Pattern Recognition 96 (2019) 106954

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

SegLink++: Detecting Dense and Arbitrary-shaped Scene Text by Instance-aware Component Grouping

Jun Tang[a,b], Zhibo Yang[b], Yongpan Wang[b], Qi Zheng[b], Yongchao Xu[a,*], Xiang Bai[a]

[a] School of EIC, Huazhong University of Science and Technology, Wuhan 430074, China
[b] Alibaba-Group, Hangzhou 311121, China

ARTICLE INFO

Article history:
Received 1 April 2019
Revised 6 June 2019
Accepted 24 June 2019
Available online 25 June 2019

Keywords:
Scene text detection
Multi-oriented text
Curve text
Dense text

ABSTRACT

State-of-the-art methods have achieved impressive performances on multi-oriented text detection. Yet, they usually have difficulty in handling curved and dense texts, which are common in commodity images. In this paper, we propose a network for detecting dense and arbitrary-shaped scene text by instance-aware component grouping (ICG), which is a flexible bottom-up method. To address the difficulty in separating dense text instances faced by most bottom-up methods, we propose attractive and repulsive link between text components which forces the network learning to focus more on close text instances, and instance-aware loss that fully exploits context to supervise the network. The final text detection is achieved by a modified minimum spanning tree (MST) algorithm based on the learned attractive and repulsive links. To demonstrate the effectiveness of the proposed method, we introduce a dense and arbitrary-shaped scene text dataset composed of commodity images (DAST1500). Experimental results show that the proposed ICG significantly outperforms state-of-the-art methods on DAST1500 and two curved text datasets: Total-Text and CTW1500, and also achieves very competitive performance on two multi-oriented datasets: ICDAR15 (at 7.1FPS for 1280×768 image) and MTWI.

1. Introduction

Recently, scene text detection has drawn much attention thanks to its wide applications such as image retrieval, scene understanding and automatic driving. Scene text detection differs from general object detection, because texts vary significantly on aspect ratio, scale, orientation, and shape [1]. Besides, scene text usually occurs on uncontrollable scene with important light and perspective variations, which is quite different from traditional OCR.

To deal with these challenges, traditional text detection pipelines [2–10] usually consist of multi-stage process: they first extract component regions with engineered features [11,12], then filter the candidate components [13], finally group the extracted components [5,8] into text instances. These traditional methods are limited by the engineered features, and usually involve a heavy post-processing.

In recent years, thanks to the development of deep learning, many methods [14–45] adopt convolutional neural network (CNN) to extract features and achieve impressive improvements. In general, recent deep learning-based methods can be divided into top-down and bottom-up methods. The top-down methods are mainly inspired by the development of general object detection pipelines [46–48] and focus on addressing the multi-orientation and large aspect ratio problems of text detection. A top-down method [14–16,18–23] usually directly regresses horizontal/oriented rectangles or quadrangles. The bottom-up methods [17,24–40], on the other hand, follow the key idea of traditional text detection: first detect text components with CNN and then group these components into text instances. Bottom-up methods are more flexible and capable of detecting texts of arbitrary shapes. Considering the granularity of the components, bottom-up methods can be further divided into pixel-level bottom-up methods [27–34,36,37,49] namely segmentation-based methods, and part-level bottom-up methods [17,24–26,35,40].

The top-down deep learning-based methods have achieved impressive performances on multi-oriented text detection. Yet, they usually fail to detect curved texts and texts of large aspect ratio. The performance of top-down methods drops significantly on curved text detection. Thanks to the flexibility of bottom-up methods, curved texts can be properly detected in a bottom-up way. Recently, on curved text detection benchmarks [23,34], bottom-up methods [40] surpass the top-down methods by a large margin. Yet, due to heavy post-processing, bottom-up methods have a

* Corresponding author.
E-mail addresses: tjbestehen@gmail.com (J. Tang), zhibo.yzb@alibaba-inc.com (Z. Yang), yongpan@taobao.com (Y. Wang), yongqi.zq@taobao.com (Q. Zheng), yongchaoxu@hust.edu.cn (Y. Xu), xbai@hust.edu.cn (X. Bai).

https://doi.org/10.1016/j.patcog.2019.06.020
0031-3203/

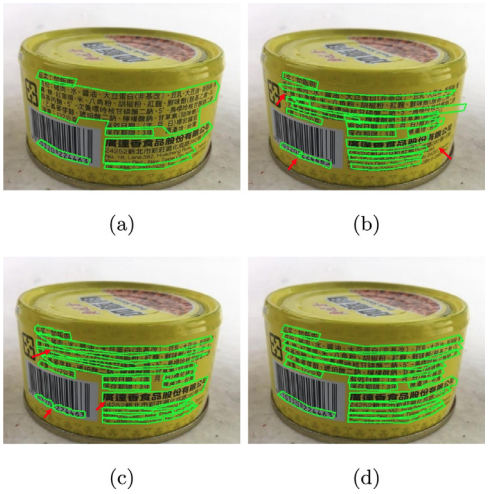


Fig. 1. Comparison of different scene text detectors on one proposed DAST1500 image. (a) SegLink [25]; (b) CTD+TLOC [23]; (c) PixelLink [36]; (d) Proposed ICG. The proposed ICG can accurately detect dense and arbitrary-shaped scene texts.

bottleneck in efficiency. Furthermore, it is usually difficult for bottom-up methods to separate close text instances. Though dense and curved texts are rather rare in current curved text detection benchmarks, they frequently appear in scene images especially commodity images (see Fig. 1 for example). So, it is of great interest to introduce a dense and arbitrary-shaped text detection dataset and propose a method that can accurately deal with the dense and arbitrary-shaped scene text detection.

In this paper, we propose an instance-aware component grouping (ICG) framework to address the dense and arbitrary-shaped text detection problem. It is inspired by the mutex watershed algorithm [50] for neuron segmentation. By explicitly introducing repulsive links between pixels or parts of different text instances, together with the attractive links within the same text instance, the proposed method ICG is able to cope with very dense and arbitrary-shaped text detection. For the sake of simplicity, we follow the design in SegLink [25] to learn to detect text components. Such text component extraction and attractive/repulsive link estimation are achieved with shared CNN feature. To fully exploit context that facilitates to separate close text instances, we also propose an instance-aware loss to give more loss weights on the components and attractive/repulsive links of poorly detected target text regions. It is worth to note that both proposed attractive/repulsive and instance-aware loss are technically not limited to SegLink, and are likely to be beneficial for other bottom-up scene text detection methods.

To advance dense and arbitrary-shaped text detection and demonstrate the effectiveness of the proposed ICG, we also introduce a dense and arbitrary-shaped scene text dataset of commodity images named DAST1500. It mainly consists of commodity images with detailed descriptions of the commodity on small wrinkled package, which are collected on from Internet. We adopt the standard PASCAL VOC evaluation protocol for benchmark on this dataset. We release this dataset by the following link: https://tianchi.aliyun.com/dataset/dataDetail?dataId=12084. The proposed ICG boosts the performance on DAST1500 by over 10 percents compared to its baseline, and significantly outperforms other methods. It also achieves state-of-the-art results on curved text detection dataset CTW1500 [23] and TotalText [34], and is also very competitive on multi-oriented text detection on ICDAR15 [51] and MTWI [52]. Note that the proposed method ICG is not limited to the baseline SegLink, and can be applied to most bottom-up methods to boost the performance on dense and arbitrary-shaped text detection.

The main contributions of this paper are three folds: (1) We propose an instance-aware component grouping framework, which can be applied to most bottom-up methods to boost the performance on dense and arbitrary-shaped text detection. (2) We introduce a dense and arbitrary-shaped scene text dataset to advance dense and arbitrary-shaped text detection. (3) The proposed ICG boosts the performance on DAST1500 by a large margin, it also outperforms state-of-the-art methods on curved text detection datasets, and achieves very competitive results on multi-oriented text detection datasets.

The rest of the paper is organized as follows. We shortly review some related works on scene text detection in Section 2, followed by the detail of the proposed method in Section 3. Then we present experiment results in Section 4. Finally, we conclude and give some perspectives on the future work in Section 5.

2. Related work

2.1. Scene text detection

Text Detection before the deep learning era follows a similar and typical bottom-up pipeline, including text component extraction and filtering, text region grouping and text region candidate filtering. Many works [2,4,6–8] focus on text component extraction with engineered features, such as Maximally Stable Extremal Regions (MSER) [11] and Stroke Width Transform (SWT) [12]. In the past few years, deep learning-based methods are fully explored on text detection, and the performance of these methods exceed traditional methods by a large margin both in accuracy and efficiency. In general, they can be roughly divided into two categories: top-down methods and bottom-up methods.

Top-down scene text detection. Top-down methods are mainly inspired by recent development on general object detection [46–48]. Based on the pre-extracted proposal, regression is calculated to produce bounding boxes. TextBoxes [15] adopted SSD network and applies long default boxes as well as convolution kernel to deal with large aspect ratio variation of text in text detection. TextBoxes++ [19] further extended TextBoxes to detect multi-oriented text by regressing corner coordinates of text polygon. SSTD [16] introduced attention map by FCN to guide the training process of text detection and enhanced the detection of multi-scale text. To handle the variants of orientation, many methods are proposed. R2CNN [18] adapted the Faster-RCNN pipeline and added Rotated Regional Proposal to produce oriented bounding boxes. Liu et al. [14] proposed Deep Matching Prior Network to detect multi-oriented boxes. Liao et al. [20] introduced rotation-sensitive feature for detection branch and rotation-invariant feature for classification branch to learn better regression of long oriented text. Wang et al. [21] proposed instance transformation network to learn the geometry-aware representation of text orientation.

Bottom-up scene text detection. Bottom-up methods follow a similar pipeline: first detect text components, and then group these components into text instances. Recently, more and more bottom-up methods are present to detect arbitrary-shaped scene texts. From the perspective of representation, bottom-up methods can be divided into pixel-level and part-level methods.

Pixel-level Pixel-level bottom-up scene text detection methods consider text detection as a text area segmentation problem. Fully Convolutional Network (FCN) is thus usually adopted to produce pixel-wise classification map, and some post-processing is then involved to group text pixels into text instances. Zhang et al. [28] predicted text segmentation map and the centroids of each character, which are then used to generate text instances. Yao et al. [29] followed a similar pipeline to extract characters and text

regions via a FCN in a holistic fashion. In [31], multi-scale outputs of FCN are utilized to produce text instances with cascade FCN. Wu et al. [30] proposed to regard the text region segmentation problem as three class segmentation by introducing a border class. Xue et al. [37] further developed this idea by introducing semantic-aware text borders and bootstrapping technique to generate more training examples. Deep Direct Regression [33] and EAST [32] followed a similar pipeline to learn pixel-based text polygon estimation. PixelLink [36] introduced 8-direction link to identify text borders and group text instances. Liu et al. [27] formulated the image as stochastic flow graph in pixel level, followed by a Markov clustering to generate text regions.

Part-level For part-level methods, text region is regarded as a group of text components. Tian et al. [24] proposed Connectionist Text Proposal Network (CTPN) to first detect vertical text parts of fixed width, and then grouped these text parts by recurrent neural network (RNN). SegLink [25] learned segments and links between 8-neighbor segments to group into text instances. In [26], the authors proposed to combine four detected corner boxes along with four part segmentation maps to generate text instances. Recently, in order to detect curved text, Liu et al. [23] proposed a method named CTD to regress multiple points on text instances and refine the results with relations between points by proposed TLOC. TextSnake [40] regarded text region as a group of disks to achieve curved text detection.

2.2. Comparison with related works

Compared with the traditional text detection methods, the proposed ICG follows a similar pipeline, but significantly enhances the performance (including detection precision and efficiency) of text detection with learnable text components and attractive/repulsive links. Compared with top-down deep learning-based methods, the proposed ICG has the advantages to accurately detect arbitrary-shaped texts while maintaining competitive results on multi-oriented text detection. The proposed ICG falls into bottom-up methods. ICG aims to deal with the issue of dense and arbitrary-shaped text detection, which is not thorough discussed in the related works of bottom-up methods. The proposed attractive and repulsive links help to separate close text instances, leading to enhanced performance in detecting dense and arbitrary-shaped scene texts. The proposed instance-aware loss somehow compensates the drawback of bottom-up methods which usually involve a post-processing that cannot be trained in an end-to-end way. Furthermore, the proposed attractive and repulsive links and instance-aware loss can be applied to most other bottom-up methods to boost performance on dense and arbitrary-shaped text detection.

3. Methodology

3.1. Overview

Top-down text detection methods are usually driven by general object detection pipeline based on deep learning. They have achieved impressive results on multi-oriented text detection. Yet, they often have problems in generalizing to curved text detection, which are common in scene images. Bottom-up text detection methods are more flexible in handling texts of arbitrary shapes, and become the mainstream for detecting arbitrary-shaped scene texts. Bottom-up text detection methods mainly suffer from two major issues: (1) Difficulty in separating close text instances. For the area of dense texts, several text instances lie very close to each other. The estimated text regions may stick together, making it difficult to extract each individual text instance; (2) Heavy post-processing which is not optimized in an end-to-end way. Bottom-up text methods usually detect texts by first extracting text parts or pixels, and then grouping them into text instances with a post-processing. The involved post-processing is usually not differentiable and is thus not optimized together with the network training, leading to non-optimized text detections.

We follow the bottom-up text detection pipeline to detect dense and arbitrary-shaped texts, which are very common in commodity images. To address the aforementioned two major problems faced by bottom-up methods, we propose an instance-aware component grouping framework. The whole pipeline is depicted in Fig. 2. Specifically, we leverage convolutional neural network to

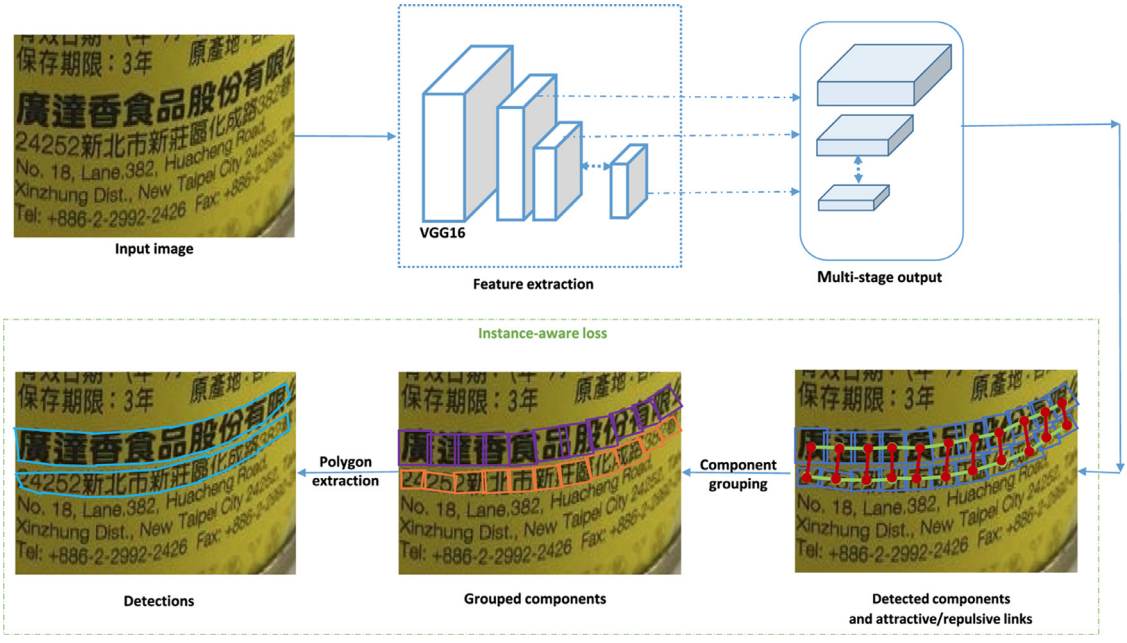


Fig. 2. Pipeline of the proposed method. Given an image, the network extracts multi-level features to predict text components and attractive/repulsive link estimation. For the sake of simplicity, we only illustrate part of text components (blue quadrangles), attractive links (green edges linking pairs of points), and repulsive links (red edges linking pairs of points). The text component grouping is then achieved by a modified minimum spanning tree based on the predicted attractive/repulsive links (different colors for different groups of text components). The final detection result is given in terms of polygons extracted from the grouped text components. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

regress text component parts, and explicitly learn attractive and repulsive links between text component parts. We also introduce an instance-aware loss to supervise the network learning such that the post-processing step is better involved in the network optimization. The final text detection is achieved by a modified minimum spanning tree algorithm based on the learned attractive and repulsive links, followed by polygon extraction and polygon non-maximum suppression (NMS).

The proposed instance-aware component grouping framework detailed in Section 3.2 is rather general and can be applied for any bottom-up methods. In this paper, we adopt SegLink [25] (but not limited to this baseline) based on SSD [47] as the backbone for the sake of flexibility and efficiency. This adopted network architecture is depicted in Section 3.3, followed by the ground-truth generation for training the network in Section 3.4. Then we present the network optimization in Section 3.5. The inference and post-processing to achieve the final text detection is given in Section 3.6.

3.2. Instance-aware component grouping framework

Bottom-up text detection methods are usually more flexible in detecting dense and arbitrary-shaped scene texts. To alleviate the two major problems of bottom-up text detection methods: difficulty in separating close text instances and non-optimized post-processing, we propose an instance-aware component grouping (ICG) framework. Specifically, the proposed method consists of two modules, each of which aims to address one of the two challenges, respectively. They are detailed in the following:

Text component grouping with attractive and repulsive links. A text instance in image is usually composed of a sequence of close characters with the same geometrical properties. Bottom-up methods are flexible in handling arbitrary-shaped texts by first extracting text component parts followed by component grouping. The post-grouping process is either based on heuristic rules, combination rules, or learned link relationships between text parts. For the sake of flexibility, we also follow the bottom-up methods. We leverage the convolutional neural network to regress text component parts following SSD [47] via default boxes. In addition to learn the attractive links between text component parts, we also explicitly learn the repulsive links between text parts, helping to separate close text instances. To cope with multi-scale text detections, we learn the attractive and repulsive links for different image resolutions (*i.e.*, different stages in the deep network). Note that both attractive and repulsive links are defined and learned for within-layer links and cross-layer links between neighboring components. This forms an edge-weighted graph-like representation $G = (V, E)$, where the nodes V are the points in multi-resolution pyramid images and edges E are the links between neighboring points within-layer or cross-layer. Each edge e is weighed by an attractive force $w_a(e)$ and a repulsive force $w_r(e)$. We then apply a modified minimum spanning tree (MST) inspired by mutex watershed [50] to group text components into text instances based on the attractive weights w_a and repulsive weights w_r.

Network training with instance-aware loss. Bottom-up text detection methods often suffer from non-optimized post-processing in the sense that it is usually difficult to integrate the post-processing in network training. To alleviate such issue, we propose an instance-aware loss by integrating the MST-based post-processing in network training and considering overlapping degree between detected region with ground-truth text instances. Specifically, For each ground-truth text instance g_i, we compute the intersection-over-union (IoU) with each detected text instance d, and identify the detected text instance d_i^m having the largest IoU (denoted as IoU_i^m) with respect to g_i. Then we weigh the loss for text component regression and attractive/repulsive estimation within g_i with $1/IoU_i^m$. This instance-aware loss weight ω make the network training focusing more on hard text instance, boosting the detection performance.

Note that such instance-aware component grouping framework is suitable for most bottom-up deep learning-based text detection methods. Hopefully, explicitly learning repulsive links between text parts helps to separate close text instances, and involving post-processing in the network training could boost the performance.

3.3. Network architecture

For the sake of flexibility and simplicity, we adopt a network architecture (depicted in Fig. 3) similar to SegLink [25] based on SSD [47]. Concretely, we adopt VGG16 [53] as the backbone network to extract image features, and convert the last two fully connected layers (*i.e.*, *fc*6 and *fc*7) to convolutional layers denoted as *conv*6 and *conv*7, respectively. A few extra convolutional layers (ranging from *conv*8_1 to *conv*11) are added to extract deeper features with larger receptive fields, better coping with multi-scale text detection. We perform text component extraction and attractive/repulsive link estimation by 3×3 convolutions on six selected layers indexing with $l = \{1, 2, \ldots, 6\}$: *conv*4_3, *conv*7, *conv*8_2, *conv*9_2, *conv*10_2, and *conv*11. The leanring target in terms of network output can be divided into two parts detailed in the following:

Text component extraction. For arbitrary-shaped text detection, we first learn to extract oriented text component parts represented by (x, y, w, h, θ), where (x, y) is the coordinate of component center, w and h are the width and height of the component, θ is the orientation. For that, we adopt square default boxes in SSD and set the height of default boxes in each involved layer $h_d^l, l = 1, 2, \ldots, 6$

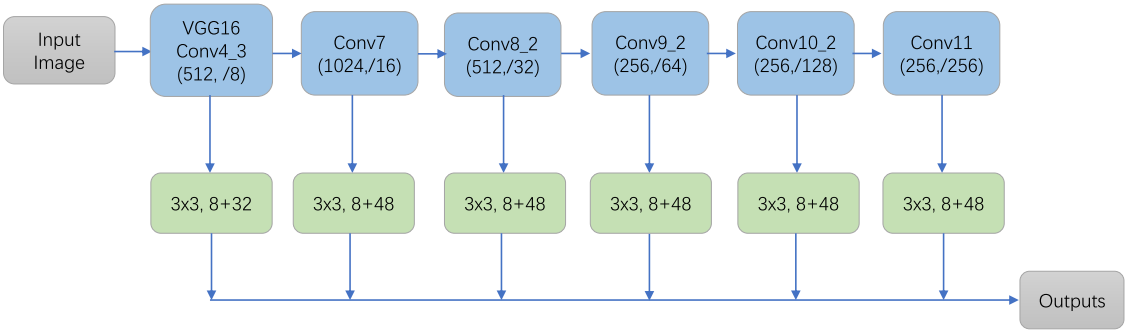


Fig. 3. Network architecture. We adopt pre-trained VGG16 as backbone network and follow the multi-stage prediction of SegLink [25] to produce multi-stage outputs of text component extraction and attractive/repulsive link estimation.

to 12, 24, 45, 90, 150, 285, respectively. The default box height for each layer l is approximately set by the ratio to adjust the scale of components $h_d^l = a_l \approx \gamma \frac{w_l}{w_l}$, where w_l and w_l are image width and feature map width of layer l, respectively, and γ is a hyper-parameter (set to 1.5 in this paper). This text component extraction part outputs a 8-channel map, where 2-channel in terms of Softmax layer is used to produce classification score s, and the other 6-channel is reserved for the geometrical properties of oriented text component representation. Note that for the orientation θ, we regress $sin\theta$ and $cos\theta$ instead of directly regressing θ. Concretely, for the considering layer l, let $(x_g, y_g, w_g, h_g, \theta_g)$ be the ground-truth text component (see Section 3.4 for its generation) at position (x_d^l, y_d^l). Then in addition to text classification, this text component extraction part aims to regress $(\Delta x, \Delta y, \Delta w, \Delta h, \Delta \sin\theta, \Delta \cos\theta)$ given by:

$$\Delta x = \frac{x_g - x_d^l}{a_l}, \tag{1}$$

$$\Delta y = \frac{y_g - y_d^l}{a_l}, \tag{2}$$

$$\Delta w = \log\left(\frac{w_g}{a_l}\right), \tag{3}$$

$$\Delta h = \log\left(\frac{h_g}{a_l}\right), \tag{4}$$

$$\Delta \sin\theta = \sin\theta_g, \tag{5}$$

$$\Delta \cos\theta = \cos\theta_g. \tag{6}$$

Attractive and repulsive link estimation. The proposed network then estimates the attractive and repulsive links between text components. We formulate the attractive and repulsive estimation as two binary-classification problems via Softmax layer, which requires four channels. As described in Section 3.2, we predict attractive/repulsive weights for both within-layer and cross-layer links. For the within-layer links, we adopt 8-connectivity. For the cross-layer links, except the first layer $l = 1$, each point p in layer l is linked to the corresponding four points in layer $l-1$ from which p is pooled. Thus, the first layer $l = 1$ outputs $8 \times 4 = 32$ channels, and each of the rest layer outputs $(8+4) \times 4 = 48$ for attractive and repulsive link estimation.

3.4. Ground-truths generation

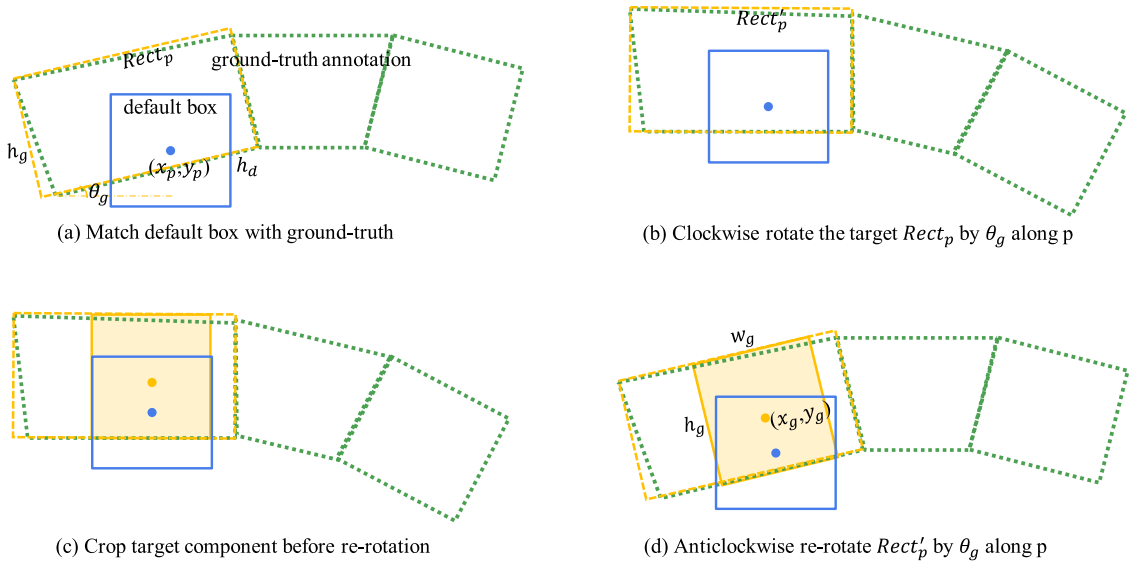
For scene text detection, the ground-truths are usually annotated in terms of oriented bounding boxes or quadrangles for multi-oriented text detection, and polygons composed of multiple quadrangles for curved text detection. To train the proposed network, we need to generate text component level ground-truths and local attractive and repulsive link ground-truths. Without loss of generality, we illustrate how to generate such ground-truths in Fig. 4 using curved text annotation in terms of polygons composed of multiple quadrangles.

For a given text pixel $p = (x_p, y_p)$, let $Rect_p$ be the besting fitting oriented rectangle of the corresponding (divided) quadrangle containing p. We aim to compute the ground-truth text component $(x_g, y_g, w_g, h_g, \theta_g)$ on pixel p. θ_p is given by the angle of oriented rectangle $Rect_p$. For the other four geometrical properties, we first clockwise rotate the oriented rectangle $Rect_p$ by θ_g along the underlying default box center $p = (x_p, y_p)$ denoted as $Rect'_p$, aligning with the horizontal square default box (see Fig. 4(b)). We then crop the rotated $Rect'_p$ by horizontally aligning with the default box and vertically fitting $Rect'_p$ (see Fig. 4(c)). Note that if part of the default box is horizontally outside $Rect'_p$, then the cropping is horizontally limited to the horizontal border of $Rect'_p$. Finally, we anticlockwise re-rotate $Rect'_p$ by θ_g along the center point p to get back to the original position. The rotated yellow box in Fig. 4(d) is the corresponding ground-truth text component for the underlying blue default box in Fig. 4.

For a given text pixel p, the underlying default box centered at p is positive only when the following condition is satisfied:

$$max\left(\frac{h_d}{h_g}, \frac{h_g}{h_d}\right) \leq 1.5. \tag{7}$$

where h_d is the height of the underlying default box. When multiple ground-truth text instances contain a given p (this may happen in case of annotations of very dense texts), the ground-truth text component with smallest ratio in Eq. (7) is considered as the



(a) Match default box with ground-truth

(b) Clockwise rotate the target $Rect_p$ by θ_g along p

(c) Crop target component before re-rotation

(d) Anticlockwise re-rotate $Rect'_p$ by θ_g along p

Fig. 4. Ground-truth text component generation. For a given default box (in blue) centered at $p = (x_p, y_p)$, the yellow box in (d) is its corresponding target ground-truth text component. See the corresponding text for details about the generation. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

ground-truth text component. The pixel p is considered to belong to that ground-truth text instance for the following attractive/repulsive weight estimation. When the condition in Eq. (7) is not satisfied for any ground-truth text instance, the default box is negative. Note that all the default boxes centered at non-text pixels are considered as negative default boxes. Thus, there is no need to generate ground-truths text components on non-text pixels.

For a given link $e = (p_1, p_2) \in E$, when both p_1 and p_2 belong to the same ground-truth text instance, then the attractive weight for e is set to 1 $w_a(e) = 1$. When p_1 and p_2 belong to different ground-truth text instances, then the repulsive weight for e is set to 1 $w_r(e) = 1$. For the other cases of an edge e linking two points, both attractive and repulsive weights are set to 0 $w_a(e) = w_r(e) = 0$.

3.5. Optimization

Training objective. The proposed network can be regarded as a multi-task network aiming at text component extraction and attractive/repulsive estimation. For the text component extraction, we adopt classic object detection loss. Specifically, let s_g (resp., l_g) and s_p (resp., l_p) be the ground-truth and predicted text score (location), respectively. Then the loss L_C for text component extraction is given by:

$$L_C(s_g, l_g, s_p, l_p) = (L_{conf}(s_g, s_p, \omega) + \alpha \times L_{loc}(l_g, l_p, \omega))/N_d, \tag{8}$$

where N_d is the number of matched default boxes, α is a hyper-parameter, L_{conf} is a 2-class Softmax loss, L_{loc} is the smooth L1 loss, and ω is the instance-aware loss weight described in Section 3.2.

For the attractive and repulsive weight estimation, let w_a^g (resp., w_r^g) and w_a^p (resp., w_r^p) be the ground-truth and predicted attractive (resp., repulsive) weight, respectively, then the loss for attractive and repulsive weight estimation is given by:

$$L_E = \left(L_{conf}\left(w_a^g, w_a^p, \omega\right) + \beta \times L_{conf}\left(w_r^g, w_r^p, \omega\right)\right)/(N_a + N_r), \tag{9}$$

where N_a and N_r are the number of attractive and repulsive links, respectively, β is a hyper-parameter. We adopt a 2-class Softmax loss for L_{conf}.

The final loss L for the proposed network training is given by:

$$L = \lambda_1 \times L_C + \lambda_2 \times L_E, \tag{10}$$

where λ_1 and λ_2 are two hyper-parameters.

Online hard negative mining. Since text region usually occupies a small area of a scene image, we also adopt online hard negative mining when optimizing the network. Specifically, we set the ratio between positive and negative examples to 3 for both text component extraction and attractive/repulsive weight estimation. Note that for the hard negative mining on attractive/repulsive weights, we only consider the negative edges that link to at least one text pixel.

3.6. Inference and post-processing

During inference, we feed a given image to the network and produce text component classification score s_p and corresponding geometrical properties for text component extraction, and attractive and repulsive weight estimation w_a and w_r. Note that for the within-layer links, each edge e is predicted twice for $e = (p_1, p_2)$ and $e' = (p_2, p_1)$, the final w_a and w_r for edge e is given by the maximum value: $w_a(e) = \max(w_a(e), w_a(e'))$, $w_r(e) = \max(w_r(e), w_r(e'))$. We then adopt a modified minimum spanning tree algorithm inspired by mutex watershed [50] to group text components into candidate text instances, followed by a polygon combination and polygon NMS to achieve the final text detection.

Modified MST for text component grouping. We adopt a modified minimum spanning tree inpired by mutex watershed [50] to group predicted text components into candidate text detections. The algorithm is depicted in Algorithm 1. Specifically, we first identify

Algorithm 1: Modified MST for text component grouping based on learned text score s_p, attractive w_a and repulsive w_r links.

```
Text_Inference(w_a, w_r, s_p, t_s, t_l)
//Initialization
E+ = {e = (p1, p2) ∈ E | w_a(e) > t_l and max(s_p(p1), s_p(p2)) > t_s};
E− = {e = (p1, p2) ∈ E | w_r(e) > t_l and max(s_p(p1), s_p(p2)) > t_s};
A+ ← ∅, A− ← ∅; //set of attractive (resp. repulsive) links in final MST ;
for e = (p1, p2) ∈ E+ ∪ E− in descending order of max(w_a, w_r) do
    if w_a(e) > w_r(e) then
        if not connect(p1, p2) and not mutex(p1, p2) then
            // merge p1 and p2 and update mutex constraints;
            merge(p1, p2), A+ ← A+ ∪ e ;
    else if w_a(e) ≤ w_r(e) then
        if not connect(p1, p2) then
            addmutex(p1, p2) : A− ← A− ∪ e // add mutex constraint between p1 and p2;
D ← CC_Labeling(A+); //Grouping by connected labeling;
return D;
```

the candidate text components by thresholding predicted text score s_p with a thresholding value t_s, and locate the important attractive links E^+ (*resp.* repulsive links E^-) whose weight w_a (*resp.* w_r) is larger than a thresholding value t_l. Then we traverse each linking edge $e \in E^+ \cup E^-$ in decreasing order of maximum attractive and repulsive weights. For each underlying linking edge $e = (p_1, p_2)$, we compare its attractive weight and repulsive weight, deciding whether merging the text components containing p_1 and p_2, respectively, or add mutex constraint. For two text components already having a mutex constraint added by a larger repulsive weight, the current linking edge with a lower attractive weight will not result in a merging process, and *vice versa*. For each merging, we also update the mutex constraints for the merged text component. Finally, a connected labeling based on the selected merging linking edges A^+ is applied to achieve the final text component grouping, which is represented by a label map D.

polygon combination. We then transform the grouped text components to be coherent with the ground-truth annotation. Concretely, we first filter out some predicted text components that are not grouped to any other text components. Then for multi-oriented (resp., curved) text detection, we transform each grouped text component into a minimum oriented rectangle (resp., a polygon with minimum edges). Finally, we also filter out some detections having small average height or small area.

polygon NMS. To further get rid of some redundant polygons, we adopt a polygon non maximum suppression (NMS) based on a modified $IoU' = |A \cap B|/min(|A|, |B|)$ for two polygons A and B, where $|\cdot|$ denotes the cardinality. The score for NMS is the ratio between area of polygon and its average height. This polygon NMS helps to get rid of some small and redundant detections.

4. Experiments

The proposed method is dedicated for dense and arbitrary-shaped scene text deteciton. To demonstrate the effectiveness in detecting such texts, we first introduce a dataset of commodity images named DAST1500, which contains many dense and arbitrary-shaped texts. We conduct ablation study on this dataset, and compare with other state-of-the-art methods on this dataset. We also benchmark the proposed method on two multi-oriented text detection datasets and two curved text detection datasets.

4.1. Dataset and evaluation protocol

DAST1500 Datasets Various challenging datasets are constructed to advance the development of robust OCR. Examples are [23,34,51,52]. To the best of our knowledge, current publicly available datasets contain very little dense and arbitrary-shaped texts, which are frequently appeared on commodity detail images. Density comes from the limited space for introduction, while irregularity results from the easily wrinkled packing. Robustly and automatically reading texts in these commodity images greatly facilitates many applications such as goods surveillance, products classification, and intelligent retrieval or recommendation.

To raise the interest in reading commodity images, we introduce a dense and arbitrary-shaped text detection dataset named DAST1500, which contains 1538 images and 45,963 line-level detection annotations (including 7441 curved bounding boxes). The images are manually collected from the Internet and of size around 800×800. This dataset is multi-lingual, including mostly Chinese, and few English and Japanese texts. The images are divided as follows: 1038 images for training and 500 images for testing.

For these dense and arbitrary-shaped texts, we bound each text line by indefinite length point sets. To annotate each text instance, we use an outline polygon composed of a set of adjacent quadrilaterals to fit a curved line. The dataset is mainly annotated at line-level. For the case where the space between text parts exceeds the height of underlying text line, we divide the text line into separate text instances. This may decouple text extraction step from the impact of various layouts of commodity images. Some annotation examples are given in Fig. 5. This dataset DAST1500 is released by the following link: https://tianchi.aliyun.com/dataset/dataDetail?dataId=12084.

SynthText in the Wild [54]: SynthText is consist of 800k synthetic images generated by adding variants of text with random fonts, size, orientation, and color in natural images. Annotations are given in character, word, and line level. This dataset with word level annotation is used to pretrain the network.

ICDAR2015 Incidental Scene Text (IC15) [51]: This dataset is widely used to serve as a multi-oriented text benchmark. It was released for the Challenge 4 of ICDAR2015 Robust Reading Competition. Images in this dataset are taken by Google Glasses in an incidental manner without considerig text quality. Therefore, texts in this dataset exhibit large variations in scales, orientations, contrast, blurring, and viewpoint, making it more challenging for text detection. This dataset includes 1000 training images and 500 testing images. Annotations are provided with word-level bounding quadrilaterals.

MTWI [52]: MTWI dataset is a large web image dataset focusing on multi-type text, which varies in text font, text size, text layout and background. The texts are mainly English and Chinese texts. It contains 10,000 training images and 10,000 testing images. Annotations are given in text-line level considering the space between text instances.

SCUT-CTW1500 [23]: Different from classical multi-oriented text datasets, this dataset is dedicated to build a quite challenging dataset with curved texts. It consists of 1000 training images and 500 testing images. This dataset has more than 10k text annotations and every image at least contains one curved text. Each text instance is labeled by a polygon with 14 points. The annotation is given in line-level, including straight and curved text line.

TotalText [34]: Total-Text dataset also aims at raising the issue of arbitrary-shaped text detection. It contains 1555 scene images, divided into 1255 training images and 300 testing images. This dataset contains many curved and multi-oriented texts. Annotations are given in word level with polygon-shaped bounding boxes, and the number of the vertex is not fixed.

We use the associated evaluation protocol of each dataset if it is provided, otherwise, we use the standard PASCAL VOC evaluation protocol to evaluate the performance of the proposed method ICG.

4.2. Implementation details

We first pretrain the proposed model on SynthText dataset, then fine-tune it on each target dataset. The hyper-parameters α involved in Eq. (8), β in Eq. (9), and λ_1 and λ_2 in Eq. (10) are all set to 1 for all experiments. The network is optimized by the standard SGD algorithm with a momentum of 0.9. For pretraining and fine-tuning, images are resized to 384×384 and 512×512 respectively after random cropping. Batch size is set to 16. In pretraining, the learning rate is set to 10^{-3} for the first 60k iterations, then decayed to 10^{-4} for the rest 30k iterations. During fine-tuning, the learning rate is fixed to 10^{-4}. For the first 10-20k iterations of finetuning, the instance-aware weight described in Section 3.2 is set to 1. Then we calculate instance-aware weight as described in Section 3.2 for every iteration and apply the weight to the instance-aware loss in Eq. (10). This training step lasts 5-10K iterations, which depends on the size of the dataset. The hyper-parameters involved in the inference process text score threshold t_s and linking edge weight threshold t_l are decided with grid search on a validation set. In the training stage with instance-aware loss, these two parameters t_s and t_l are chosen based on the initially trained model without

Fig. 5. Some annotation examples of DAST1500.

instance-aware scheme. During inference and post-processing, as described in Section 3.6, we filter out polygons whose heights are below 10 and areas are smaller than 300. The proposed method is implemented with Tensorflow. All the experiments are carried out on a workstation with a Tesla P100 GPU.

4.3. Ablation study on DAST1500

To demonstrate the effect of proposed instance-aware component grouping framework, we evaluate several variants of the proposed ICG on DAST1500 dataset. These variants are summarized as below:

- **Baseline:** the proposed method without introducing repulsive links and instance-aware loss. This variant is very similar with SegLink. The difference is that the negative links for this variant is limited to the edges between text pixels and non-text pixels (*i*.e., background) and we adopt the filtering strategy and polygon NMS described in Section 3.6.
- **Baseline+ ins.-aware loss:** the proposed method trained with instance-aware loss, but without repulsive links.
- **Baseline+ att/rep links:** the proposed method using both attractive and repulsive links, but without instance-aware loss.
- **Baseline+ att/rep links + ins-aware loss (ICG):** the proposed method with attractive and repulsive links and instance-aware loss, named ICG.

The proposed method is dedicated for dense and arbitrary-shaped scene text detection, which is quite challenging. We conduct ablation study on this dataset. In the testing phrase, images of DAST1500 are resized to 768×768. For the proposed ICG, the two parameters t_s and t_l described in Section 3.6 are set to 0.5 and 0.45, respectively. We adopt the *approxpolyDP* funciton in OpenCV to extract polygons. The ablation study result is depicted in Table 1.

Effect of instance-aware loss We first assess the effectiveness of the proposed instance-aware loss in detecting dense and arbitrary-shaped scene texts. As depicted in Table 1, such instance-aware loss boosts the performance of baseline by 4.6%, showing the effectiveness of the proposed instance-aware loss.

Effect of attractive and repulsive links Compared with the Baseline model, the proposed attractive and repulsive links are very effective in handling dense and arbitrary-shaped scene text detection. Specifically, the proposed attractive and repulsive links boost the performance of baseline model by 8.7 percents.

Combining both instance-aware loss and attractive/repulsive links further boosts the performance to 0.794.

We also compare the proposed method ICG with other state-of-the-art methods including three recent multi-oriented text detectors: TextBoxes++ [19], EAST [32], and RRD [20], and three arbitrary-shaped text detectors: SegLink [25], PixelLink [36], and CTD+TLOC [23]. For the three multi-oriented methods, we use oriented rectangular bounding boxes of text regions as the ground-truth annotations. Concerning SegLink and PixelLink, we make necessary adjustments on ground-truth generation and extract polygons with *approxpolyDP*, and for CTD+TLOC, we transfer the annotation format of DAST1500 to that of CTW1500. The comparison is given in Table 1. Note that the results for the other state-of-the-art methods are reproduced based on their open source codes and trained on DAST1500. As depicted in Table. 1, the proposed ICG significantly outperforms other methods on DAST1500. Specifically, the three recent multi-oriented methods TextBoxes++, EAST, and RRD do not achieve good results on DAST1500. This is as expected since they are not suitable for curved text detection. Compared to SegLink and CTD+TLOC, the improvement in terms of F-measure is 14.1 and 12.8 percents, respectively. The proposed ICG also outperforms PixelLink by 4.7 percents. Yet, as depicted in Fig. 1, Pixellink has difficulty in accurately detect the whole text intances, which may pose problem for the following recognition. Some qualitative results of the proposed method ICG on DAST1500 are shown in Figs. 1 and 6.

4.4. Results on multi-oriented text detection

We conduct experiments on two multi-oriented text datasets: ICDAR15 and MTWI to further demonstrate that the proposed method ICG also performs well on multi-oriented scene text detection.

ICDAR2015 Incidental Scene Text ICDAR15 is a dataset of complicated background and the text size is small. During inference, images are resized to 1280×768, and t_s and t_l are set to 0.55 and 0.85, respectively. Some qualitative detection results on this dataset are given in Fig. 7(a). The quantitative evaluation compared with other methods are depicted in Table 2. The proposed ICG boosts a lot the original SegLink, and is also very competitive with other methods using the VGG16 backbone under a single scale test setting.

MTWI MTWI contains multi-lingual texts with significant variations in size, shape, and font type. We perform experiments on MTWI to testify the generality of the proposed ICG on large dataset. During testing, we fixed the image size to 768×768 and the parameters t_s and t_l are set to 0.4 and 0.65, respectively. For comparison, we conducted experiments using open source code of TextBoxes++ [19], PixelLink [36], EAST [32], and SegLink [25]. The quantitative results is depicted in Table. 3, showing that the proposed ICG significantly outperforms TextBoxes++ and PixelLink by 11.0% and 10.6%, respectively. When compared to EAST and SegLink, the improvement is 4.4% and 4.5%, respectively. Though the MTWI dataset does not contain a lot of close text instances, the proposed ICG still achieves 1.5% performance gain comparing to baseline. Some qualitative results are illustrated in Fig. 7(b).

Table 1
Ablation study and quantitative comparison with other state-of-the-art methods on DAST1500 dataset.

Models	recall	precision	f-measure
TextBoxes++* [19]	0.409	0.673	0.509
RRD* [20]	0.438	0.672	0.530
EAST* [32]	0.557	0.700	0.620
SegLink* [25]	0.647	0.660	0.653
CTD+TLOC* [23]	0.608	0.738	0.666
PixelLink* [36]	0.750	0.745	0.747
Baseline	0.673	0.712	0.692
Baseline+ins.-aware loss	0.690	0.795	0.738
Baseline+att/rep links	0.763	0.795	0.779
Baseline+att/rep links+ins.-aware loss (ICG)	0.792	**0.796**	**0.794**

* indicates reproduced results from open source code trained on DAST1500.

Fig. 6. Some qualitative results of the proposed ICG on DAST1500 dataset.

Fig. 7. Some qualitative results of the proposed method ICG on ICDAR15 in (a), MTWI in (b), CTW1500 in (c), and TotalText in (d).

4.5. Results on curved text detection

To further demonstrate the capability of the proposed ICG in detecitng arbitrary-shaped scene texts, we also conduct experiments on two publicly available curved text detection datasets: TotalText [34] and SCUT-CTW1500 [23].

TotalText This dataset contains both curved texts and oriented texts, which are annotated in word level. Image size varies a lot in this dataset. In testing, the short side of images is resized to 768 while keeping the original ratio between height and width. The threshold parameters t_s and t_l are set to 0.6 and 0.45, respectively. The quantitative results are given in Table 4. The proposed ICG achieves 81.5% in F-measure, surpassing the state-of-the-art result by 3.1% on this dataset. The proposed attractive and repulsive links as well as the instance-aware loss brings 6.9 percents performance gain than the baseline method. This demonstrates that the proposed ICG can also handle well word-level arbitrary-shaped text deteciton. Some qualitative results are shown in Fig. 7(c).

SCUT-CTW1500 We also evaluate the proposed ICG on SCUT-CTW1500 whose annotation is given in text-line level such that

Table 2
Quantitative results of different methods evaluated on ICDAR15.

Models	recall	precision	f-measure	FPS
Zhang et al. [28]	0.430	0.708	0.536	0.48
SegLink [25]	0.768	0.731	0.750	–
MCN [27]	0.800	0.720	0.760	–
EAST [32]	0.728	0.795	0.764	6.52
SSTD [16]	0.730	0.800	0.770	7.7
RRPN [18]	0.730	0.820	0.770	–
ITN [21]	0.741	0.857	0.795	–
EAST† [32]	0.735	0.836	0.782	**13.2**
Lyu et al. [26]	0.707	**0.941**	0.807	3.6
He et al.† [33]	0.800	0.820	0.810	1.1
TextBoxes++ [19]	0.767	0.872	0.817	11.6
RRD [20]	0.790	0.856	0.822	6.5
TextSnake [40]	0.804	0.849	0.826	1.1
PixelLink [36]	**0.820**	0.855	**0.837**	3.0
Baseline (Ours)	0.737	0.863	0.795	9.5
ICG (Ours)	0.803	0.837	0.820	7.1

† indicates that the backbone network is not VGG16.

Table 3
Quantitative results of different methods evaluated on MTWI.

Models	recall	precision	f-measure
TextBoxes++* [19]	0.563	0.668	0.611
PixelLink* [36]	0.635	0.596	0.615
EAST* [32]	0.612	0.758	0.677
SegLink* [25]	0.654	0.700	0.676
Baseline (Ours)	0.648	**0.776**	0.706
ICG (Ours)	**0.697**	0.747	**0.721**

* means the result is reproduced with open source code.

Table 4
Quantitative results of different methods evaluated on Totaltext.

Models	recall	precision	f-measure
Ch'ng et al. [34]	0.400	0.330	0.360
CTD+TLOC [23]	0.710	0.740	0.730
TextSnake [40]	0.745	**0.827**	0.784
Baseline (Ours)	0.727	0.767	0.746
ICG (Ours)	**0.809**	0.821	**0.815**

a complete sentence is annotated as a single polygon. The short side of images in this dataset is resized to 512 while keeping the original ratio between height and width. The threshold parameters t_s and t_l are set to 0.6 and 0.6, respectively. The quantitative results are shown in Table 5. The proposed ICG can also precisely

Table 5
Quantitative results of different methods evaluated on SCUT-CTW1500.

Models	recall	precision	f-measure
CTPN* [24]	0.538	0.604	0.569
EAST* [32]	0.491	0.787	0.604
CTD [23]	0.652	0.743	0.695
CTD+TLOC [23]	0.698	0.774	0.734
TextSnake [40]	**0.853**	0.679	0.756
Baseline (Ours)	0.785	0.816	0.800
ICG (Ours)	0.798	**0.828**	**0.813**

* indicates the result are obtained from [23].

detect arbitrary-shaped text in text-line level. The proposed ICG significantly outperforms the state-of-the-art methods on SCUT-CTW1500 by 5.7% and achieves 81.3% in f-measure. The performance gain with respect to the baseline model is not as significant as that on Total-Text. This is probably because that this dataset is annotated in text-line level, having less close texts cases than that for the word-level annotations in Total-Text. Some qualitative results are given in Fig. 7(d).

4.6. Runtime

The proposed ICG first extracts text components and predicts attractive and repulsive weights via the proposed network, followed by a post-processing based on modified minimum spanning tree. The runtime could be divided into two parts: network inference and post-processing. The inference using VGG16 backbone takes about 90ms for a 1280×768 IC15 image. For post-processing, the main part is the modified minimum spanning tree, which takes about 30ms. The total inference time is about 140ms. As depicted in Table 2, the proposed method ICG runs at 7.1FPS using VGG16 backbone. Note that the code for post-processing based on modified minimum spanning tree (currently in Python) is not optimized yet.

4.7. Weakness

Though the proposed ICG can handle well arbitrary-shaped text detection in most cases. It still fails for some difficult cases such as text-line whose direction is hard to identify and very small texts. Examples are presented in Fig. 8. Note that they are also difficult for other state-of-the-art methods.

Fig. 8. Some failure examples: ground-truths are depicted in red, detections are depicted in green. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

5. Conclusion

In this paper, we propose an instance-aware component grouping framework (ICG) for dense and arbitrary-shaped scene text detection. The proposed ICG consists of attractive and repulsive link estimation between text components and an instance-aware loss to force the network training to focus more on difficult text areas. Explicitly learning repulsive links between text components helps to separate close text instances, alleviating the major issue of separating close texts faced by most bottom-up methods, especially for very dense texts. We also introduce a dense and arbitrary-shaped scene text detection dataset of commodity images (DAST1500) to demonstrate the effectiveness of the proposed method. The proposed ICG can boost the performance on DAST1500 by a large margin. It also outperforms state-of-the-art methods on curved text detection and is very competitive with other methods on multi-oriented text detection. In the future, we plan to explore the end-to-end system for robust arbitrary-shaped text detection and recognition.

Acknowledgments

This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFB1004600, in part by the National Natural Science Foundation of China under Grant 61703171, and in part by the Natural Science Foundation of Hubei Province of China under Grant 2018CFB199. This work was also supported by Alibaba Group through Alibaba Innovative Research (AIR) program. The work of Y. Xu was supported by the Young Elite Scientists Sponsorship Program by CAST. The work of X. Bai was supported by the National Program for Support of Top-Notch Young Professionals and in part by the Program for HUST Academic Frontier Youth Team.

References

[1] Q. Ye, D. Doermann, Text detection and recognition in imagery: a survey, IEEE Trans. Pattern Anal. Mach. Intell. 37 (7) (2015) 1480–1500.
[2] K. Wang, S. Belongie, Word spotting in the wild, in: Proc. of European Conference on Computer Vision, 2010, pp. 591–604.
[3] Y.-F. Pan, X. Hou, C.-L. Liu, et al., A hybrid approach to detect and localize texts in natural scene images, IEEE Trans. Image Process. 20 (3) (2011) 800–813.
[4] L. Neumann, J. Matas, Real-time scene text localization and recognition, in: Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition, 2012, pp. 3538–3545.
[5] B. Bai, F. Yin, C.L. Liu, Scene text localization using gradient local correlation, in: Proc. of International Conference on Document Analysis and Recognition, 2013, pp. 1380–1384.
[6] W. Huang, Z. Lin, J. Yang, J. Wang, Text localization in natural images using stroke feature transform and text covariance descriptors, in: Porc. of IEEE Intl. Conf. on Computer Vision, 2013, pp. 1241–1248.
[7] W. Huang, Y. Qiao, X. Tang, Robust scene text detection with convolution neural network induced mser trees, in: Proc. of European Conference on Computer Vision, 2014, pp. 497–511.
[8] X.-C. Yin, W.-Y. Pei, J. Zhang, H.-W. Hao, Multi-orientation scene text detection with adaptive clustering, IEEE Trans. Pattern Anal. Mach. Intell. 37 (9) (2015) 1930–1937.
[9] S. Lu, T. Chen, S. Tian, J.-H. Lim, C.-L. Tan, Scene text extraction based on edges and support vector regression, Int. J. Doc. Anal.Recognit. 18 (2) (2015) 125–135.
[10] S. Tian, Y. Pan, C. Huang, S. Lu, K. Yu, C. Lim Tan, Text flow: a unified text detection system in natural scene images, in: Porc. of IEEE Intl. Conf. on Computer Vision, 2015, pp. 4651–4659.
[11] J. Matas, O. Chum, M. Urban, T. Pajdla, Robust wide-baseline stereo from maximally stable extremal regions, Image Vision Comput. 22 (10) (2004) 761–767.
[12] B. Epshtein, E. Ofek, Y. Wexler, Detecting text in natural scenes with stroke width transform, in: Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition, 2010, pp. 2963–2970.
[13] M. Jaderberg, K. Simonyan, A. Vedaldi, A. Zisserman, Reading text in the wild with convolutional neural networks, Int. J. Comput. Vision 116 (1) (2016) 1–20.
[14] Y. Liu, L. Jin, Deep matching prior network: toward tighter multi-oriented text detection, in: Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition, 2017, pp. 3454–3461.
[15] M. Liao, B. Shi, X. Bai, X. Wang, W. Liu, Textboxes: a fast text detector with a single deep neural network., in: Proc. of the AAAI Conf. on Artificial Intelligence, 2017, pp. 4161–4167.
[16] P. He, W. Huang, T. He, Q. Zhu, Y. Qiao, X. Li, Single shot text detector with regional attention, in: Porc. of IEEE Intl. Conf. on Computer Vision, 2017, pp. 3047–3055.
[17] H. Hu, C. Zhang, Y. Luo, Y. Wang, J. Han, E. Ding, Wordsup: exploiting word annotations for character based text detection, in: Porc. of IEEE Intl. Conf. on Computer Vision, 2017, pp. 4950–4959.
[18] J. Ma, W. Shao, H. Ye, L. Wang, H. Wang, Y. Zheng, X. Xue, Arbitrary-oriented scene text detection via rotation proposals, IEEE Trans. Multimedia 20 (11) (2018) 3111–3122.
[19] M. Liao, B. Shi, X. Bai, Textboxes++: a single-shot oriented scene text detector, IEEE Trans. Image Process. 27 (8) (2018) 3676–3690.
[20] M. Liao, Z. Zhu, B. Shi, G. Xia, X. Bai, Rotation-sensitive regression for oriented scene text detection, in: Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition, 2018, pp. 5909–5918.
[21] F. Wang, L. Zhao, X. Li, X. Wang, D. Tao, Geometry-aware scene text detection with instance transformation network, in: Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition, 2018, pp. 1381–1389.
[22] P. Lyu, M. Liao, C. Yao, W. Wu, X. Bai, Mask textspotter: an end-to-end trainable neural network for spotting text with arbitrary shapes, in: Proc. of European Conference on Computer Vision, 2018, pp. 67–83.
[23] Y. Liu, L. Jin, S. Zhang, C. Luo, S. Zhang, Curved scene text detection via transverse and longitudinal sequence connection, Pattern Recognit. (2019).
[24] Z. Tian, W. Huang, T. He, P. He, Y. Qiao, Detecting text in natural image with connectionist text proposal network, in: Proc. of European Conference on Computer Vision, 2016, pp. 56–72.
[25] B. Shi, X. Bai, S. Belongie, Detecting oriented text in natural images by linking segments, in: Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition, 2017, pp. 3482–3490.
[26] P. Lyu, C. Yao, W. Wu, S. Yan, X. Bai, Multi-oriented scene text detection via corner localization and region segmentation, in: Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition, 2018, pp. 7553–7563.
[27] Z. Liu, G. Lin, S. Yang, J. Feng, W. Lin, W.L. Goh, Learning markov clustering networks for scene text detection, in: Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition, 2018, pp. 6936–6944.
[28] Z. Zhang, C. Zhang, W. Shen, C. Yao, W. Liu, X. Bai, Multi-oriented text detection with fully convolutional networks, in: Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition, 2016, pp. 4159–4167.
[29] C. Yao, X. Bai, N. Sang, X. Zhou, S. Zhou, Z. Cao, Scene text detection via holistic, multi-channel prediction, arXiv preprint arXiv:1606.09002 (2016).
[30] Y. Wu, P. Natarajan, Self-organized text detection with minimal post-processing via border learning, in: Porc. of IEEE Intl. Conf. on Computer Vision, 2017, pp. 5010–5019.
[31] D. He, X. Yang, C. Liang, Z. Zhou, G. Alexander, I. Ororbia, D. Kifer, C.L. Giles, Multi-scale fcn with cascaded instance aware segmentation for arbitrary oriented word spotting in the wild., in: Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition, 2017, pp. 474–483.
[32] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, J. Liang, East: an efficient and accurate scene text detector, in: Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition, 2017, pp. 2642–2651.
[33] W. He, X.-Y. Zhang, F. Yin, C.-L. Liu, Deep direct regression for multi-oriented scene text detection, in: Porc. of IEEE Intl. Conf. on Computer Vision, 2017, pp. 745–753.
[34] C.K. Ch'ng, C.S. Chan, Total-text: a comprehensive dataset for scene text detection and recognition, in: Proc. of International Conference on Document Analysis and Recognition, 1, 2017, pp. 935–942.
[35] S. Tian, S. Lu, C. Li, Wetext: scene text detection under weak supervision, in: Porc. of IEEE Intl. Conf. on Computer Vision, 2017, pp. 1492–1500.
[36] D. Deng, H. Liu, X. Li, D. Cai, Pixellink: detecting scene text via instance segmentation, in: Proc. of the AAAI Conf. on Artificial Intelligence, 2018.
[37] C. Xue, S. Lu, F. Zhan, Accurate scene text detection through border semantics awareness and bootstrapping, in: Proc. of European Conference on Computer Vision, 2018, pp. 355–372.
[38] T. He, Z. Tian, W. Huang, C. Shen, Y. Qiao, C. Sun, An end-to-end textspotter with explicit alignment and attention, in: Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition, 2018, pp. 5020–5029.
[39] F. Zhan, S. Lu, C. Xue, Verisimilar image synthesis for accurate detection and recognition of texts in scenes, in: Proc. of European Conference on Computer Vision, 2018, pp. 249–266.
[40] S. Long, J. Ruan, W. Zhang, X. He, W. Wu, C. Yao, Textsnake: a flexible representation for detecting text of arbitrary shapes, in: Proc. of European Conference on Computer Vision, 2018, pp. 20–36.
[41] L. Sun, Q. Huo, W. Jia, K. Chen, A robust approach for text detection from natural scene images, Pattern Recognit. 48 (9) (2015) 2906–2920.
[42] V. Khare, P. Shivakumara, P. Raveendran, M. Blumenstein, A blind deconvolution model for scene text detection and recognition in video, Pattern Recognit. 54 (2016) 128–148.
[43] P. Shivakumara, R. Raghavendra, L. Qin, K.B. Raja, T. Lu, U. Pal, A new multi-modal approach to bib number/text detection and recognition in marathon images, Pattern Recognit. 61 (2017) 479–491.
[44] L. Gómez, D. Karatzas, Textproposals: a text-specific selective search algorithm for word spotting in the wild, Pattern Recognit. 70 (2017) 60–74.
[45] B.B. Chaudhuri, C. Adak, An approach for detecting and cleaning of struck-out handwritten text, Pattern Recognit. 61 (2017) 282–294.
[46] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: towards real-time object detection with region proposal networks, IEEE Trans. Pattern Anal. Mach. Intell. (6) (2017) 1137–1149.

[47] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A.C. Berg, Ssd: single shot multibox detector, in: Proc. of European Conference on Computer Vision, 2016, pp. 21–37.
[48] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: unified, real-time object detection, in: Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition, 2016, pp. 779–788.
[49] Y. Xu, Y. Wang, W. Zhou, Y. Wang, Z. Yang, X. Bai, Textfield: learning a deep direction field for irregular scene text detection, IEEE Trans. Image Process. (2019). To appear, doi: 10.1109/TIP.2019.290058.
[50] S. Wolf, C. Pape, A. Bailoni, N. Rahaman, A. Kreshuk, U. Kothe, F. Hamprecht, The mutex watershed: Efficient, parameter-free image partitioning, in: Proc. of European Conference on Computer Vision, 2018, pp. 546–562.
[51] 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 International Conference on Document Analysis and Recognition, 2015, pp. 1156–1160.
[52] M. He, Y. Liu, Z. Yang, S. Zhang, C. Luo, F. Gao, Q. Zheng, Y. Wang, X. Zhang, L. Jin, Icpr2018 contest on robust reading for multi-type web images, in: Proc. of Intl. Conf. on Pattern Recognition, 2018, pp. 7–12.
[53] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, CoRR (2014). arXiv: abs/1409.1556.
[54] A. Gupta, A. Vedaldi, A. Zisserman, Synthetic data for text localisation in natural images, in: Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition, 2016, pp. 2315–2324.

Jun Tang received his B.S. degree from the School of Optical and Electronic Information, Huazhong University of Science and Technology(HUST), Wuhan, China, in 2017. He is currently pursuing the masters degree in School of Electronics Information and Communications, Huazhong University of Science and Technology (HUST), Wuhan, China. His main research interests include object detection and scene text detection.

Zhibo Yang received his B.S. degree from the Department of Automation, Harbin Institute of Technology, Harbin, China, in 2010, and M.S. degree from the Department of Automation, Beijing, China, in 2014. He is currently a Senior Algorithm Engineer in the TaoBao Technology Department, Alibaba, Hangzhou, China. His research interests include object detection and text detection in Images/videos.

Yongpan Wang received her B.S. degree from Sichuan University, Sichuan, China, in 2007 and M.S degree from Zhejiang University, Zhejiang, China, in 2010. She is currently a Senior Algorithm Specialist in the TaoBao Technology Department, Alibaba, Hangzhou, China. She is the leader of DuGuang, which is an OCR platform of Alibaba. Her interests focus on OCR, document analysis, deep learning and algorithm efficiency optimization.

Qi Zheng received his B.S. degree from Shanghai Jiao Tong University, Shanghai, China, in 2008, and M.S. degree from Shanghai Jiao Tong University, Shanghai, China, in 2011. Hi is currently an Algorithm Specialist in Taobao Technology Department, Alibaba, Hangzhou, China. His interests include OCR and document analysis.

Yongchao Xu received in 2010 both the engineer degree in electronics & embedded systems at Polytech Paris Sud and the master degree in signal processing & image processing at Université Paris Sud, and the Ph.D. degree in image processing and mathematical morphology at Université Paris Est in 2013. He is currently an Associate Professor with the School of Electronic Information and Communications, HUST. His research interests include mathematical morphology, image segmentation, medical image analysis, and deep learning.

Xiang Bai received the B.S., M.S., and Ph.D. degrees from the Huazhong University of Science and Technology (HUST), Wuhan, China, in 2003, 2005, and 2009, respectively, all in electronics and information engineering. He is currently a Professor with the School of Electronic Information and Communications, HUST. He is also the Vice-director of the National Center of Anti-Counterfeiting Technology, HUST. His research interests include object recognition, shape analysis, scene text recognition and intelligent systems.