Symmetry-Based Text Line Detection in Natural Scenes

Zheng Zhang[1] Wei Shen[2] Cong Yao[1] Xiang Bai[1*]
[1] School of Electronic Information and Communications, Huazhong University of Science and Technology
[2] Key Lab of Specialty Fiber Optics and Optical Access Networks, Shanghai University

Abstract

Recently, a variety of real-world applications have triggered huge demand for techniques that can extract textual information from natural scenes. Therefore, scene text detection and recognition have become active research topics in computer vision. In this work, we investigate the problem of scene text detection from an alternative perspective and propose a novel algorithm for it. Different from traditional methods, which mainly make use of the properties of single characters or strokes, the proposed algorithm exploits the symmetry property of character groups and allows for direct extraction of text lines from natural images. The experiments on the latest ICDAR benchmarks demonstrate that the proposed algorithm achieves state-of-the-art performance. Moreover, compared to conventional approaches, the proposed algorithm shows stronger adaptability to texts in challenging scenarios.

1. Introduction

The mass popularization of smart phones and rapid development of the Internet have brought forth tremendous new products and services, which have triggered huge demand for practical vision techniques. Scene text detection and recognition, affording a way to directly access and utilize the textual information in natural scenes, are obviously among the most pressing techniques. Consequently, text localization and recognition in natural scenes have attracted much attention from the computer vision community and document analysis community.

Though extensively studied in the past decade [5, 7, 25, 35, 41, 27, 3, 42, 12], detecting and reading texts in natural scenes are still difficult tasks. The major challenges stem from three aspects [43]: (1) Diversity of scene text: Texts in uncontrolled environments may exhibit entirely different fonts, colors, scales and orientations; (2) Complexity of background: The backgrounds in natural scenes can be very complex. Elements like signs, fences, bricks and grasses are virtually undistinguishable from true text, and thus are easily to cause confusions and errors; (3) Interference factors: Various interference factors, such as noise, distortion, low resolution, non-uniform illumination and partial occlusion, may give rise to failures in scene text detection and recognition.

Figure 1. Though the sizes of the characters within the yellow rectangles are small, human can easily discover and localize such text lines.

In this paper, we tackle the problem of scene text detection, which involves discovering and localizing texts from natural scene images. There are mainly two classes of mainstream methods for scene text detection: those based on a sliding window [5, 36, 27] and those based on connected component extraction [7, 25, 9]. The latter category has become the mainstream in the field of scene text detection, since these methods are usually more efficient and relatively insensitive to variations in scale, orientation, font, and language type. In these methods, Maximally Stable Extremal Regions (MSER) [25] and Stroke Width Transform (SWT) [7] are widely adopted as the basic representation due to their efficiency and stability. However, such representations may perform poorly under severe conditions, such as blur, non-uniform illumination, low resolution and disconnected strokes.

To address these issues, we propose in this work a novel representation for localizing text regions. Unlike conventional text detection methods, which typically start from finding character candidates via connected component extraction or sliding-window scanning, the proposed representation directly hunts text lines from natural images.

The mechanism of how mankind identify and recognize text in natural scenes is still not clear at present, but it has been shown that people with normal vision can effortlessly discover text regions without looking into each individual character, even at a glance. For example, we can easily distinguish the text regions in Fig. 1, even though the charac-

*Email: xbai@hust.edu.cn

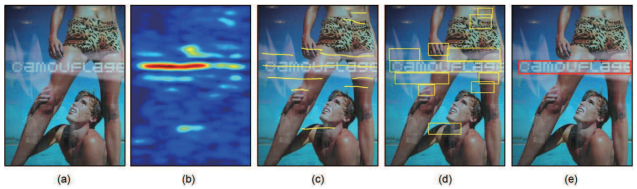

Figure 2. Schematic pipeline of our symmetry-based text-line detection algorithm. (a) Input image; (b) Response map of the symmetry detector; (c) Symmetrical point grouping; (d) Estimated bounding boxes based on the detected symmetrical axes. (e) Detection result after false alarm removal.

ters in those regions are difficult to recognize for us. Different from individual characters, text lines always bear distinctive symmetry and self-similarity properties. The symmetry property of text lines comes from both themselves and their local backgrounds. Taking advantage of this property, we approach the text detection problem from another perspective and propose to seek the symmetrical axes of text lines in natural images, with a symmetry detector.

The pipeline of the proposed symmetry-based text line detection approach is shown in Fig. 2. For each pixel in the image the probability of being on the symmetrical axis of a text line is estimated using a predefined symmetry template (see Fig. 3) at first. Then, text line candidates are formed by grouping the pixels on symmetrical axes and estimating their corresponding bounding boxes. Finally, false positives (non-text candidates) are identified and eliminated with CNN classifiers [16, 15, 37, 11]. To deal with texts of different sizes, the above described procedure is performed at multiple scales. Detection activations from different scales are merged and non-maximum suppression is adopted to remove redundant detections.

The proposed algorithm is able to handle several challenging scenarios (for instance, the characters with dot matrix font as shown in Fig. 2) where MSER and SWT based methods may fail. The experiments on the up-to-date ICDAR benchmarks [32, 13] demonstrate that the proposed algorithm has a broader adaptability (higher detection rate) than conversational methods and outperforms other competing algorithms regarding the final F-measure.

In summary, the core contribution of this work is a symmetry-based text line detector, which directly operates on character group level and achieves state-of-the-art performance on standard benchmarks.

The remainder of this article is organized as follows. In Sec. 2, we briefly review previous works that are related to the proposed algorithm. In Sec. 3, we describe the proposed algorithm in detail, including the symmetry template and feature design, and strategies for text line candidate generation and false alarm removal. The experimental results and discussions are presented in Sec. 4. Finally, conclusion remarks and future works are given in Sec. 5.

2. Related Work

In recent years, the community has witnessed a surge of research efforts on text detection in natural images. A rich body of novel ideas and effective methods have been proposed [7, 25, 47, 41, 26, 48, 11, 10]. For comprehensive surveys, refer to [19, 44, 50]. In this section, we focus on works that are most relevant to the proposed algorithm.

Sliding-window based methods [5, 27] have been very popular in the field of scene text detection. Such methods make use of the texture or local structure property of text and scan all possible positions and scales in the image. The algorithm proposed in this paper also works in a sliding-window fashion. The main difference is that previous methods seek scene text either at a fairly coarse granularity (whole text lines [5]) or at a fine granularity (characters parts or strokes [27]), while our algorithm capture scene text at a moderate granularity (several adjacent characters). The advantages are two-fold: (1) It allows to exploits the symmetry property of character groups, which cannot be excavated at stroke level; (2) It can handle variations within a word or text line, such as mixed case and minor bending.

SWT [7] and MSER [25] are two representative component-based methods for scene text detection, which constitute the basis of a lot of subsequent works [41, 26, 9, 48]. These algorithms assume that characters consist of one or several connected components and utilize this property to seek individual characters or strokes. These algorithms obtained excellent performance on a variety of standard benchmark datasets. However, the weakness of them lies in their inability to handle characters that do not meet the connection assumption, for instance, those composed of broken strokes (see Fig. 2). In contrast, the proposed algorithm abandons the assumption of connection and exploits the vertical symmetry property of character groups, which take advantage of the characteristics of text at a higher level and is applicable to more forms of characters in real-world scenarios, thus leading to higher detection rate.

Over the past few years, there has emerged a new development trend of adopting deep convolutional neural networks [16, 15, 17] for scene text detection. These deep learning based methods [37, 11, 10] usually achieve superior performance over conventional methods [7, 25, 47, 29, 41, 26]. In this work, we also leverage the powerful discrimination ability of deep convolutional neural networks to better eliminate false positives produced in the candidate generation stage, while maintaining a relatively high recall.

The proposed algorithm is inspired by a number of works on symmetry detection [34, 18], which aim at discovering symmetrical structures in generic natural images. In this paper, we make use the symmetry property of text at character group level and draw lessons from such symmetry detection approaches. In this sense, the algorithm proposed in this paper introduces a general technique into a specific domain.

3. Proposed Methodology

In this section, we will describe in detail the proposed algorithm. Generally, this algorithm works in a hypothesis-verification manner. Text proposals are extracted via a symmetry detector 3.1 at first and these proposals are then identified by a verification procedure 3.2, in which non-text proposals are eliminated.

3.1. Symmetry-Based Text Line proposals

At stroke level, the symmetry of text lies in the gradient orientation and magnitude on the stroke boundary. This property has been explored in the SWT work [7]. In this paper we employ the symmetry property at a higher level. The key observation is that a text region usually exhibits high self-similarity to itself and strong contrast to its local background, regarding low-level image cues, such as gradient and texture.

Taking advantage of this property, we propose a novel representation to describe texts in natural scenes. This representation facilitates a text detection approach, which can directly discover text lines from natural images. In this approach, text lines are sought by detecting symmetry axes in the image, followed by bounding box estimation.

3.1.1 Feature Extraction

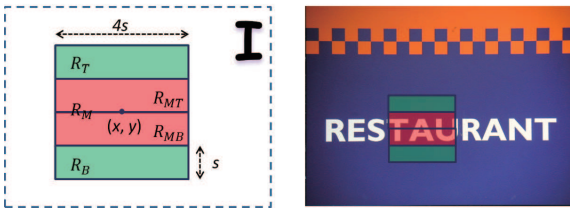

Figure 3. *Left*: Template used to compute the features for symmetry axis detection, which consists of four rectangles with equal size. The height and the width of each rectangle are s and $4s$, respectively. The scale of the template is determined by s. *Right*: The contents within the two middle rectangles are similar to each other but dissimilar to the contents of the top and bottom rectangles. Therefore, the symmetry response on the center line (the adjacent edge of the two middle rectangles) of the text region should be high.

We devise a symmetry template that is suitable for seeking symmetrical structures, following [34]. The template, as illustrated in Fig. 3, consists of four rectangles with equal size $s \times 4s$, denoted by R_T, R_{MT}, R_{MB} and R_B, respectively. The rectangle formed by the two middle rectangles R_{MT} and R_{MB} are denoted by R_M. The height of each rectangle, i.e s, is defined as the scale of the template.

To detect symmetry axes as text line proposals, we employ two types of features: symmetry feature and appearance feature, which capture the intrinsic properties of text. Assume that the template is centered at location (x, y) on the image plane and let $h^c_{x,y}(R_P)(P \in \{T, M, B, MT, MB\})$ denote the histogram of the low-level image cue c in the rectangle R_P. The details for computing the two types of features at location (x, y) are described as follow:

Symmetry Feature. This feature is used to characterize the self-similarity and symmetry property of character groups. Character groups have self-similarity since adjacent characters bear similar color and structure. The self-similarity is defined as the difference between the two middle rectangles in low-level image cues:

$$S^c_{x,y} = \chi^2(h^c_{x,y}(R_{MT}), h^c_{x,y}(R_{MB})), \tag{1}$$

where $\chi^2(\cdot)$ is the χ^2-distance function [31].

Meanwhile, a text region is usually highly dissimilar to its local background. This can be seen as another kind of symmetry, since the contents in the middle rectangles (R_{MT} and R_{MB}) are both different from those in the outer rectangles (R_T and R_B). To measure this dissimilarity, we define the contrast feature as the differences of the low-level image cues within the rectangle pairs:

$$Ct^c_{x,y} = \chi^2(h^c_{x,y}(R_T), h^c_{x,y}(R_{MT})), \tag{2}$$

and

$$Cb^c_{x,y} = \chi^2(h^c_{x,y}(R_B), h^c_{x,y}(R_{MB})). \tag{3}$$

Appearance Feature. The symmetry feature is effective at finding text lines in images, but it also fires on some non-text symmetrical structures. To better distinguish text and non-text, we employ appearance feature, as it has been widely used in previous works [5, 29]. Specifically, the local binary pattern (LBP) feature [28] of the middle rectangle R_M is taken as the appearance feature.

To compute the above described symmetry and appearance features, we adopt four kinds of low-level image cues: brightness, color, texture and gradient. In order to obtain the brightness and color histograms, images are convert to the LAB color space and the pixel values from the brightness channel L^* and the color channels a^* and b^* are quantized into 32 bins respectively. For texture T^* we use the texton implementation proposed in [23]. For gradient G^*, we compute the gradient magnitudes of the pixels and quantize them into a histogram of 16 bins. For the appearance feature, the standard uniform LBP with 59 bins is adopted. All these features are concatenated to represent the pixel at location (x, y), which results in a 74-dimensional feature vector.

3.1.2 Symmetry Axis Detection

For symmetry axis detection, we train a strong classifier to estimate the probability of being on a symmetry axis at each pixel. Random Forest [4] is chosen as the classifier

for its high efficiency and performance. To train the symmetry axis detector, the ground truth rectangles of text lines are required. However, the current text detection benchmarks, such as ICDAR 2011 and 2013, only provide bounding boxes that correspond to parts of text. To produce text line level ground truth, we simply compute the center lines of the bounding boxes.

In the training phase, we sample about 450k positive pixels (pixels whose distances to the ground truth are less than 2 pixels) and 450k negative pixels (the pixels whose distances to the ground truth are larger than 5 pixels) from the training images. For each negative pixel, we compute multiple feature vectors for it, based on templates with multi-scales to form multiple negative training samples. For each positive pixel, as it corresponds to an annotated bounding box, we compute one feature vector for it, based on the template with the size equals to half of the height of the annotated bounding box. The training samples are fed into the tree and split recursively into different leaf nodes. The splitting is determined by the feature selection mechanism, which is important for training a "good" tree. As the dimensions of the proposed two types of features are not equal, we assign different selection weights to them to avoid unbalance selection results. Intuitively, the weights should be in inverse proportion to the feature dimensions.

In the testing phase, as neither the locations nor the scales of text lines are known, we visit each pixel in the image and compute multiple feature vectors for it (Fig. 4 (b)). The learned Random Forest classifier predicts the probability of the image pixel being on a symmetry axis or not, given the feature vector computed on it. Since feature vectors of multiple scales are computed for each image pixel, multiple symmetry probability maps (Fig. 4 (c)) are generated for a testing image.

3.1.3 Proposals Generation

After the symmetry detection stage, we obtain multiple symmetry probability maps (Fig. 4 (c)) for a testing image. Based on these maps, we can aggregate axis pixels to form text line proposals (Fig. 4 (d)). At first, we directly group pixels whose distance is smaller than 3 pixels to produce symmetry axis fragments. Then, we adopt a graph model to further aggregate the fragments. Each fragment is represented as a vertex in the graph and an edge is constructed between two fragments if they satisfy the following geometric constraints:

Angular Difference Constraint. The direction of fragments who belongs to the same text region is usually closed. Based on this observation, we define the angular difference between two fragments as:

$$\Phi(A,B) = |\phi(A) - \phi(B)|, \phi(A), \phi(B) \in (-\frac{\pi}{2}, \frac{\pi}{2}], \quad (4)$$

where A and B represent two fragments, and ϕ represents the direction angular of a fragment. In practice, we use the average direction angles of each pixels to estimate it. If $\Phi(A,B) > \frac{\pi}{16}$, A and B are labeled as unconnected.

Distance Constraint. If two fragments are far away from each other, they shouldn't be grouped together. We define the distance between two fragments as:

$$D(A,B) = \min(\|p-q\|), p \in A, q \in B, \quad (5)$$

where p and q are two points in fragment A and B respectively. $\|p-q\|$ indicates the distance between p and q. The height of a fragment H is defined as the scale of the corresponding template. If $D(A,B) > \max(H(A), H(B))$, A and B are labeled as unconnected.

Text line proposals are formed by simply seeking connected subsets in the graph. Each connected subset corresponds to a text line proposal. The bounding box of each proposal (Fig. 4 (e)) is calculated as follows: The width is determined by the horizontal axis coordinates of the axis pixels belong to the proposal and the height is the scale of the corresponding template.

To handle text lines of different sizes, we extract proposals at multiple scales and merge all text line proposals from all different scales (Fig. 4 (f)).

In this paper, we only consider horizontal or near-horizontal texts. However, the strategies presented are actually general and thus are readily applicable to texts of different orientations.

3.2. False Positive Removal

Figure 5. False positive removal. For simplicity, we only show the procedure of false positive removal at a single scale.

A portion of the text candidates generated in the proposal generation stage are non-text (Fig. 5 (a)). The purpose of false positive removal is to identify and eliminate such non-text candidates. Inspired by the deep learning methods of Jaderberg *et al.* [11] and Huang *et al.* [10], we also adopt CNN classifiers for false positive removal. Different from [11, 10], which only used CNN classifier for patch or character level discrimination, we train two classifiers that work at character level and text region level, respectively.

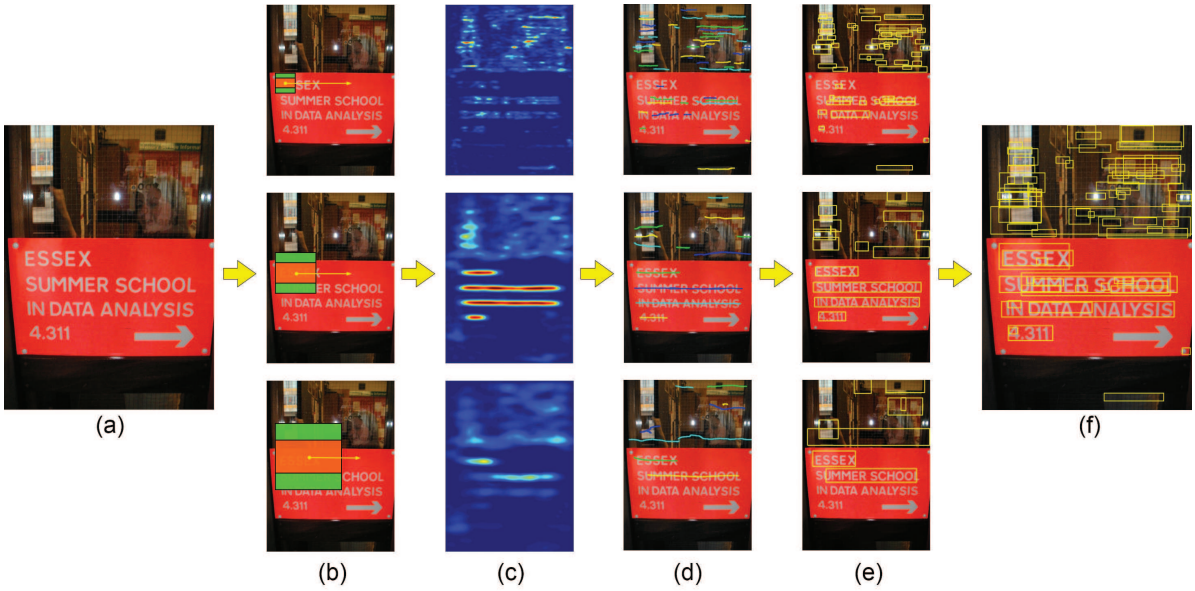

Figure 4. Procedure of text line proposal generation. (a) Input image. (b) Feature extraction at multiple scales. (c) Symmetry probability maps. (d) Axes sought in the symmetry probability maps. (e) Bounding box estimation. (f) Proposals from different scales

The training examples for the character level classifier is from [11], which is publicly available[1]. The training examples for the text region classifier are harvested from several text related datasets (ICDAR 2011 [32], SVT [36] and IIIT 5K-Word [24]) and generic image datasets (PASCAL VOC [8] and BSDS500 [2]).

In the false positive removal procedure, the character level classifier is first applied to the text line proposals, which are later partitioned into "words" (Fig. 5 (b)), using the method proposed in [11]. The text region classifier is then applied to these "word" regions and those with low scores are discarded. After false positive removal, the remained proposals are considered as the final detection results (Fig. 5 (c)).

4. Experiments

We implemented the proposed algorithm in Matlab (with C/C++ mex functions) and evaluated it on the latest ICDAR datasets: ICDAR 2011 [32] and ICDAR 2013 [13], as well as the SWT dataset [7]. The proposed algorithm is compared with other methods for scene text detection, including the top performers [10, 11] on these two benchmarks.

All the experiments were carried out on a regular computer (2.0GHz 8-core CPU, 64G RAM and Windows 64-bit OS). For the Random Forest classifier, 50 trees were used and the maximum depth of the trees was set to 100. At runtime, all the testing images were rescaled to a standard height of 800 pixels, with aspect ratio kept unchanged. The symmetry detector ran at 24 different scales and the scales of the symmetry templates (s) range from 2 to 256 pixels.

4.1. Datasets and Evaluation Protocol

In this paper, we evaluated the proposed algorithm on standard datasets and followed the standard evaluation protocols in this field.

ICADR 2011. The datasets used in ICDAR 2011[2] and 2013[3] are inherited from the benchmark used in the previous ICDAR competitions [21, 20], but have undergone extension and modification, since there are some problems with the previous dataset (e.g., imprecise bounding boxes and inconsistent definitions of "word"). The ICDAR 2011 dataset includes 299 training images and 255 testing images.

ICDAR 2013. The ICDAR 2013 dataset is a subset of ICDAR 2011. Several images that duplicated over training and testing sets of the ICDAR 2011 dataset is removed. In addition, a small port of the ground truth annotations has been revised. There are 229 images for training and 233 images for testing.

SWT Dataset. The SWT dataset, which is introduced by [7], consists of 307 color images with sizes ranging from 1024×368 to 1024×768. This dataset is more challenging than ICDAR datasets, because of the smaller texts, repeating patterns, various plants, etc.

Evaluation Protocol. In scene text detection, there are three important metrics in performance assessment: precision, recall and F-measure. Precision measures the ratio between true positives and all detections, while recall mea-

[1] https://bitbucket.org/jaderberg/eccv2014_textspotting/overview

[2] http://robustreading.opendfki.de/wiki/SceneText

[3] http://dag.cvc.uab.es/icdar2013competition/

sures the ratio true positives and all true texts that should be detected. F-measure, as an overall, single indicator of algorithm performance, is the harmonic mean of precision and recall.

The evaluation method used in ICDAR 2011 was originally proposed by Wolf *et al.* [38]. The protocol of Wolf *et al.* [38] considers three matching cases: one-to-one, one-to-many and many-to-many. Precision and recall are defined as follows:

$$precision(G, D, t_r, t_p) = \frac{\sum_j Match_D(D_j, G, t_r, t_p)}{|D|}, \quad (6)$$

$$recall(G, D, t_r, t_p) = \frac{\sum_i Match_G(G_i, D, t_r, t_p)}{|G|}. \quad (7)$$

G and D represent ground truth rectangle set and detection rectangle set, respectively. $t_r \in [0, 1]$ is the constraint on area recall and $t_p \in [0, 1]$ is the constraint on area precision. The typical values of t_r and t_p are 0.8 and 0.4, respectively. $Match_D$ and $Match_G$ are functions which take different types of matches into consideration. The evaluation protocol for ICDAR 2013 is similar with that of ICDAR 2011, expect for a number of heuristics cues. For more details, please refer to [13].

The evaluation protocol for SWT dataset was proposed by [7]. Because this dataset does not clearly seperate training set and testing set, we followed [30] to use all images as testing set.

4.2. Experimental Results and Discussions

4.2.1 Text Detection Performance

Fig. 6 illustrates several detection examples of the proposed algorithm on the ICDAR 2011 dataset. As can be seen, the algorithm works fairly well under various challenging cases, such as dot matrix fonts (Fig. 6 (a) and (j)), low contrast (Fig. 6 (b) and (i)), low resolution (Fig. 6 (g)), non-uniform illumination (Fig. 6 (f)), inner texture (Fig. 6 (h)), and broken strokes (Fig. 6 (c)). Note that for these challenging cases, conventional methods (such as SWT and MSER) usually produce unsatisfactory results.

The proposed algorithm might chop letters in some cases(see the last image in Fig. 7), due to mixed case or special alignment of certain characters. But in most cases the estimation error of final bounding boxes is within acceptable range (Fig. 6 (d) and (k)), because: (1) We generate bounding boxes at different scales independently. The most proper scale will be selected in the last stage. (2) We did not solely rely on symmetry feature. Appearance feature and false positive removal (CNN based verification) also play an important role in rejecting improper bounding boxes.

Table 1. Performances of different algorithms evaluated on the ICDAR 2011 dataset.

Algorithm	Precision	Recall	F-measure
Proposed	0.84	**0.76**	**0.80**
Huang *et al.* [10]	**0.88**	0.71	0.78
Yin *et al.* [48]	0.863	0.683	0.762
Koo *et al.* [14]	0.814	0.687	0.745
Yao *et al.* [39]	0.822	0.657	0.730
Huang *et al.* [9]	0.82	0.75	0.73
Neumann *et al.* [27]	0.793	0.664	0.723
Shi *et al.* [33]	0.833	0.631	0.718
Kim *et al.* [32]	0.830	0.625	0.713
Neumann *et al.* [26]	0.731	0.647	0.687
Yi *et al.* [45]	0.672	0.581	0.623
Yang *et al.* [32]	0.670	0.577	0.620
Neumann *et al.* [32]	0.689	0.525	0.596
Shao *et al.* [32]	0.635	0.535	0.581

Table 2. Performances of different algorithms evaluated on the ICDAR 2013 dataset.

Algorithm	Precision	Recall	F-measure
Proposed	**0.88**	**0.74**	**0.80**
iwrr2014 [49]	0.86	0.70	0.77
USTB TexStar [48]	**0.88**	0.66	0.76
Text Spotter [26]	**0.88**	0.65	0.74
CASIA_NLPR [1]	0.79	0.68	0.73
Text_Detector_CASIA [33]	0.85	0.63	0.72
I2R_NUS_FAR [1]	0.75	0.69	0.72
I2R_NUS [1]	0.73	0.66	0.69
TH-TextLoc [1]	70	0.65	0.67

The performances of the proposed algorithm as well as other methods on the ICDAR 2011 dataset are shown in Tab. 1. The proposed algorithm achieves an F-measure of 0.80, outperforming other methods. Compared to the closest competitor [10], the recall of our algorithm (0.76) is much higher that of [10] (0.71). This confirms the effectiveness of our algorithm, especially its advantage in handling various challenging scenarios.

The performances of the proposed algorithm as well as other methods on the ICDAR 2013 dataset are depicted in Tab. 2. The proposed algorithm obtains 0.88, 0.74, 0.80 in precision, recall and F-measure, respectively. As on ICDAR 2011, the proposed method achieves state-of-the-art performance on this dataset.

The performance of the proposed algorithm as well as other methods on the SWT dataset are depicted in Tab. 3. The proposed algorithm obtains 0.68, 0.53, 0.60 in precision, recall and F-measure, outperforming other competitors. This demonstrates the advantages of the proposed algorithm.

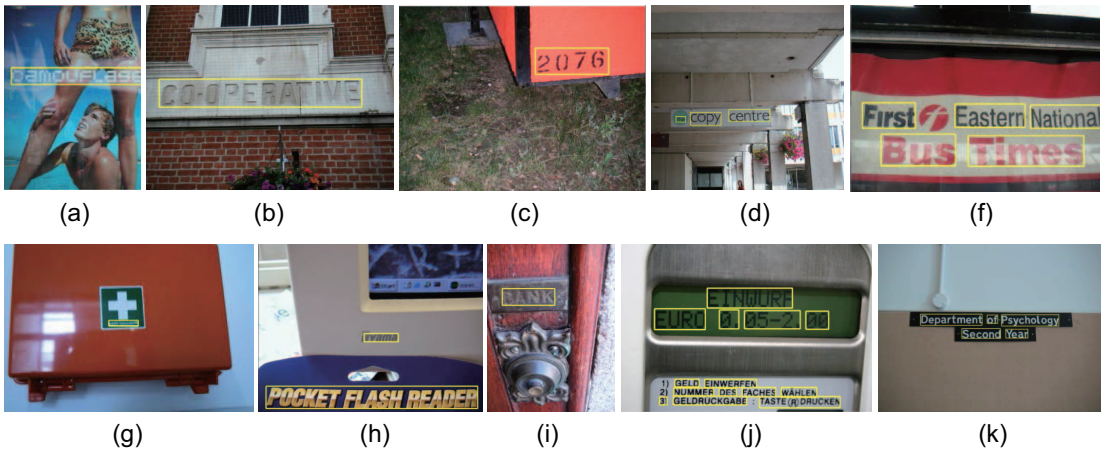

Figure 6. Detection examples of the proposed method.

Table 3. Performances of different algorithms evaluated on the SWT dataset.

Algorithm	Precision	Recall	F-measure
Proposed	**0.68**	**0.53**	**0.60**
Du *et al.* [6]	0.66	0.51	0.58
Phan *et al.* [30]	0.50	0.51	0.51
Yi *et al.* [46]	0.42	0.60	0.49
Mao *et al.* [22]	0.58	0.41	0.48
Epshtein *et al.* [7]	0.54	0.42	0.47

4.2.2 Character Detection Rate

To demonstrate the effectiveness and robustness of the proposed symmetry-based representation, we compared it with MSER [25, 26] with respect to the text candidate extraction ability. This ability is measured using the character detection rate on the training set of the ICDAR 2013 dataset. We chose it because it provides an detailed annotations for single characters. It includes 229 images and 4786 characters.

Since our symmetry-based representation works at characters group level while MSER extracts characters or character parts, their character detection rates cannot be compared directly. To make fair comparison possible, we adopted the following definition of character detection rate:

$$R = \frac{\sum_{i=1}^{N}\sum_{j=1}^{|G_i|}\max_{k=1}^{|D_i|} m(G_i^{(j)}, D_i^{(k)})}{\sum_{i=1}^{N}|G_i|}, \qquad (8)$$

where N is the total number of images in the dataset. $G_i^{(j)}$ and $D_i^{(k)}$ are the jth ground truth rectangle and kth detection rectangle in image i. $m(G_i^{(j)}, D_i^{(k)})$ is the match score between the jth ground truth rectangle and kth detection rectangle $D_i^{(k)}$. The match score is defined as:

Table 4. Detection rates of different methods on the ICDAR 2013 dataset.

Algorithm	Detection Rate	Proposal Number
Proposed	**0.977**	**1310**
MSER (Gray+LUV)	0.964	8415

$$m(G_i^{(j)}, D_i^{(k)}) = \begin{cases} 1 & \frac{|G_i^{(j)} \cap D_i^{(k)}|}{|G_i^{(j)}|} \geq 0.8 \text{ and} \\ & \frac{\max(h(G_i^{(j)}), h(D_i^{(k)}))}{\min(h(G_i^{(j)}), h(D_i^{(k)}))} \leq 1.5, \\ 0 & \text{otherwise} \end{cases} \qquad (9)$$

where h is the height of a rectangle.

As shown in Tab. 4, MSER ran in 4 channels(gray and L,U,V, respectively) and detected about 96.4% of the characters, and the average number of proposals per image is 8415. The proposed method detected 97.7% of the characters and only produced 1310 proposals on average for each image. This demonstrates that the adaptability of the proposed representation is stronger than that of MSER. Upon investigation, we found that our method can handle challenging cases (for example, characters with broken strokes, dot matrix fonts, low resolution or partial occlusion, as shown in Fig. 6) where MSER failed.

In this paper, we mainly exploited the symmetry property of text at character group level, which only reflects a portion of the characteristics of text. Obviously, to build more effective and reliable systems for text detection, one should take full advantage of the characteristics of text. We believe higher performance could be attained, if the proposed representation is integrated with conventional representations,

Table 5. Contributions of different types of features.

Feature	Precision	Recall	F-measure
symmetry	0.80	0.65	0.72
appearance	0.79	0.57	0.66
symmetry+appearance	0.84	0.76	0.80

such as SWT [7] and MSER [25].

4.2.3 Applicability to Texts in Different Languages

Figure 7. Detection examples on texts in different languages.

Fig. 7 depicts several examples of the proposed text detection algorithm on texts in different languages. As can be seen, even though the detector was only trained on English texts, it can be easily applied to texts of other languages. This further confirms the applicability of the proposed symmetry-based text detector.

4.2.4 Contributions of Different Types of Features

In the proposed symmetry detector, we employed two types of features: symmetry feature and appearance feature. To assess the contributions of these features, we conducted a experiment on the ICDAR 2011 dataset with different settings: symmetry feature, appearance feature and their combination (symmetry+appearance). The performances of these three settings are shown in Tab. 5. As can be seen, these two types of features already achieve promising results when used in isolation. The symmetry feature works better than the appearance feature. These two types of features are indeed complementary. Their combination leads to a significant boost in F-measure (from 0.72 to 0.80).

4.3. Limitations of Proposed Algorithm

Though the proposed algorithm is capable of dealing with various challenging scenarios and achieves excellent performance on standard benchmarks, it is far from perfect. It may give false positives or miss true texts in certain situations. Fig. 8 depicts some failure cases of the proposed method. The algorithm failed to detect characters with extremely low contrast (Fig. 8 (c)) or strong reflect light (Fig. 8 (a) and (b)), or missed single character (Fig. 8 (f)), or tremendous size difference between characters(Fig. 8 (d)). Note that the characters in the bottom of Fig. 8 (e) were successfully detected by the proposed algorithm, but these characters are not included in the ground truth, so they were regarded as false alarms. This indicates the broad adaptability of our algorithm.

Figure 8. Failure cases of the proposed method.

Another shortcoming of the proposed algorithm is that the processing speed is relatively slow, since it is partially implemented in Matlab and should scan tens of scales for a given image. Without optimization and parallelization, the average runtime on the ICDAR 2011 dataset [32] is about 30 seconds for each image.

In summary, there is still great room for improvement in both accuracy and efficiency, as relevant real-world products and services pose high requirements for effective and efficient textual information extraction from natural scenes.

5. Conclusion

In this paper, we have presented a novel algorithm for text detection in natural scenes. Different from traditional methods, which focus on hunting characters or strokes via connected component extraction [7, 25] or sliding window scanning [36, 27], this algorithm makes use of the symmetry and self-similarity properties of character groups and is able to directly discover text lines from natural images. The core contribution of the proposed algorithm is a novel symmetry-based representation, which can detect challenging characters that are usually missed by conventional component extractors, like SWT [7] and MSER [25]. The experiments on the latest ICDAR datasets [32, 13] demonstrate that the proposed algorithm outperforms other competing methods in the literature.

One major drawback of the proposed algorithm lies in its low efficiency. We will investigate better strategies, including multi-thread and GPU techniques, to speed up the procedure. Another direction worthy of exploring is to design symmetry templates that can handle texts of varying orientations. Moreover, we could apply the proposed idea to other detection problems, such as human detection [40].

Acknowledgement

This work was primarily supported by National Natural Science Foundation of China (NSFC) (No. 61222308), and in part by NSFC (No. 61303095), CCF-TencentRAGR (No. 20140116) and Program for New Century Excellent Talents in University (No. NCET-12-0217).

References

[1] ICDAR 2013 robust reading competition challenge 2 results. http://dag.cvc.uab.es/icdar2013competition, 2014. [Online; accessed 11-November-2014].

[2] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE Trans. PAMI*, 33(5):898–916, 2011.

[3] A. Bissacco, M. Cummins, Y. Netzer, and H. Neven. PhotoOCR: Reading text in uncontrolled conditions. In *Proc. of ICCV*, 2013.

[4] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[5] X. Chen and A. Yuille. Detecting and reading text in natural scenes. In *Proc. of CVPR*, 2004.

[6] Y. Du, G. Duan, and H. Ai. Context-based text detection in natural scenes. In *Image Processing (ICIP), 2012 19th IEEE International Conference on*, pages 1857–1860. IEEE, 2012.

[7] B. Epshtein, E. Ofek, and Y. Wexler. Detecting text in natural scenes with stroke width transform. In *Proc. of CVPR*, 2010.

[8] M. Everingham, L. V. Gool, C. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2011 (VOC2011) Results. http://www.pascal-network.org/challenges/VOC/voc2011/workshop/index.html.

[9] W. Huang, Z. Lin, J. Yang, and J. Wang. Text localization in natural images using stroke feature transform and text covariance descriptors. In *Proc. of ICCV*, 2013.

[10] W. Huang, Y. Qiao, and X. Tang. Robust scene text detection with convolution neural network induced mser trees. In *Proc. of ECCV*, 2014.

[11] M. Jaderberg, A. Vedaldi, and A. Zisserman. Deep features for text spotting. In *Proc. of ECCV*, 2014.

[12] L. Kang, Y. Li, and D. Doermann. Orientation robust text line detection in natural images. In *Proc. of CVPR*, 2014.

[13] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L. G. i Bigorda, S. R. Mestre, J. Mas, D. F. Mota, J. A. Almazan, and L. P. de las Heras. ICDAR 2013 robust reading competition. In *Proc. of ICDAR*, 2013.

[14] H. Koo and D. H. Kim. Scene text detection via connected component clustering and nontext filtering. *IEEE Trans. on Image Processing*, 22(6):2296–2305, 2013.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proc. of NIPS*, 2012.

[16] Y. LeCun, B. Boser, J. S. Denker, R. E. Howard, W. Habbard, L. D. Jackel, and D. Henderson. Handwritten digit recognition with a back-propagation network. In *Proc. of NIPS*, 1990.

[17] C. Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu. Deeply-supervised nets. In *Proc. of AISTATS*, 2015.

[18] T. Lee, S. Fidler, and S. Dickinson. Detecting curved symmetric parts using a deformable disc model. In *Proc. of ICCV*, 2013.

[19] J. Liang, D. Doermann, and H. Li. Camera-based analysis of text and documents: a survey. *IJDAR*, 7(2):84–104, 2005.

[20] S. M. Lucas. ICDAR 2005 text locating competition results. In *Proc. of ICDAR*, 2005.

[21] S. M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, and R. Young. ICDAR 2003 robust reading competitions. In *Proc. of ICDAR*, 2003.

[22] J. Mao, H. Li, W. Zhou, S. Yan, and Q. Tian. Scale based region growing for scene text detection. In *Proceedings of the 21st ACM international conference on Multimedia*, pages 1007–1016. ACM, 2013.

[23] D. R. Martin, C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(5):530–549, 2004.

[24] A. Mishra, K. Alahari, and C. V. Jawahar. Scene text recognition using higher order language priors. In *Proc. of BMVC*, 2012.

[25] L. Neumann and J. Matas. A method for text localization and recognition in real-world images. In *Proc. of ACCV*, 2010.

[26] L. Neumann and J. Matas. Real-time scene text localization and recognition. In *Proc. of CVPR*, 2012.

[27] L. Neumann and J. Matas. Scene text localization and recognition with oriented stroke detection. In *Proc. of ICCV*, 2013.

[28] T. Ojala, M. Pietikäinen, and T. Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(7):971–987, 2002.

[29] Y. Pan, X. Hou, and C. Liu. A hybrid approach to detect and localize texts in natural scene images. *IEEE Trans. Image Processing*, 20(3):800–813, 2011.

[30] T. Q. Phan, P. Shivakumara, and C. L. Tan. Detecting text in the real world. In *Proceedings of the 20th ACM international conference on Multimedia*, pages 765–768. ACM, 2012.

[31] Y. Rubner, J. Puzicha, C. Tomasi, and J. M. Buhmann. Empirical evaluation of dissimilarity measures for color and texture. *Computer Vision and Image Understanding*, 84(1):25–43, 2001.

[32] A. Shahab, F. Shafait, and A. Dengel. ICDAR 2011 robust reading competition challenge 2: Reading text in scene images. In *Proc. of ICDAR*, 2011.

[33] C. Shi, C. Wang, B. Xiao, Y. Zhang, and S. Gao. Scene text detection using graph model built upon maximally stable extremal regions. *Pattern Recognition Letters*, 34(2):107–116, 2013.

[34] S. Tsogkas and I. Kokkinos. Learning-based symmetry detection in natural images. In *Proc. of ECCV*, 2012.

[35] K. Wang, B. Babenko, and S. Belongie. End-to-end scene text recognition. In *Proc. of ICCV*, 2011.

[36] K. Wang and S. Belongie. Word spotting in the wild. In *Proc. of ECCV*, 2010.

[37] T. Wang, D. J. Wu, A. Coates, and A. Y. Ng. End-to-end text recognition with convolutional neural networks. In *Proc. of ICPR*, 2012.

[38] C. Wolf and J. M. Jolion. Object count/area graphs for the evaluation of object detection and segmentation algorithms. *IJDAR*, 8(4):280–296, 2006.

[39] C. Yao, X. Bai, and W. Liu. A unified framework for multi-oriented text detection and recognition. *IEEE Trans. Image Processing*, 23(11):4737–4749, 2014.

[40] C. Yao, X. Bai, W. Liu, and L. J. Latecki. Human detection using learned part alphabet and pose dictionary. In *Proc. of ECCV*, 2014.

[41] C. Yao, X. Bai, W. Liu, Y. Ma, and Z. Tu. Detecting texts of arbitrary orientations in natural images. In *Proc. of CVPR*, 2012.

[42] C. Yao, X. Bai, B. Shi, and W. Liu. Strokelets: A learned multi-scale representation for scene text recognition. In *Proc. of CVPR*, 2014.

[43] C. Yao, X. Zhang, X. Bai, W. Liu, Y. Ma, and Z. Tu. Rotation-invariant features for multi-oriented text detection in natural images. *PLoS One*, 8(8), 2013.

[44] Q. Ye and D. Doermann. Text detection and recognition in imagery: A survey. *IEEE Trans. on PAMI*, (99), 2014.

[45] C. Yi and Y. Tian. Text detection in natural scene images by stroke gabor words. In *Proc. of ICDAR*, 2011.

[46] C. Yi and Y. Tian. Text detection in natural scene images by stroke gabor words. In *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, pages 177–181. IEEE, 2011.

[47] C. Yi and Y. Tian. Text string detection from natural scenes by structure-based partition and grouping. *IEEE Trans. Image Processing*, 20(9):2594–2605, 2011.

[48] X. C. Yin, X. Yin, K. Huang, and H. Hao. Robust text detection in natural scene images. *IEEE Trans. on PAMI*, 36(5):970–983, 2014.

[49] A. Zamberletti, L. Noce, and I. Gallo. Text localization based on fast feature pyramids and multi-resolution maximally stable extremal regions. In *Proc. of ACCV workshop*, 2014.

[50] Y. Zhu, C. Yao, and X. Bai. Scene text detection and recognition: Recent advances and future trends. *Frontiers of Computer Science*, 2015.