods relying on heuristic rules in clustering, we train anAdaBoost classifier that determines the adjacency relationship and cluster CCs by using their pairwise relations. Then wenormalize candidate word regions and determine whether each region contains text or not. Since the scale, skew, and colorof each candidate can be estimated from CCs, we develop atext/nontext classifier for normalized images. This classifier is based on multilayer perceptrons and we can control recall and precision rates with a single free parameter. Finally, we extendour approach to exploit multichannel information. Experimentalresults on ICDAR 2005 and 2011 robust reading competitiondatasets show that our method yields the state-of-the-art performanceboth in speed and accuracy.

Index Terms: Connected component (CC)-based approach, CC clustering, machine learning classifier, nontext filtering, and scene text detection.

1. INTRODUCTION

SINCE mobile devices equipped with high-resolution digital cameras are widely available, research activities using these devices in the field of human computer interaction (HCI) have received much attention for the last decades. Among them, text detection and recognition in camera captured images have been considered as very important problems in computer vision community [1]–[3]. It is because text information is easily recognized by machines and can be used in a variety of applications. Some examples are aids for visually impaired people, translators for tourists, information retrieval systems in indoor and outdoor environments, and automatic robot navigation.

Although there exist a lot of research activities in this field, scene text detection is still remained as a challenging problem. This is because scene text images usually suffer from photometric degradations as well as geometrical distortions so that many algorithms faced the accuracy and/or speed (complexity) issues [4]–[6].

1.1 Related Work

Most of the scene text detection algorithms in the literature can be classified into region-based and connected component (CC)-based approaches [1]–[3]. Region-based methods [7], [8] adopted a sliding window scheme,

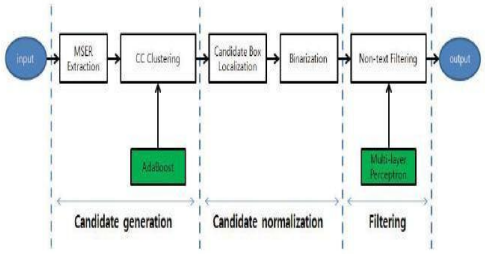


Fig. 1. Block diagram of our system.

Which is basically a brute force approach which requires a lot of local decisions. Focused on an efficient binary classification (text versus nontext) of a small image patch. In other words, they have focused on the following problem: 1) Problem (A): to determine whether a given patch is a part of a text region. For efficient classification, researchers addressed this problem by adopting cascade structures. In their approaches, simple features such as horizontal and vertical derivatives were used at the early stages of the cascade and complex features were incrementally employed [7], [8]. Even though this structure enables efficient text detection, Problem-

(A)is still challenging. It is not straightforward even for human to determine the class of a small image patch when we do not have knowledge of text properties such as scale, skew, and color. Multi-scale scheme using different window sizes can alleviate the scale issues; however, it makes boxes indifferent scales overlap. Experimental results on ICDAR2005 dataset have shown that this region based approach is efficient, however, it yields worse performance compared with CC-based approaches [5], [9], [10]. CC-based methods begin with CC extraction and localize text regions by processing only CC-level information.

Therefore, they have focused on the following problems:
Problem (B): to extract text-like CCs.
Problem (C): to filter out nontext CCs.
Problem (D): to infer text blocks from CCs.

In the literature, many CC extraction methods were developed to address Problem (B). For example, some methods assumed that the boundaries of text components should show strong discontinuities and they extracted CCs from edge maps. Others were inspired by the observation that text is written in the same color and they applied color segmentation (or reduction) techniques [11]. On the other hand, some researchers developed their own CC extraction methods from the scratch: the curvilinearity of text was exploited in [10], [] and local binarization by using estimated scales was adopted in [9].After the CC extraction, CC-based approaches filter outnontext CCs. In the end, features such as "aspect ratio," "the number of holes in a CC," and "the variance of the stroke width within each CC" were employed in [10],. In [9], conditional random fields (CRFs) were adopted in order to consider binary (relational) features as well as unary features. In, a neural network was used to filter out nontextcomponents.

Finally, CC-based approaches infer text blocks from theremaining CCs. This step is also known as text line aggregation, text line formation, or text line grouping. Interestingly, many methods were based on similar rules. For example, theheight ratio between two letters and color difference has been used in a number of methods [10]–. Although CC-based approaches have shown better performance than region-based ones, they usually suffer from the computational complexity.It is because their performances depend on the quality ofCCs and they adopted sophisticated CC extraction and filtering methods [9], [10], , [15].

1.2 Our Appr**o**ach

We have illustrated the block diagram of our system in Fig. 1. As shown in the figure, our method consists of three steps: candidate generation, candidate normalization, and nontext filtering. Our candidate generation step is based on popular CC-based approaches; however, we have focused on Problem-(D) rather than Problem-(B) and(C). That is, we use an efficient CC extraction method and we do not adopt CC filtering ideas which are usually time-consuming. Rather, we address the problems caused by the absence of low level CC processing with the ideas in region-based approaches, i.e. ,we address Problem-(A). We can normalize candidate regions with CC-level information and this normalization allows us to build a simple but reliable text/nontext classifier.

In our approach, both problems ((D) and (A)) are addressed based on machine learning techniques, so that our method is largely free from heuristics. We have trained a classifier that determines adjacency relationship between CCs for Problem-(D) and we generate candidates by identifying adjacent pairs. In training, we have selected efficient features and trained the classifier with the Ada Boost algorithm .As mentioned, we apply the above CC clustering method to raw CC sets (that usually contains many nontext CCs)and some candidates may correspond to "nontext clusters." Therefore, nontext rejection scheme should be followed, which is the main problem of region-based methods. However, our situation is different from conventional ones because we can use normalized inputs in classification. We estimate the skew and scale of a candidate from the distribution of CCs, and estimate text and background colors from the CCs. Examples of our normalized results are shown in Fig. 8(f). Finally, we reject non-texts among normalized candidates with a multi-layer perceptron that is trained with the back-propagation algorithm, Our CC extraction algorithm is the maximally stable external region (MSER) algorithm that is invariant to scales and affine intensity changes, and other blocks in our method are also designed to be invariant to these changes. This in variance allows us to exploit multi-channel information: we can apply our method to multiple channels at the same time and treat their outputs as if

they are from a single source. In this way, we are able to detect text that is not salient in luminance channel images. We submitted our preliminary results to 2011 ICDAR robus treading competition [6] and won the first prize. Experimentalresults on 2005 competition results have also shown that our method yields better performance with small computational complexity. The rest of this paper is organized as follows In Section II, we explain the way of adding more information to ground truth and explain frequently used notations in this paper. We present our candidate generation method in Section III, which consists of a MSER- based CC extraction block and an Ada Boost-based CC clustering block .In Section IV and V, we present our normalization method and nontext rejection algorithm respectively. Note that normalization in our paper means not only geometric normalization but also binarization (color normalization). Finally, we show experimental results in Section VI, and conclude the paper in Section VII.

2. AUGMENTED GROUND TRUTH ANDFREQUENTLY USED NOTATIONS

2.1 Construction of New Ground Truth

The ICDAR dataset consists of natural images annotatedwith bounding boxes around each instance of a word [4]– [6].Although it contains sufficient information in performanceevaluation, we need more information for the training ofour classifiers.

Fig. 2. (a) Input image. (b) Illustration of our ground truth. We have pixel-level annotations (binarization results) and text-line numbers. (c) MSER results of (a). We assigned random colors to CCs for better visibility. Note that many CCs are nested due to the properties of stable regions.

As shown in Fig. 1, we use a classifier that tells us the adjacency relation between CCs in the candidate generation step, and CC-level information is essential for the training of such a classifier (which will be clarified in Section III-B). Therefore, we augmented ground truth that was released as 2011 ICDAR training set by adding pixel-level annotations (binarization results) and text-line information. For example, in case of Fig. 2(a), there was bounding box information of four words in original ground truth. In addition to them, we build binarization results as shown in Fig. 2(b) and assign a text-line number for each CC. That is, we assign "1"to CCs in a box containing "SUMMER," assign "2" to CCs ina box containing "WHATEVER" and "THE," and assign "3"to CCs in a box containing "WEATHER."

2.2 Frequently Used Notations

We say that two CCs are adjacent when both are text components in the same word and the number of characters between them is less than 2. For a word "WHATEVER," we say that "W is adjacent to H" and "W is adjacent to A, however, we say that "W is not adjacent to T" because there are two characters (i.e., "H" and "A") between them. We also use a notation

ci€c j when ci and c j are adjacent, and ci_ c j otherwise. Let t (ci)denote the text- line number of ci..

Note that ci ∼ c j means that t (ci)= t (c j), however, t (ci) = t (c j)does not necessarily mean ci €c j . In addition, µi€_3indicates the mean color of pixels in ci, and si= |ci | where| · | indicates the number pixels in a given CC. Let us assume that we apply a CC extraction method (the MSER algorithm in our case) to an image and get a set of CC sas illustrated in Fig. 2(c). Fig. 3 shows some CCs excerpted from Fig. 2(c) for clarity. We denote the set of CCs as C, and partition this set into a text component set T and a nontext component set N:

$$T \cup N = C \quad (4)$$
$$T \cap N = \emptyset \quad (5)$$

Here we consider an element in C as a text component when there is a corresponding CC in the ground truth binary image. In other words, a CC in Fig. 2(c) is considered as an element of T if there exists a corresponding CC in Fig. 2(b). However ,it is very unlikely that c ∼C is identical to a certain CC in the ground truth binary image, and we relax this condition. That is, c ∼T means that there exists a CC, i.e., e, in the ground truth binary image satisfying

$$|c \cap e| \geq 0.8 \times |c| \quad (6)$$
$$|c \cap e| \geq 0.8 \times |e|. \quad (7)$$

Fig. 3. Illustration of possible relations between CCs. For clarity, we have only illustrated some CCs in $\mathcal{C}$.

For example, a CC "S" in Fig. 3 is considered as a text component because there is a similar CC in Fig. 2(b). On the other hand, a CC consisting of "M" and "E" in Fig. 3 is different from any CCs in Fig. 2(b) and it is considered as an on text component.

3. CANDIDATE GENERATION

For the generation of candidates, we extract CCs in images and partition the extracted CCs into clusters, where our clustering algorithm is based on an adjacency relation classifier. In this section, we first explain our CC extraction method. Then, we will explain our approaches (i) to build training samples,(ii) to train the classifier, and (iii) to use that classifier in our CC clustering method.

3.1 CC Extraction

Among a number of CC extraction methods, we have adopted the MSER algorithm because it shows good performance with a small computation cost , . This algorithm can be considered as a process to find local binarization results that are stable over a range of thresholds, and this property allows us to find most of the text components,[15]. The MSER algorithm yields CCs that are either darker or brighter than their surroundings. In Fig. 2(c), we have illustrated brighter CCs by assigning random colors to them .Note that many CCs are overlapping due to the properties of stable regions . More MSER extraction examples can be found in Fig. 8(a) and (b), which show brighter and darker CCs, respectively.

3.2 Building Training Sets

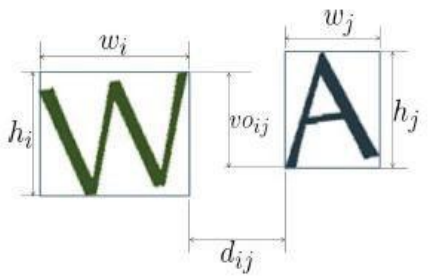
Our classifier is based on pairwise relations between CCs, and let us first consider cases that can happen for a CC pair

$(c_i, c_j) \in \mathcal{C} \times \mathcal{C}\ (i \neq j)$:

1) $c_i \in \mathcal{T}, c_j \in \mathcal{T}, c_i \sim c_j$
2) $c_i \in \mathcal{T}, c_j \in \mathcal{T}, c_i \nsim c_j, t(c_i) = t(c_j)$
3) $c_i \in \mathcal{T}, c_j \in \mathcal{T}, c_i \nsim c_j, t(c_i) \neq t(c_j)$
4) $c_i \in \mathcal{T}, c_j \in \mathcal{N}$
5) $c_i \in \mathcal{N}, c_j \in \mathcal{N}$.

We have illustrated them in Fig. 3. In a strict sense, we may have to classify the case (1) from the (2) ~(5) cases (of course, this classification will allow us to find all words having more than one character). However, it is not straight forward to train such a classifier. For example, let us consider "R" and "T" in the second line in Fig. 2(b). It is not straight forward to determine whether they are in the same word without considering other characters. Therefore, rather than focusing on this difficult problem, we address a relatively simple problem by adopting an idea in region-based approaches. That is, we adopt a nontext filtering block as shown in Fig. 1, and we are no longer required to care about the case (5). If we have ci ~c j for some ci , c j €N, it will yield a candidate consisting of nontext CCs and this candidate will be rejected at the nontext rejection step. Also, we will perform word segmentation as a post processing step and the case (2) does not mean negative samples. Based on these observations, we build training sets. Specifically, we first obtained sets of CCs by applying the MSER algorithm to a training set released by[6]. Then, for every pair (ci , c j) €C ×C (i ≠j), we identify its category among 5 cases. A positive set is built by gathering samples corresponding to the case (1) and a negative set by gathering samples corresponding to the case (3) or (4). Samples from other cases were discarded. Note that this process can be automated by using our

augmented ground truth in Section II-A



] Fig. 4. Illustration of local properties between two CCs.

3.3 CC Clustering

The Ada Boost algorithm yields a function φ :C × C →_ and we use this function in binary decisions: φ(ci , c j) > τ1 ⇐⇒ ci ~c j with a threshold τ1. Given φ(·, ·) and τ1, we can find all adjacent pairs by evaluating that function for all possible pairsin C. Based on these adjacency relations, C is partitioned into a set of clusters W = {wk} where wk€C. Formally speaking, ci , c j €wk(ie., ci and c jare in the same cluster) means that there exists {ei}mi=1€Csuch that ci ~e1 ~e2 ~ · · · em€c j . We build W by using the union-find algorithm [23]. After clustering, we have discarded clusters having only one CC.

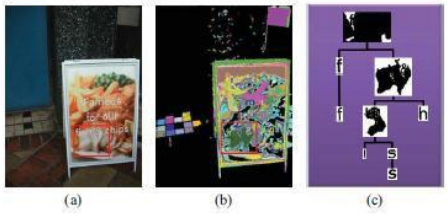


Fig. 5. (a) Input image. (b) Color-coded MSER results. (c) Hierarchical tree of MSER results in the red box in (b).

Fig. 6. (a) Input image. (b) Hierarchical tree of MSER results in the red box in (a).

3.4 Comparison to Other MSER-Based Methods

The MSER algorithm has desirable properties for text detection:(i) detection results are invariant to affine transformation of image intensities and (ii) no smoothing operation is involved so that both very fine and very large structures can be detected at the same time . Therefore, the algorithm has been adopted in many methods –[15]. However, unlike our approach, they focused on the retrieval of CCs corresponding to individual characters: the authors in developed a variant of MSER in order to prevent the merging of individual characters, and a Support Vector Machine (SVM) based classifier was developed for the character and non-character classification in [15]. That is, they tried to develop MSER-based CC extractors yielding individual characters (i.e., high precision and high recall).

On the other hand, we mainly focus on retrieving the text components as much as possible. As a result, redundant and noisy CCs could be involved in finding clusters. As shown in Figs. 5(c) and 6(b), due to the characteristics of the MSER algorithm, some characters are detected more than once and there are lots of nontext components. Moreover, some of them do not correspond to individual characters (e.g., "ST" and "RS" in Fig. 6). The advantages of our approach areits efficiency and robustness.

Fig. 7. We denote points in B_k as red. By computing the angles connecting two points in B_k, we can estimate the skew of a word.

Our method can be efficiently implemented because CC-level feature extraction and classification are not involved. We can also deal with the variations of characters (caused by the font variations and blurs) because we do not exploit the features of individual characters (our algorithm successfully detects texts in Fig. 6(a)). This approach has drawbacks that text regions could be overlapping and nontext regions are sometimes detected, which will be addressed in the following sections.

4. CANDIDATE NORMALIZATION

After CC clustering, we have a set of clusters. In this section, we normalize corresponding regions for the reliable text/nontext classification.

4.1 Geometric Normalization

Given wk €W, we first localize its corresponding region. Even though text boxes can experience

perspective distortions, we approximate the shape of text boxes with parallelograms whose left and right sides are parallel to y-axis. This approximation alleviates difficulties in estimating text boxes having a high degree of freedom (DOF): we only have to find a skew and four boundary supporting points. To estimate the skew of a given word candidate wk , we build two sets:

Tk= {t (ci)|ci €wk}

Bk= {b(ci)|ci €wk }

wheret (ci) and b(ci) are the top-center point and the bottom center point of a bounding box of ci , respectively. We illustrate Bkin Fig. 7.

Fig. 8. (a) and (b) Brighter and darker CCs extracted by the MSER algorithm [16]. (c) We localize candidate regions in image domains. (d) Nontext blocks are filtered out by exploiting the statistical properties of the regions (our final results). (e) Geometrically normalized results of localized boxes. (f) Our normalization results.

For every pair in Bk and Tk, the slope of a line connecting the pair is discretized into one of 32 levels in [−π8, π8], and each pair votes for the skew angle. After voting, the most common angle is considered as a skew .Localized blocks are shown in Fig. 8(c). Then, we perform geometric normalization by applying an affine mapping that transforms the corresponding region to a rectangle. During the transformation, we use a constant target height (48 pixels in experiments) and preserve the aspect ratio of the box. Geometrically normalized results of localized blocks in Fig. 8(c)are shown in Fig. 8(e).

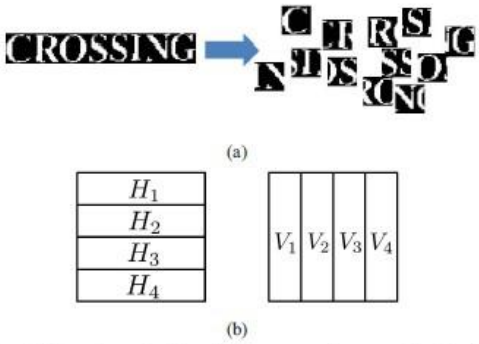


Fig. 9. (a) In order to handle variable aspect ratios, we split a block into squares. (b) For the feature extraction, we divide a square block into four horizontal and four vertical blocks.

4.2 Binarization

Given geometrically normalized images, we build binary images. In many cases, MSER results can be considered as binarization results as shown in Fig. 5(c). However, we perform the binarization separately by estimating text and background colors. It is because (i) the MSER results may miss some character components and/or yield noisy regions(mainly due to the blur) and (ii) we have to store the point information of all CCs for the MSER-based binarization. We consider the average color of CCs as the text color and consider the average color of an entire block as the background color. Then, we obtain a binary value of each pixel by comparing the distances to the estimated text color and the estimated background color. We have used l2 norm in RGB space. The binarization results of Fig. 8(e) are shown in Fig.8(f).

5. TEXT/NONTEXT CLASSIFICATION

In order to get final results like Fig.8(d) from Fig. 8(c) and (f), we develop a text/nontext classifier that rejects nontext blocks among normalized images. In our classification, we do not adopt sophisticated techniques such as cascade structures, since the number of samples to be classified is usually small. However, one challenge for our approach is the variable aspect ratio as shown in Fig. 8(f). One possible approach to this problem is to split the normalized images into patches covering one of the letters and develop a character/non- character classifier as [15]. However, character segmentation is not an easy problem [24] and there are examples where this approach may fail (See Fig. 6). Rather, we split a normalized block into overlapping squares as

illustrated in Fig. 9(a), and develop a classifier that assigns a textnessvalue to each square block. Finally, decision results for all square blocks (in the right hand side of Fig. 9(a)) are integrated so that the original block (in the left hand side) is classified. In this section, we first present our training method that allows us to have a text ness value for each square. Then, we explain our text/nontext classification method for normalized images such as Fig. 8(f).

Fig. 10. (a) Our detection results. (b) Transformed results of (a) for the comparison with the rectangle-based ground truth.

5.1 Feature Extraction from a Square Block

Our feature vector is based on mesh and gradient features as adopted in [25]. We divide each square into 4 horizontal and vertical ones as shown in Fig. 9(b) and extract features. For a horizontal block Hi (i = 1, 2, 3, 4), we consider the number of white pixels, the number of vertical white-black transitions, the number of vertical black-white transitions as features, and features for a vertical block is similarly defined.

5.2 Multilayer Perceptron Learning

For the training, we need normalized images such as Fig. 8(f). For this goal, we applied our algorithm presented in the previous sections (i.e., candidate generation and normalization algorithms) to the training images in [6]. Then, we manually classified them into text and nontext. We discarded some images showing poor binarization results, and collected676 text block images and 863 nontext block images. However ,we have found that more negative samples are needed for there liable rejection of nontext components and collected more negative samples by applying the same procedure to images that do not contain any text. Finally, we have 3, 568 nontext images. These text/nontext images are divided into squares as illustrated in Fig. 9(a) and we have trained a multi- layer perceptron for the classification of square patches .We use one hidden layer consisting of 20 nodes and set the output value to +1 for text samples and 0 otherwise. To help the learning, input features are normalized.

5.3 Integration of Decision Results

For the integration of square classification results, we accumulate the outputs of the classifier: ψ(wk) =_i€PFi whereP is the square patch set (e.g., the right hand side ofFig. 9(a)) and Fi is the continuous output of the classifier forthe i - th square block in P. We consider ψ(wk) as a textnessmeasure and classify wk as a text region when1 |P|ψ(wk) > τ2 where|P| is the number of patches. Textness measure (19) is also useful for imposing anon- overlap constraint, i.e., two text blocks should not be overlapping. When two localized regions are significantly overlapping, we simply choose a block showing a higher ψ(·)value.

6. CONCLUSION

In this paper, we have presented a novel scene text detection algorithm based on machine learning techniques. To be precise, we developed two classifiers: one classifier was designed to generate candidates and the other classifier was for the filtering of nontext candidates. We have also presented an ovel method to exploit multi-channel information. We have conducted experiments on ICDAR 2005 and 2011 datasets which showed that our method yielded the state-of- the-art performance in both new and traditional evaluation protocols.

REFERENCES

[1] K. Jung, “Text information extraction in images and video: A survey, “Pattern Recognition., vol. 37, no. 5, pp. 977– 997, May 2004.

[2] J. Liang, D. Doermann, and H. Li, “Camera-based analysis of text and documents: A survey,” Int. J. Document Anal. Recognit., vol. 7, nos. 2–3,pp. 84–104, 2005.

[3] J. Zhang and R. Kasturi, “Extraction of text objects in video documents: Recent progress,” in Proc. 8th IAPR Int. Workshop Document Anal. Syst.,Sep. 2008, pp. 5–17.

[4] S. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, and R. Young,“ICDAR 2003 robust reading

competitions," in Proc. Int. Conf. Document Anal. Recognit., 2003, pp. 682– 687.

[5] S. Lucas, "Icdar 2005 text locating competition results," in Proc. Int.Conf. Document Anal. Recognit., 2005, pp. 80– 84.

[6] A. Shahab, F. Shafait, and A. Dengel, "ICDAR 2011 robust reading competition challenge 2: Reading text in scene images," in Proc. Int.Conf. Document Anal. Recognit., 2011, pp. 1491–1496.

[7] X. Chen and A. Yuille, "Detecting and reading text in natural scenes,"in Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., 2004,pp. 366–373.

[8] X. Chen and A. Yuille, "A time-efficient cascade for real- time object detection: With applications for the visually impaired," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Workshops, Jun. 2005, pp. 1–8.

[9] Y.-F. Pan, X. Hou, and C.-L. Liu, "A hybrid approach to detect and localize texts in natural scene images," IEEE Trans. Image Process., vol. 20, no. 3, pp. 800–813, Mar. 2011.

[10] B. Epshtein, E. Ofek, and Y. Wexler, "Detecting text in natural scenes with stroke width transform," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2010, pp. 2963–2970.

[11] C. Yi and Y. Tian, "Text string detection from natural scenes by structure-based partition and grouping," IEEE Trans. Image Process.,vol. 20, no. 9, pp. 2594–2605, Sep. 2011.