Multi-Oriented Text Detection with Fully Convolutional Networks

Zheng Zhang[1*] Chengquan Zhang[1*] Wei Shen[2] Cong Yao[1] Wenyu Liu[1] Xiang Bai[1†]
[1] School of Electronic Information and Communications, Huazhong University of Science and Technology
[2] Key Laboratory of Specialty Fiber Optics and Optical Access Networks, Shanghai University
macaroniz1990@gmail.com, zchengquan@gmail.com, xbai@hust.edu.cn

Abstract

In this paper, we propose a novel approach for text detection in natural images. Both local and global cues are taken into account for localizing text lines in a coarse-to-fine procedure. First, a Fully Convolutional Network (FCN) model is trained to predict the salient map of text regions in a holistic manner. Then, text line hypotheses are estimated by combining the salient map and character components. Finally, another FCN classifier is used to predict the centroid of each character, in order to remove the false hypotheses. The framework is general for handling text in multiple orientations, languages and fonts. The proposed method consistently achieves the state-of-the-art performance on three text detection benchmarks: MSRA-TD500, ICDAR2015 and ICDAR2013.

1. Introduction

Driven by the increasing demands for many computer vision tasks, reading text in the wild (from scene images) has become an active direction in this community. Though extensively studied in recent years, text spotting under uncontrolled environments is still quite challenging. Especially, detecting text lines with arbitrary orientations is an extremely difficult task, as it takes much more hypotheses into account, which drastically enlarges the searching space. Most existing approaches are successfully designed for detecting horizontal or near-horizontal text [3, 4, 15, 17, 2, 8, 6, 35, 23, 20, 27]. However, there is still a large gap when applying them to multi-oriented text, which has been verified by the low accuracies reported in the recent ICDAR2015 competition for text detection [10].

Text, which can be treated as sequence-like objects with unconstrained lengths, possesses very distinctive appearance and shape compared to generic objects. Consequently, the detection methods in scene images based on sliding windows [8, 25, 3, 24, 17] and connected component [16, 6, 4, 32, 28] have become mainstream in this specific domain. In particular, the component-based methods utilizing Maximally Stable Extremal Regions (MSER) [15] as the basic representations achieved the state-of-the-art performance on ICDAR2013 and ICDAR2015 competitions [11, 10]. Recently, [6] utilized a convolution neural network to learn highly robust representations of character components. Usually, the component grouping algorithms including clustering algorithms or some heuristic rules are essential for localizing text at a word or line level. As an unconventional approach, [35] directly hits text lines from cluttered images, benefiting from symmetry and self-similarity properties of them. Therefore, it seems that both local (character components) and global (text regions) information are very helpful for text detection.

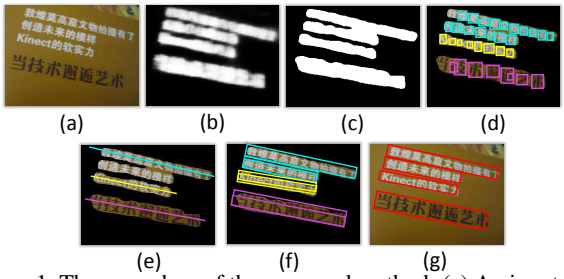


Figure 1. The procedure of the proposed method. (a) An input image; (b) The salient map of the text regions predicted by the Text-Block FCN; (c) Text block generation; (d) Candidate character component extraction; (e) Orientation estimation by component projection; (f) Text line candidates extraction; (g) The detection results of the proposed method.

In this paper, an unconventional detection framework for multi-oriented text is proposed. The basic idea is to integrate local and global cues of text blocks with a coarse-to-fine strategy. At the coarse level, a pixel-wise text/non-text salient map is efficiently generated by utilizing a Fully Convolutional Network (FCN) [12]. We show that the salient map provides a powerful guidance for estimating orientations and generating candidate bounding boxes of text lines, while combining it with local character components. More

*Authors contributed equally
†Corresponding author

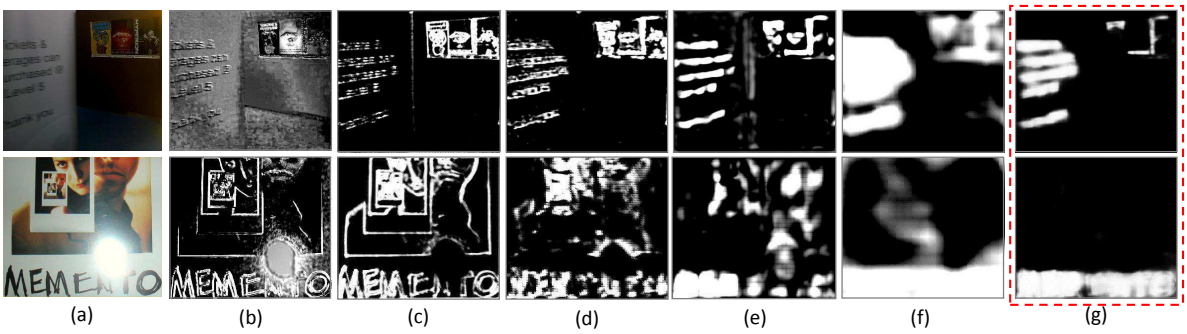


Figure 2. The illustration of feature maps generated by the Text-Block FCN. (a) Input images; (b)∼(f) The feature maps from stage1∼stage5. Lower level stages capture more local structures, and higher level stages capture more global information; (g) The final salient maps.

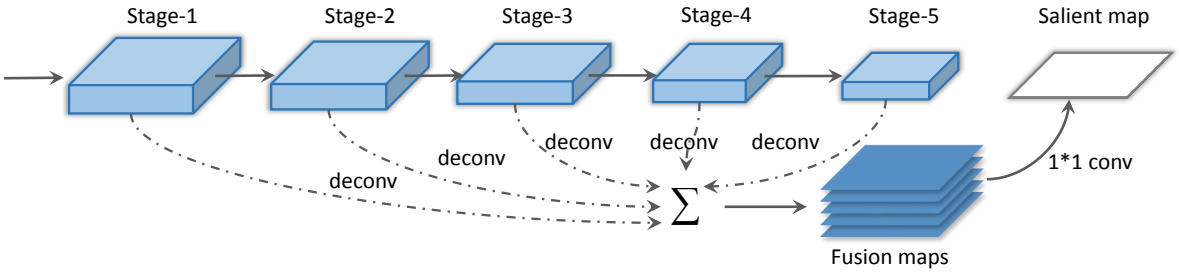


Figure 3. The network architecture of the Text-Block FCN whose 5 convolutional stages are inherited from VGG 16-layer model. For each stage, a deconvolutional layer (equals to a 1×1 convolutional layer and a upsampling layer) is connected. All the feature maps are concatenated with a 1×1 convolutional layer and a sigmoid layer.

specifically, the pipeline of the proposed detection framework is shown in Fig. 1. First, a salient map is generated and segmented into several candidate text blocks. Second, character components are extracted from the text blocks. Third, projections of the character components are used for estimating the orientation. Then, based on the estimated orientation, all candidate bounding boxes of text lines are constructed by integrating cues from the components and the text blocks. Finally, detection results are obtained by removing false candidates through a filtering algorithm.

Our contributions are in three folds: First, we present a novel way for computing text salient map, through learning a strong text labeling model with FCN. The text labeling model is trained and tested in a holistic manner, highly stable to the large scale and orientation variations of scene text, and quite efficient for localizing text blocks at the coarse level. In addition, it is also applicable to multi-script text. Second, an efficient method for extracting bounding boxes of text line candidates in multiple orientations is presented. We show that the local (character components) and the global (text blocks from the salient map) cues are both helpful and complementary to each other. Our third contribution is to propose a novel method for filtering false candidates. We train an efficient model (another FCN) to predict character centroids within the text line candidates. We show that the predicted character centroids provide accurate positions of each character, which are effective features for removing the false candidates. The proposed detection framework achieves the state-of-the-art performance on both horizontal and multi-oriented scene text detection benchmarks.

The remainder of this paper is organized as follows: In Sec. 2, we briefly review the previously related work. In Sec. 3, we describe the proposed method in detail, including text block detection, strategies for multi-oriented text line candidate generation, and false alarm removal. Experimental results are presented in Sec. 4. Finally, conclusion remarks and future work are given in Sec. 5.

2. Related Work

Text detection in natural images has received much attention from the communities of computer vision and document analysis. However, most text detection methods focus on detecting horizontal or near-horizontal text mainly in two ways: 1) localizing the bounding boxes of words [4, 3, 17, 15, 18, 33, 5, 6], 2) combining detection and recognition procedures into an end-to-end text recognition method [8, 28]. Comprehensive surveys for scene text detection and recognition can be referred to [30, 36].

In this section, we focus on the most relevant works that are presented for multi-oriented text detection. Multi-

oriented text detection in the wild is first studied by [31, 29]. Their detection pipelines are similar to the traditional methods based on connected component extraction, integrating orientation estimation of each character and text line. [9] treated each MSER component as a vertex in a graph, then text detection is transferred into a graph partitioning problem. [32] proposed a multi-stage clustering algorithm for grouping MSER components to detect multi-oriented text. [28] proposed an end-to-end system based on SWT [4] for multi-oriented text. Recently, a challenging benchmark for multi-oriented text detection has been released for the ICDAR2015 text detection competition, and many researchers have reported their results on it.

In addition, it is worth mentioning that both of the recent approaches [25, 8, 6] and our method, which used the deep convolutional neural network, have achieved superior performance over conventional approaches in several aspects: 1) learn a more robust component representation by pixel labeling with CNN [8]; 2) leverage the powerful discrimination ability of CNN for better eliminating false positives [6, 35]; 3) learn a strong character/word recognizer with CNN for end-to-end text detection [25, 7]. However, these methods only focus on horizontal text detection.

3. Proposed Methodology

In this section, we describe the proposed method in detail. First, text blocks are detected via a fully convolutional network (named Text-Block FCN). Then, multi-oriented text line candidates are extracted from these text blocks by taking the local information (MSER components) into account. Finally, false text line candidates are eliminated by the character centroid information. The character centroid information is provided by a smaller fully convolutional network (named Character-Centroid FCN).

3.1. Text Block Detection

In the past few years, most of the leading methods in scene text detection are based on detecting characters. In early practice [16, 18, 29], a large number of manually designed features are used to identify characters with strong classifiers. Recently, some works [6, 8] have achieved great performance, adopting CNN as a character detector. However, even the state-of-the-art character detector [8] still performs poorly at complicated background (Fig. 4 (b)). The performance of the character detector is limited due to three aspects: firstly, characters are susceptible to several conditions, such as blur, non-uniform illumination, low resolution, disconnected stroke, etc.; secondly, a great quantity of elements in the background are similar in appearance to characters, making them extremely hard to distinguish; thirdly, the variation of the character itself, such as fonts, colors, languages, etc., increases the learning difficulty for classifiers. By comparison, text blocks possess more distinguishable and stable properties. Both local and global appearances of text block are useful cues for distinguishing between text and non-text regions (Fig. 4 (c)).

Fully convolutional network (FCN), a deep convolutional neural network proposed recently, has achieved great performance on pixel level recognition tasks, such as object segmentation [12] and edge detection [26]. This kind of network is very suitable for detecting text blocks, owing to several advantages: 1) It considers both local and global context information at the same time.; 2) It is trained in an end-to-end manner; 3) Benefiting from the removal of fully connected layers, FCN is efficient in pixel labeling. In this section, we learn a FCN model, named Text-Block FCN, to label salient regions of text blocks in a holistic way.

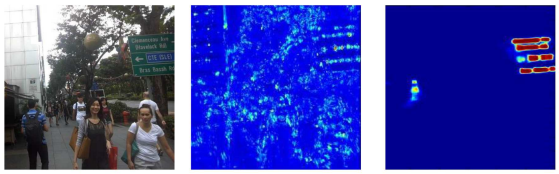


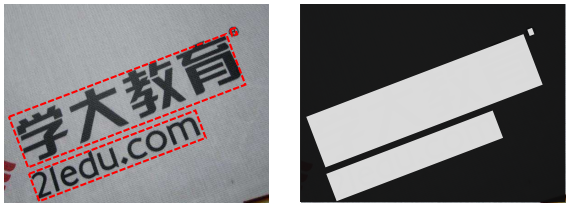
Figure 4. The results of a character detector and our method. (a) An input image; (b) The character response map, which is generated by the state-of-the-art method [8]; (c) The salient map of text regions, which is generated by the Text-Block FCN.

Text-Block FCN We convert the VGG 16-layer net [22] into our text block detection model that is illustrated in Fig. 3. The first 5 convolutional stages are derived from the VGG 16-layer net. The receptive field sizes of the convolutional stages are variable, contributing to that different stages can capture context information with different sizes. Each convolutional stage is followed by a deconvolutinal layer (equals to a 1×1 convolutional layer and a upsampling layer) to generate feature maps of the same size. The discriminative and hierarchical fusion maps are then the concatenation in depth of these upsampled maps. Finally, the fully-connected layers are replaced with a 1×1 convolutional layer and a sigmoid layer to efficiently make the pixel-level prediction.

In the training phase, pixels within the bounding box of each text line or word are considered as the positive region for the following reasons: firstly, the regions between adjacent characters are distinct in contrast to other non-text regions; secondly, the global structure of text can be incorporated into the model; thirdly, bounding boxes of text lines or words are easy to be annotated and obtained. An example of the ground truth map is shown in Fig. 5. The cross-entropy loss function and stochastic gradient descent are used to train this model.

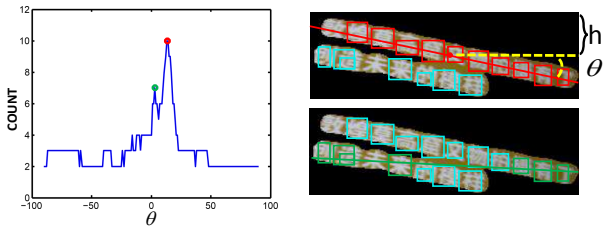
In the testing phase, the salient map of text regions, leveraging all context information from different stages, is computed by the trained Text-Block FCN model at first. As

shown in Fig. 2, the feature map of stage-1 captures more local structures like gradient (Fig. 2 (b)), while the higher level stages capture more global information (Fig. 2 (e) (f)). Then, the pixels whose probability is larger than 0.2 are reserved, and the connected pixels are grouped together into several text blocks. An example of the text block detection result is shown in Fig. 1 (c).



(a) (b)

Figure 5. The illustration of the ground truth map used in the training phase of the Text-Block FCN. (a) An input image. The text lines within the image are labeled with red bounding boxes; (b) The ground truth map.



(a) (b)

Figure 6. (a) The line chart about the counting number of components in different orientations. The right direction has the maximum value (red circle), and the wrong direction has smaller value (green circle); (b) The red line and green line correspond to circles on the line chart.

3.2. Multi-Oriented Text Line Candidate Generation

In this section, we introduce how to form multi-oriented text line candidates based on text blocks. Although the text blocks detected by the Text-Block FCN provide coarse localizations of text lines, they are still far from satisfactory. To further extract accurate bounding boxes of text lines, taking the information about the orientation and scale of text into account is required. Character components within a text line or word reveal the scale of text. Besides, the orientation of text can be estimated by analyzing the layout of the character components. At first, we extract the character components within the text blocks by MSER [16]. Then, similar to many skew correction approaches in document analysis [19], the orientation of the text lines within a text block is estimated by component projection. Finally, text line candidates are extracted by a novel method that effectively combines block-level (global) cue and component-level (local) cue.

Character Components Extraction. Our approach uses MSER [16] to extract the character components (Fig. 1(d)) since MSER is insensitive to variations in scales, orientations, positions, languages and fonts. Two constraints are adopted to remove the most of false components: area and aspect ratio. Specifically, the minimal area ratio of a character candidate needs to be more than the threshold T_1, and the aspect ratio of them must be limited to $[\frac{1}{T_2}, T_2]$. Under these two constraints, the most of the false components are excluded.

Orientation Estimation. In this paper, we assume that text lines from the same text block have a roughly uniform spatial layout, and characters from one text line are in the arrangement of straight or near-straight line. Inspired by projection profile based skew estimation algorithms in documents analysis [19], we propose a projection method according to counting components, in order to estimate the possible orientation of text lines. Suppose the orientation of text lines within a text block is θ, and the vertical-coordinate offset is h, we can draw a line across the text block (as the green or red line is shown in Fig. 6(b)). And the value of counting components $\Phi(\theta, h)$ equals the number of the character components that are passed through by the line. Since the component number in the right direction often has the maximum value, the possible orientation θ_r can be easily found if we have statistics on the peak value of counting component in all directions (Fig. 6(a)). By this means, θ_r can be easily calculated as the following formulation:

$$\theta_r = \arg\max_{\theta} \max_{h} \Phi(\theta, h) \tag{1}$$

where $\Phi(\theta, h)$ represents the number of components when the orientation is θ and the vertical-coordinate offset is h.

Text Line Candidate Generation. Different from component based methods [16, 4, 6], the process of generating text line candidates in our approach does not require to catch all the characters within a text line, under the guidance of a text block. First, we divide the components into groups. A pair of the components (A and B) within the text block α are grouped together if they satisfy following conditions:

$$\frac{2}{3} < \frac{H(A)}{H(B)} < \frac{3}{2}, \tag{2}$$

$$-\frac{\pi}{12} < O(A, B) - \theta_r(\alpha) < \frac{\pi}{12}, \tag{3}$$

where $H(A)$ and $H(B)$ represent the heights of A and B, $O(A, B)$ represents the orientation of the pair, and $\theta_r(\alpha)$ is the estimated orientation of α.

Then, for one group $\beta = \{c_i\}$, c_i is i-th component, we draw a line l along the orientation $\theta_r(\alpha)$ passing the center

of β. The point set $\mathcal{P}$ is defined as:

$$\mathcal{P} = \{p_i\}, p_i \in l \cap \mathbb{B}(\alpha), \tag{4}$$

where $\mathbb{B}(\alpha)$ represents the boundary points of α.

Finally, the minimum bounding box bb of β is computed as a text line candidate:

$$bb = \bigcup\{p_1, p_2, ...p_i, c_1, c_2, ..., c_j\}, p_i \in \mathcal{P}, c_j \in \beta, \tag{5}$$

where $\bigcup$ denotes the minimum bounding box that contains all points and components.

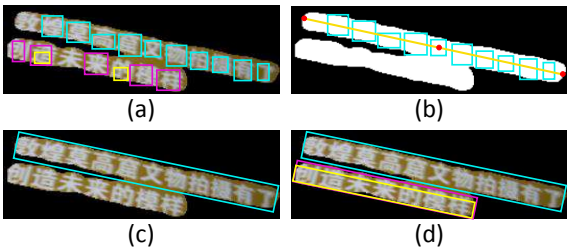


Figure 7. The illustration of text line candidate generation. (a) The character components within a text block are divided into groups; (b) The middle red point is the center of a group and the other two red points belong to $\mathcal{P}$; (c) The minimum bounding box is computed as a text line candidate; (d) All the text line candidates of this text block are extracted.

Fig. 7 illustrates this procedure. We repeat this procedure for each text block to obtain all the text line candidates within an image. By considering both of the two level cues at the same time, our approach has two advantages compared to component based methods [16, 4, 6]. First, under the guidance of text blocks, MSER components are not required to catch all characters accurately. Even though some characters are missed or partially detected by MSER, the generation of text line candidates will not be affected (such as the three candidates found at Fig. 7). Second, the previous works [29, 9, 32] of multi-oriented text detection in natural scenes usually estimate the orientation of text based on the character level with some fragile clustering/grouping algorithms. This kind of methods is sensitive to missing characters and non-text noise. Our method estimates the orientation from the holistic profile by using a projection method, which is more efficient and robust than character clustering/grouping based methods.

3.3. Text Line Candidates Classification

A fraction of the candidates generated in the last stage (Sec. 3.2) are non-text or redundancy. In order to remove false candidates, we propose two criteria based on the character centroids of text line candidates. To predict the character centroids, we employ another FCN model, named Character-Centroid FCN.

Character-Centroid FCN The Character-Centroid FCN is inherited from the Text-Block FCN (Sec. 3.1), but only the first 3 convolutional stages are used. Same as the Text-Block FCN, each stage is followed by a 1×1 convolutional layer and a upsampling layer. The fully-connected layers are also replaced with a 1×1 convolutional layer and a sigmoid layer. This network is trained with the cross-entropy loss function as well. In general, the Character-Centroid FCN is a small version of the Text-Block FCN.

Several examples along with ground truth maps are shown in Fig. 8. The positive region of the ground truth map consists of the pixels whose distance to the character centroids is less than 15% of the height of the corresponding character. In the testing phase, we can obtain the centroid probability map of a text line candidate at first. Then, extreme points $\mathcal{E} = \{(e_i, s_i)\}$ on the map are collected as the centroids, where e_i represents i-th extreme point, and s_i represents the score defined as the value of the probability map on e_i. Several examples are shown in Fig. 9.

In order to remove false candidates, two intuitive yet effective criteria based on intensity and geometric properties are adopted, after the centroids are obtained:

Intensity criterion. For a text line candidate, if the number of the character centroids $n_c < 2$, or the average score of the centroids $s_{avg} < 0.6$, we regard it as a false text line candidate. The average score of the centroids is defined as:

$$s_{avg} = \frac{1}{n_c}\sum_{i=1}^{n_c} s_i, \tag{6}$$

Geometric criterion. The arrangement of the characters within a text line candidate is always approximated to a straight line. We adopt the mean of orientation angles μ and the standard deviation σ of orientation angles between the centroids to characterize these properties. μ and σ are defined as:

$$\mu = \frac{1}{n_c}\sum_{i=1}^{n_c}\sum_{j=1}^{n_c} O(e_i, e_j), \tag{7}$$

$$\sigma = \sqrt{\frac{1}{n_c}\sum_{i=1}^{n_c}\sum_{j=1}^{n_c}(O(e_i, e_j) - \mu)^2}, \tag{8}$$

where $O(e_i, e_j)$ denotes the orientation angle between e_i and e_j. In practice, we only reserve the candidates whose $\mu < \frac{\pi}{32}$ and $\sigma < \frac{\pi}{16}$.

Through the above two constraints, the false text line candidates are excluded, but there are still some redundant candidates. To further remove the redundant candidates, a standard non-maximum suppression is applied to remaining candidates, and the score that used in non-maximum suppression is defined as the sum of the score of all the centroids.

4. Experiments

To fully compare the proposed method with competing methods, we evaluate our method on several recent stan-

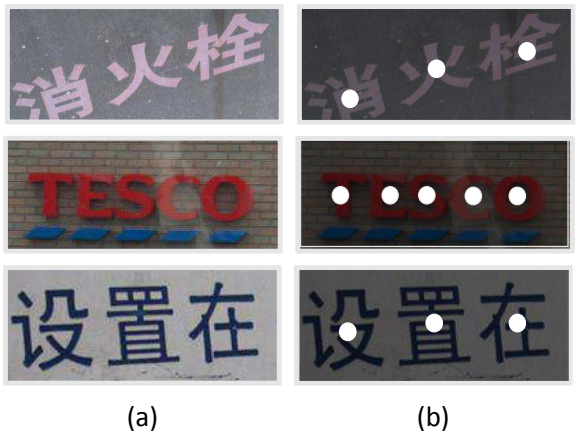


Figure 8. The illustration of the ground truth map used for training the Character-Centroid FCN. (a) Input images; (b) The ground truth maps. The white circles in (b) indicate the centroid of characters within the input images.

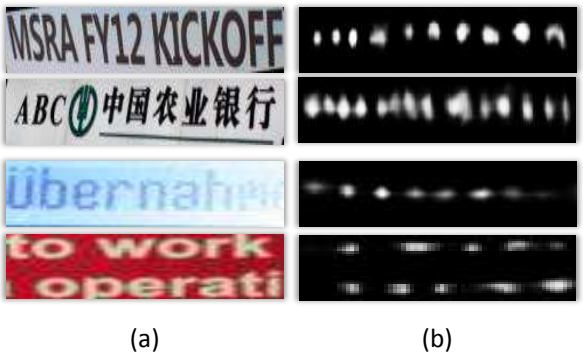


Figure 9. Examples of probability maps predicted by the Character-Centroid FCN. (a) Input images; (b) The probability maps of character centroids.

dard benchmarks: ICDAR2013, ICDAR2015 and MSRA-TD500.

4.1. Datasets

We evaluate our method on three datasets, where the first two are multi-oriented text datasets, and the third one is a horizontal text dataset.

MSRA-TD500. The MSRA-TD500 dataset introduced in [29], is a multi-orientation text dataset including 300 training images and 200 testing images. The dataset contains text in two languages, namely Chinese and English. This dataset is very challenging due to the large variation in fonts, scales, colors and orientations. Here, we followed the evaluation protocol employed by [29], which considers both of the area overlap ratios and the orientation differences between predictions and the ground truth.

ICDAR2015 - Incidental Scene Text dataset. The ICDAR2015 - Incidental Scene Text dataset is the benchmark of ICDAR2015 Incidental Scene Text Competition. This dataset includes 1000 training images and 500 testing images. Different from the previous ICDAR competition, in which the text are well-captured, horizontal, and typically centered in images, these datasets focus on the incidental scene where text may appear in any orientation and any location with small size or low resolution. The evaluation protocol of this dataset inherits from [14]. Note that this competition provides an online evaluation system and our method is evaluated in the same way.

Unlike MSRA-TD500, in which the ground truth is marked at the sentence level, the annotations of ICDAR2015 are word level. To satisfy the requirement of ICDAR 2015 measurement, we perform the word partition on the text lines generated by our method according to the blanks between words.

ICDAR2013. The ICDAR 2013 dataset is a horizontal text database which is used in previous ICDAR competitions. This dataset consists of 229 images for training and 233 images for testing. The evaluation algorithm is introduced by [11] and we evaluate our method on the ICDAR2013 online evaluation system. Since this dataset also provides word-level annotations, we adopt the same word partition procedure as we did on ICDAR 2015 dataset.

4.2. Implementation Details

In the proposed method, two models are used: the Text-Block FCN is used to generate text salient maps and the Character-Centroid FCN is used to predict the centroids of characters. Both of the two models are trained under the same network configuration. Similar to [12, 26], we also use fine-tuning with the pre-trained VGG-16 network. The two models both are trained 20×10^5 iterations in all. Learning rates start from 10^{-6}, and are multiplied by $\frac{1}{10}$ after 10×10^5 and 15×10^5 iterations. Weight decays are 0.0001, and momentums are 0.9. No dropout or batch normalization is used in our model.

All training images are harvested from the training set of ICDAR2013, ICDAR2015 and MSRA-TD500 with data augmentation. In the training phase of the Text-Block FCN, we randomly crop $30K$ 500×500 patches from the images as training examples. To compute the salient map in the testing phase, each image is proportionally resized to three scales, where the heights are 200, 500 and 1000 pixels respectively. For the Character-Centroid FCN, the patches (32×256 pixels) around the word level ground truth are collected as training examples. We randomly collect $100K$ patches in the training phase. In the testing phase, we rotate text line candidates to horizontal orientation and proportionally resize them to 32 pixels height. For all experiments, threshold values are: $T_1 = 0.5\%, T_2 = 3$.

The proposed method is implemented with Torch7 and Matlab (with C/C++ mex functions) and runs on a work-

Figure 10. Detection examples of the proposed method on MSRA-TD500 and ICDAR2013.

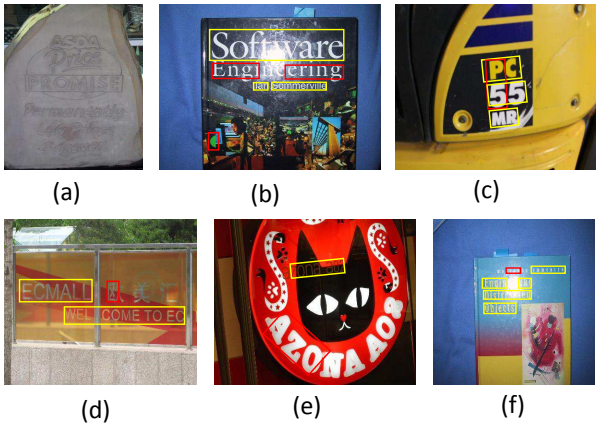


Figure 11. Several failure cases of the proposed method.

station(2.0GHz 8-core CPU, 64G RAM, GTX TitanX and Windows 64-bit OS) for all the experiments.

4.3. Experimental Results

MSRA-TD500. As shown in Tab. 1, our method outperforms other methods in both precision and recall on MSRA-TD500. The proposed method achieves precision 0.83, recall 0.67 and f-measure 0.74. Compared to [32], our method obtains significant improvements on precision (0.02), recall (0.04) and f-measure (0.03). In addition, the time cost of our method is reported in Tab. 1. Benefiting from GPU acceleration, our method takes 2.1s for each image in average on MSRA-TD500.

ICADR2015 - Incidental Scene Text. As this dataset has been released recently for the competition in ICDAR2015, there is no literature to report the experimental result on it. Therefore, we collect competition results [10] as listed in Tab. 2 for comprehensive comparisons. Our method achieves the best F-measure over all methods.

ICDAR 2013. We also test our method on the ICDAR2013 dataset, which is the most popular for horizontal text detection. As shown in Tab. 3, the proposed method achieves 0.88, 0.78, 0.83 in precision, recall and F-measure, re-

Table 1. Performance comparisons on the MSRA-TD500 dataset.

Algorithm	Precision	Recall	F-measure	Time cost
Proposed	**0.83**	**0.67**	**0.74**	2.1s
Yin *et al.* [32]	0.81	0.63	0.71	1.4s
Kang *et al.* [9]	0.71	0.62	0.66	-
Yin *et al.* [33]	0.71	0.61	0.65	**0.8s**
Yao *et al.* [29]	0.63	0.63	0.60	7.2s

Table 2. Performance of different algorithms evaluated on the ICDAR2015 dataset. The comparison results are collected from ICDAR 2015 Competition on Robust Reading [10].

Algorithm	Precision	Recall	F-measure
Proposed	0.71	0.43	**0.54**
StradVision-2	**0.77**	0.37	0.50
StradVision-1	0.53	0.46	0.50
NJU_Text	0.70	0.36	0.47
AJOU	0.47	**0.47**	0.47
HUST_MCLAB	0.44	0.38	0.41
Deep2Text-MO	0.50	0.32	0.39
CNN Proposal	0.35	0.34	0.35
TextCatcher-2	0.25	0.34	0.29

Table 3. Performance of different algorithms evaluated on the ICDAR 2013 dataset.

Algorithm	Precision	Recall	F-measure
Proposed	0.88	**0.78**	**0.83**
Zhang *et al.* [35]	0.88	0.74	0.80
Tian *et al.* [23]	0.85	0.76	0.80
Lu *et al.* [13]	**0.89**	0.70	0.78
iwrr2014 [34]	0.86	0.70	0.77
USTB TexStar [33]	0.88	0.66	0.76
Text Spotter [16]	0.88	0.65	0.74
Yin *et al.* [32]	0.84	0.65	0.73
CASIA_NLPR [1]	0.79	0.68	0.73
Text_Detector_CASIA [21]	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]	0.70	0.65	0.67

spectively, outperforming all other recent methods only designed for horizontal text.

The consistent top performance achieved on the three datasets demonstrates the effectiveness and generality of the proposed method. Besides the quantitative experimental results, several detection examples under various challenging cases of the proposed method on the MSRA-TD500 and ICDAR2013 datasets are shown in Fig. 10. As can be seen, our method successfully detects the text with inner texture in Fig. 10 (a), non-uniform illumination in Fig. 10 (b) (f), dot fonts (Fig. 10 (e)), broken strokes (Fig. 10 (h)), multiple orientations (Fig. 10 (c), and (d)), perspective distortion (Fig. 10 (h)) and mixture of multi-language (Fig. 10 (g)).

4.4. Impact of Parameters

In this section, we investigate the effect of parameters T_1 and T_2, which are used to extract MSER components for computing text line candidates. The performance of different parameters is computed on MSRA-TD500. Fig. 12 (a) and Fig. 12 (b) show how the recall of text line candidates changes under the different settings of T_1 and T_2. As we can see, the recall of text line candidates is insensitive to the change of T_1 and T_2 in a large range. This proves our method does not depend on the quality of character candidates.

4.5. Limitations of the Proposed Algorithm

The proposed method achieves excellent performance and is able to deal with several challenging cases. However, our method still has a great gap to achieve a perfect performance. Several failure cases are illustrated in Fig. 11. As can be seen, false positives and missing characters may appear in certain situations, such as extremely low contrast (Fig. 11 (a)), curvature (Fig. 11 (e)), strong reflect light (Fig. 11 (b) (f)), too closed text lines (Fig. 11 (c)), or

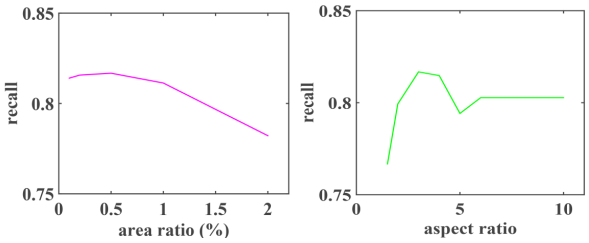


Figure 12. The recall of text line candidates with different T_1 and T_2.

tremendous gap between characters (Fig. 11 (d)). Another limitation is the speed of the proposed method, which is still far from the requirement of real-time systems.

5. Conclusion

In this paper, we presented a novel framework for multi-oriented scene text detection. The main idea that integrates semantic labeling by FCN and MSER provides a natural solution for handling multi-oriented text. The superior performance over other competing methods in the literature on both horizontal and multi-oriented text detection benchmarks verifies that combining local and global cues for text line localization is an interesting direction that is worthy of being studied. In the future, we could extend the proposed method to an end-to-end text recognition system.

Acknowledgement

This work was primarily supported by National Natural Science Foundation of China (NSFC) (No. 61222308, No. 61573160, No. 61572207, and No. 61303095), and Open Project Program of the State Key Laboratory of Digital Publishing Technology (No. F2016001).

References

[1] ICDAR 2013 robust reading competition challenge 2 results. http://dag.cvc.uab.es/icdar2013competition, 2014. [Online; accessed 11-November-2014]. 8

[2] A. Bissacco, M. Cummins, Y. Netzer, and H. Neven. PhotoOCR: Reading text in uncontrolled conditions. In *Proc. of ICCV*, 2013. 1

[3] X. Chen and A. Yuille. Detecting and reading text in natural scenes. In *Proc. of CVPR*, 2004. 1, 2

[4] B. Epshtein, E. Ofek, and Y. Wexler. Detecting text in natural scenes with stroke width transform. In *Proc. of CVPR*, 2010. 1, 2, 3, 4, 5

[5] 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. 2

[6] W. Huang, Y. Qiao, and X. Tang. Robust scene text detection with convolution neural network induced mser trees. In *Proc. of ECCV*, 2014. 1, 2, 3, 4, 5

[7] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman. Reading text in the wild with convolutional neural networks. *IJCV*, pages 1–20, 2014. 3

[8] M. Jaderberg, A. Vedaldi, and A. Zisserman. Deep features for text spotting. In *Proc. of ECCV*, 2014. 1, 2, 3

[9] L. Kang, Y. Li, and D. Doermann. Orientation robust text line detection in natural images. In *Proc. of CVPR*, 2014. 3, 5, 7

[10] D. Karatzas, L. Gomez-Bigorda, A. Nicolaou, S. Ghosh, A. Bagdanov, M. Iwamura, J. Matas, L. Neumann, V. R. Chandrasekhar, S. Lu, et al. Icdar 2015 competition on robust reading. In *Proc. of ICDAR*, 2015. 1, 7

[11] 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. 1, 6

[12] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proc. of CVPR*, 2015. 1, 3, 6

[13] S. Lu, T. Chen, S. Tian, J.-H. Lim, and C.-L. Tan. Scene text extraction based on edges and support vector regression. *IJDAR*, 18(2):125–135, 2015. 8

[14] S. M. Lucas. ICDAR 2005 text locating competition results. In *Proc. of ICDAR*, 2005. 6

[15] L. Neumann and J. Matas. A method for text localization and recognition in real-world images. In *Proc. of ACCV*, 2010. 1, 2

[16] L. Neumann and J. Matas. Real-time scene text localization and recognition. In *Proc. of CVPR*, 2012. 1, 3, 4, 5, 8

[17] L. Neumann and J. Matas. Scene text localization and recognition with oriented stroke detection. In *Proc. of ICCV*, 2013. 1, 2

[18] Y. Pan, X. Hou, and C. Liu. A hybrid approach to detect and localize texts in natural scene images. *IEEE Trans. on Image Processing*, 20(3):800–813, 2011. 2, 3

[19] C. Postl. Detection of linear oblique structures and skew scan in digitized documents. In *Proc. of ICPR*, 1986. 4

[20] S. Qin and R. Manduchi. A fast and robust text spotter. In *Proc. of WACV*, 2016. 1

[21] 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. 8

[22] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. 2015. 3

[23] S. Tian, Y. Pan, C. Huang, S. Lu, K. Yu, and C. Lim Tan. Text flow: A unified text detection system in natural scene images. In *Proc. of CVPR*, 2015. 1, 8

[24] K. Wang and S. Belongie. Word spotting in the wild. In *Proc. of ECCV*, 2010. 1

[25] 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. 1, 3

[26] S. Xie and Z. Tu. Holistically-nested edge detection. 2015. 3, 6

[27] B. Xiong and K. Grauman. Text detection in stores using a repetition prior. In *Proc. of WACV*, 2016. 1

[28] C. Yao, X. Bai, and W. Liu. A unified framework for multi-oriented text detection and recognition. *IEEE Trans. on Image Processing*, 23(11):4737–4749, 2014. 1, 2, 3

[29] C. Yao, X. Bai, W. Liu, Y. Ma, and Z. Tu. Detecting texts of arbitrary orientations in natural images. In *Proc. of CVPR*, 2012. 3, 5, 6, 7

[30] Q. Ye and D. Doermann. Text detection and recognition in imagery: A survey. *IEEE Trans. on PAMI*, (99), 2014. 2

[31] C. Yi and Y. Tian. Text string detection from natural scenes by structure-based partition and grouping. *IEEE Trans. on Image Processing*, 20(9):2594–2605, 2011. 3

[32] X.-C. Yin, W.-Y. Pei, J. Zhang, and H.-W. Hao. Multi-orientation scene text detection with adaptive clustering. *IEEE Trans. on PAMI*, (1):1–1. 1, 3, 5, 7, 8

[33] 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. 2, 7, 8

[34] 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. 8

[35] Z. Zhang, W. Shen, C. Yao, and X. Bai. Symmetry-based text line detection in natural scenes. In *Proc. of CVPR*, 2015. 1, 3, 8

[36] Y. Zhu, C. Yao, and X. Bai. Scene text detection and recognition: Recent advances and future trends. *Frontiers of Computer Science*, 10(1):19–36, 2016. 2