Fuzzy Semantics for Arbitrary-shaped Scene Text Detection

Fangfang Wang, Xiaogang Xu, Yifeng Chen, Xi Li†

***Abstract*—To robustly detect arbitrary-shaped scene texts, bottom-up methods are widely explored for their flexibility. Due to the highly homogeneous texture and cluttered distribution of scene texts, it is nontrivial for segmentation-based methods to discover the separatrixes between adjacent instances. To effectively separate nearby texts, many methods adopt the seed expansion strategy that segments shrunken text regions as seed areas, and then iteratively expands the seed areas into intact text regions. In seek of a more straightforward way that does not rely on seed area segmentation and avoid possible error accumulation brought by iterative processing, we propose a redundancy removal strategy. In this work, we directly explore two types of fuzzy semantics—text and separatrix—that do not possess specific boundaries, and separate cluttered instances by excluding the separatrix pixels from text regions. To deal with the fuzzy semantic boundaries, we also conduct reliability analysis in both optimization and inference stage to suppress false positive pixels at ambiguous locations. Experiments on benchmark datasets demonstrate the effectiveness of our method.**

***Index Terms*—Arbitrary-shaped Text Detection, Fuzzy Semantics, Segmentation-based Framework, Single-shot Network.**

I. Introduction

ARBITRARY-SHAPED scene text detection aims to accurately locate tight text regions of arbitrary shapes from natural scene images. It has wide-range applications such as text recognition, scene parsing and automatic pilot. The main challenge of robust scene text detection lies in the complex appearance of texts, such as arbitrary shape, skewed viewpoint and large aspect ratio.

To deal with the arbitrary shapes, mainstream methods seek bottom-up solutions for their flexibility and treat text detection as a segmentation problem. However, as mixtures of stroke and background pixels, text regions are highly homogeneous textures that do not possess natural and clear boundaries. Besides, as shown in Figure 1 (a), scene texts are often in cluttered distribution and sometimes even contiguous due to the coarse polygon annotations. Thus, effectively separating cluttered instances becomes the most intractable problem in segmentation-based methods. False positive pixels along the instance separatrix areas often merge adjacent instances, which can have a dramatic influence on the detection results even though these pixels are of a very small proportion in the whole image. A typical solution is two-stage processing [2], [5], [11] which avoids directly discovering text separatrixes. They tend to segment shrunken text regions at first to find separated instance seeds, and then expand these seed areas iteratively and exhaustively to recover the intact text regions. Though the seed area extraction and iterative region expansion strategy can help separate cluttered instances, its performance is highly relied on the seed area segmentation, and error may accumulate throughout the iterative expansion procedure. Given that, we seek a more straightforward strategy to discover the specific instance separatrixes by directly modeling its unique semantics.

Fig. 1. Demonstration of the fuzzy semantics in scene text image. (a) demonstrates the cluttered instances; (b) visualizes the instance separatrixes; (c) shows the separated instances with clear and intact separatrixes; and the red points in (d) are semantically ambiguous pixels. Different colors represent different semantic instances.

As shown in Figure 1 (b), the separatrixes between adjacent text instances are unique and recognizable areas bounded by contrastive textures. Intrinsically, instance separatrixes share most of their boundaries with contiguous texts along the fuzzy lines where texture changes. In this work, we define text and separatrix whose boundaries are fuzzy as two fuzzy semantic categories, and acquire text instances from a redundancy removal perspective that excludes separatrix pixels from text

Fangfang Wang is with the College of Computer Science and Technology, Zhejiang University, and Zhejiang Lab, Hangzhou, 310027, China. (E-mail: wangff@zhejianglab.com).

Xiaogang Xu is with Zhejiang Lab, Hangzhou, 310027, China. (E-mail: xxgang2013@163.com).

Yifeng Chen and Xi Li are with the College of Computer Science and Technology, Zhejiang University, Hangzhou, 310027, China. (E-mail: {yifengchen, xilizju}@zju.edu.cn).

Corresponding author: Xi Li.

regions, as illustrated in Figure 1 (c). Unlike common semantic categories, the two fuzzy semantic categories of text and separatrix are not mutually exclusive. For example, as shown in Figure 1 (d), it is reasonable and intuitive to consider pixels near the fuzzy boundaries as both text and separatrix. So when exploring the fuzzy semantics, reliability analysis is needed to decide the textness of these semantically ambiguous locations.

Based on the above analysis, we propose a segmentation-based method which directly and independently models two types of fuzzy semantics, which are text and separatrix, and discover text instances by taking both the pixel categorization and its reliability into consideration. Specifically, we design an end-to-end framework that densely conduct multi-label categorization and reliability prediction, simultaneously. The necessity of conducting multi-label categorization lies in that the two fuzzy semantics are sometimes overlapping, which means the probabilities of a pixel being text or separatrix are independent against each other. So instead of carrying out single-label categorization like common semantic segmentation task, we devise two independent branches for text and separatrix segmentation. Further more, two additional branches are adopted for reliability prediction for text and separatrix, respectively. In our work, we intuitively define the reliability on the basis of the distance transformation from a pixel to its nearest semantic boundary. Consequently, the nearer a pixel is to the semantic boundary, the less reliable its categorization result is. During optimization, to fully explore the fuzzy semantics, we employ cross-image normalized focal loss with the guidance of reliability to balance the extremely biased negative and positive ratio of pixel samples and alleviate the impact of less reliable pixels. In the inference stage, pixels are re-scored according to their reliability to solve the semantic competition between text and separatrix. The detection results are generated by extracting connected components on the final textness map.

The proposed method is advantageous in the following aspects. First, directly modeling two types of fuzzy semantics including text and separatrix makes a more simple and intuitive approach to separate instances in segmentation-based text detection frameworks. Second, our method is flexible that does not rely on seed areas which are fixed throughout their following post-processings. And third, our one-step redundancy removal strategy avoids the possible error accumulation from iterative or multi-step region recovering. The main contributions of this paper are as follows:

- we solve arbitrary-shaped scene text detection from a redundancy removal perspective by acknowledging and exploring two types of fuzzy semantics, which are text and instance separatrix in natural scene images;
- we propose to specify the intact and clear boundaries between cluttered text instances by dealing the competition between fuzzy semantics with reliability analysis;
- we present a straightforward segmentation-based scene text detection framework that takes connected component extraction as the final step without any iterative or complex post-processings.

II. Related Work

Scene text detection has been widely studied over the past few years. Many methods are inspired by successful object detection and segmentation frameworks, and thus scene text detection methods can be mainly divided into three categories: detection-based methods, segmentation-based methods and component-based methods which combines techniques from both detection and segmentation.

*a) **Detection-based methods**:* Detection-based methods treat texts as a special type of object. They usually seek top-down approaches that directly detect whole text instances. Some methods try to provide more appropriate and representative features for text extraction. For example, [12] manipulates local feature maps with discrete settings to conduct geometry normalization and enhances the geometric perceptive ability of features. Differently, instead of manipulating features, [13] rotates the feature extracting convolutional kernels and applies it several times to generate multi-channel rotation-invariant features. Furthermore, [14] utilizes instance-level affine transformations to encode text shapes and deforms the sampling grids of convolutional kernels under the guidance of affine parameters to extract geometry-aware features. These methods can handle multi-oriented text detection well but are unscalable to arbitrary shapes. To conquer this, [15] proposes a proposal-based framework to sequentially extract text proposals and conduct variable-length coordinates regression with an on-top RNN. [16] and [17] explore parameterized approaches. [16] represents text shapes with Bezier curves while [17] parameterizes arbitrary contours with polynomial curve fitting under polar system.

*b) **Segmentation-based methods**:* Segmentation-based methods solve text detection from a bottom-up perspective. These methods [4], [7], [18], [19] explore pixel-level semantics to extract text regions. However, due to the highly cluttered distribution and homogeneous texture within text instances, it is hard for pixel-level categorization to accurately and comprehensively discover the boundaries between instances. Thus separating different instances is a key issue that many methods put their efforts on. [20] and [21] separate instances by extracting text proposals, and adopt two-stage pipelines that conduct character or contour segmentation on text proposals. [22] seeks affinity information between adjacent pixels to decide whether they are from the same instance. Another typical solution is to segment a seed area for each instance, which is most commonly the text centerlines (TCL), and then iteratively expand the seed area to recover the whole text region [2], [5], [8], [9], [11]. In this paper, we propose another segmentation-based method that handle scene text detection from a redundancy removal perspective.

*c) **Component-based methods**:* Similar to segmentation-based methods, component-based methods are also bottom-up approaches that decompose text instances into components of different granularity. These methods usually combine segmentation and regression techniques. For example, [23] segments position sensitive corner components and adopts corner coordinates regression as supportive cue for region composition. [24] and [25] decompose text instances into local boxes and predict the correlation between densely detected

local boxes. TextSnake [1] models text instances as a directed sequence of discs and utilize geometric properties like radius and connection between discs to recover the instance. More typically, like some segmentation-based methods, TCLs are segmented as seed areas to indicate the existence of instances. And then local components, such as boxes [26] or local offsets [3], [6] are extracted and regressed to expand the seed areas into intact instances.

III. Proposed Method

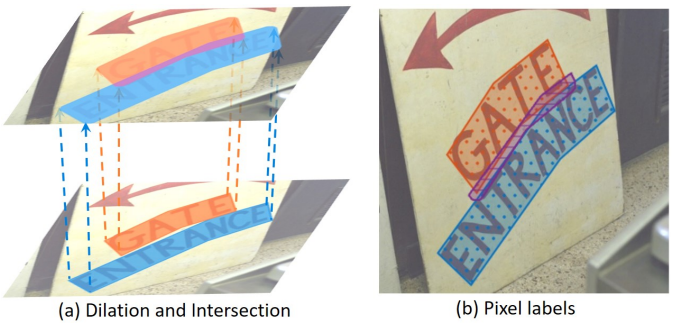
Detecting arbitrary scene texts through text region segmentation is widely adopted for its flexibility in dealing with various geometric layouts of targets. However, due to the cluttered distribution and highly homogeneous texture of texts, separating geographically close instances from segmentation maps is a nontrivial problem for segmentation-based text detection methods. In our work, we solve this problem from a straightforward redundancy removal perspective. In principle, we directly explore two types of fuzzy semantics, which are text and separatrix, in a multi-branch segmentation framework, and generate textness maps by solving the competition between text and separatrix under the assistance of reliability analysis.

A. Fuzzy Semantics for Text Detection

Text regions are highly homogeneous textures which are actually mixtures of stroke pixels and background pixels, resulting in fuzzy semantic regions that do not possess clear and specific boundaries. Due to the fuzziness of text boundaries, text regions are commonly represented by coarse polygons in current benchmark datasets, leading to frequently cluttered and contiguous text masks without clear separatrixes. So directly segmenting whole text regions without accidentally merging adjacent instances is difficult. To solve this problem, we seek a straightforward way that directly discover the separatrixes between adjacent text instances.

We acknowledge that similar to text, instance separatrix is also a unique and recognizable semantic category in natural scene text images which is bounded by the fuzzy texture boundaries of texts and structured as the narrow band between adjacent text instances. Particularly, in comparison with the whole borders of texts which usually include sides of different visual characteristics like isolated sides and those between nearby instances, separatrixes are more consistent in their visual appearances and thus make a more reasonable semantic category. So in our work, we propose to explore two types of fuzzy semantics including text regions and instance separatrixes, and extract text instances from a redundancy removal perspective by excluding separatrix pixels from text regions on the segmentation map.

With the given text regions, we define instance separatrix areas in an intuitive way. Naturally, if two text instances locate near enough, they produce a separatrix area. On the contrary, the wide gap between faraway instances is not a separatrix but common background. In practice, we obtain separatrixes by dilating text instances and labeling the intersected areas as separatrix areas. As illustrated in Figure 2, text contours



(a) Dilation and Intersection (b) Pixel labels

Fig. 2. Illustration of pixel labeling. In (a), instances are dilated and intersect with each other at separatrix area. (b) shows the pixel labeling results: dotted areas represent text regions while hatched area represents separatrix.

represented by polygons are dilated adaptively according to their scales. In this method, we set the dilation parameter that controls how nearby two instances should be to produce a separatrix area as $\frac{1}{5}$ of their line-heights. In particular, to prevent a small instance from being overwhelmed by its large neighbors during the dilation, we exclude the whole small instance from the intersected area and make sure the separatrix will not include more than $\frac{3}{10}$ pixels of any text instance.

Notably, the two fuzzy semantic categories are not mutually exclusive like categories in common semantic segmentation in both intuition and pixel labeling policy. Intuitively, as described in Figure 1 (d), pixels along the fuzzy boundaries of text and separatrix can reasonably be categorized as both. In pixel labeling, the coarsely annotated text regions contain separatrix pixels along its adjacent side with nearby instance. And the intersected separatrix areas can also sometimes include text pixels when the dilation parameter is larger than the natural width of their actual gap.

The fuzzy boundaries of text and separatrix bring about two consequences. Firstly, around their fuzzy boundaries, one pixel location can have both the labels of text and separatrix, which forms a multi-label classification problem. To this end, we conduct pixel-wise text/non-text and separatrix/non-separatrix segmentation on two separated branches, treating the probabilities of a pixel being text or separatrix independent against each other. Secondly, the actual textness of pixels labeled as both categories are ambiguous. To solve the competition between two fuzzy semantics at these ambiguous locations and alleviate the negative impact of miss-labeled data, reliability analysis is needed in both training and inference stages.

B. Reliability Analysis

As analyzed in Section III-A, the boundaries of text regions and instance separatrixes are fuzzy in both intuition and pixel labeling. Based on the prior that ambiguous pixel locations for each semantic category are most commonly around the semantic boundaries, we model the semantic reliability of a pixel according to its distance transformation to the nearest boundary, such that the nearer a pixel is to the semantic boundary the less reliable its categorization is.

Specifically, we adopt a truncated signed distance function [27] to define the reliability R_i of a pixel i:

$$R_i = \begin{cases} \frac{1}{T_d} max(T_d, dist(i, \mathcal{B})); & \text{if } y_i = 1; \\ -\frac{1}{T_d} max(T_d, dist(i, \mathcal{B})), & \text{otherwise,} \end{cases} \quad (1)$$

where $dist(i, \mathcal{B})$ measures the Euclidean distance between location i and its closest semantic boundary $\mathcal{B}$, y_i is the one-hot label at location i, T_d is the truncating threshold and also the normalization factor. We set $T_d = 10$ in the experiments. The absolute value of R_i indicate the reliability of pixel i, while the sign is to distinguish the semantic bias. We denote the reliability maps for text and separatrix segmentation tasks by $\mathbf{R}^t$ and $\mathbf{R}^s$, respectively. The reliability quantization is shown in Figure 3.

Less reliable pixels introduce noise to text and separatrix segmentation tasks, and the semantic scores at these ambiguous locations are more likely to be in a close match, causing uncertain textness. So the reliability analysis is applied in both training and inference stages. During training, the training weight of a pixel location is re-weighted according to the absolute value of its reliability to alleviate the influence of less reliable data, such that the weights of pixels in both foreground and background areas of high reliability are larger than those who are less reliable. The detailed usage is described in Section III-D. In the inference stage, the output text segmentation map $\mathbf{T}$ and separatrix segmentation map $\mathbf{S}$ are re-scored and fused to decide the final textness of all pixel locations denoted by $\mathbf{M}$:

$$\mathbf{T}' = \mathbf{T} + \lambda \bar{\mathbf{R}}^t; \quad (2)$$

$$\mathbf{S}' = \mathbf{S} + \lambda \bar{\mathbf{R}}^s; \quad (3)$$

$$\mathbf{M} = \mathbf{T}'(1 - \mathbf{S}'), \quad (4)$$

where λ is a weighting parameter to balance the re-scoring range, $\bar{\mathbf{R}}^t$ and $\bar{\mathbf{R}}^s$ are the ground-truth reliability maps, $\mathbf{T}'$ and $\mathbf{S}'$ are the final text and separatrix segmentation score maps. Consequently, highly reliable locations are basically classified as their winning semantic category, while the textness of ambiguous pixels tend to decrease so that clear separatrixes are more likely to be preserved.

So far, the redundancy removal strategy disentangle text and separatrix areas and then adopt the reliability analysis procedure to adaptively rescore the overlapped pixels and thus decide their final categories. The rationality and necessity of the reliability analysis lies in that any specific bounding of text areas and background areas is not in conformity with the fuzzy nature of their semantics and thus cannot surely decide the category of pixels along the boundaries. So compared with the entangled semantic modeling strategy of directly segmenting the separatrix-excluded text areas, the redundancy removal strategy is more intuitive and flexible.

C. Framework

As illustrated in Section III-A and III-B, we model text instances with two types of fuzzy semantics and their according reliability. In this section, we design an end-to-end segmentation framework to effectively explore the semantics and reliability.

Fig. 3. Demonstration of the reliability maps of text and separatrix. Left figures show the semantic instances and right figures visualize their corresponding reliability.

Our framework mainly consists of three parts: feature extraction, feature fusion, and multi-branch joint optimization. The overall architecture of our framework is illustrated in Figure 4. In feature extraction, we adopt ResNet50 [28] as our backbone network, and apply FPN [29] for multi-scale feature generation. Then we follow the feature fusion strategy adopted in SOLOv2 [30] to fuse multi-level feature maps. Specifically, feature maps from P2 to P5 are up-scaled to $\frac{1}{4}$ of the input image by repeated stages of 3×3 convolution, group normalization, ReLU activation and $2\times$ up-sampling, and fused by element-wise summation. Then the channel of the fused feature map is reduced from 256 to 128 with a 1×1 convolution followed by group normalization and ReLU activation. This reduced feature map is the input of four independent branches.

In our framework, text segmentation, text reliability regression, separatrix segmentation and separatrix reliability regression are conducted simultaneously with parallel branches, as shown in Figure 4. Each branch consists of three convolutional blocks, in which sequentially stack 3×3 convolution, group normalization and ReLU activation, and single channel logits are predicted at the ends of the branches. As described in Section III-B, during training, the ground-truth reliability maps will be involved in text and separatrix segmentation tasks.

In the inference stage, given an input image $\mathbf{I}$, our method predicts four score maps including text segmentation map $\mathbf{T}$, text reliability map $\mathbf{R}^t$, separatrix segmentation map $\mathbf{S}$ and separatrix reliability map $\mathbf{R}^s$, respectively. The four score maps are fused according to Equation 2-4 and generate the final textness map $\mathbf{M}$. Detections $\mathcal{D}$ are extracted from $\mathbf{M}$ by binarization and finding connected components:

$$\mathcal{D} = \text{find\_cc}(\mathbb{1}(\mathbf{M} > 0.5)), \quad (5)$$

where $\mathbb{1}(\cdot)$ is binary indicator function and $\text{find\_cc}(\cdot)$ is a function extracting contours of the connected components on

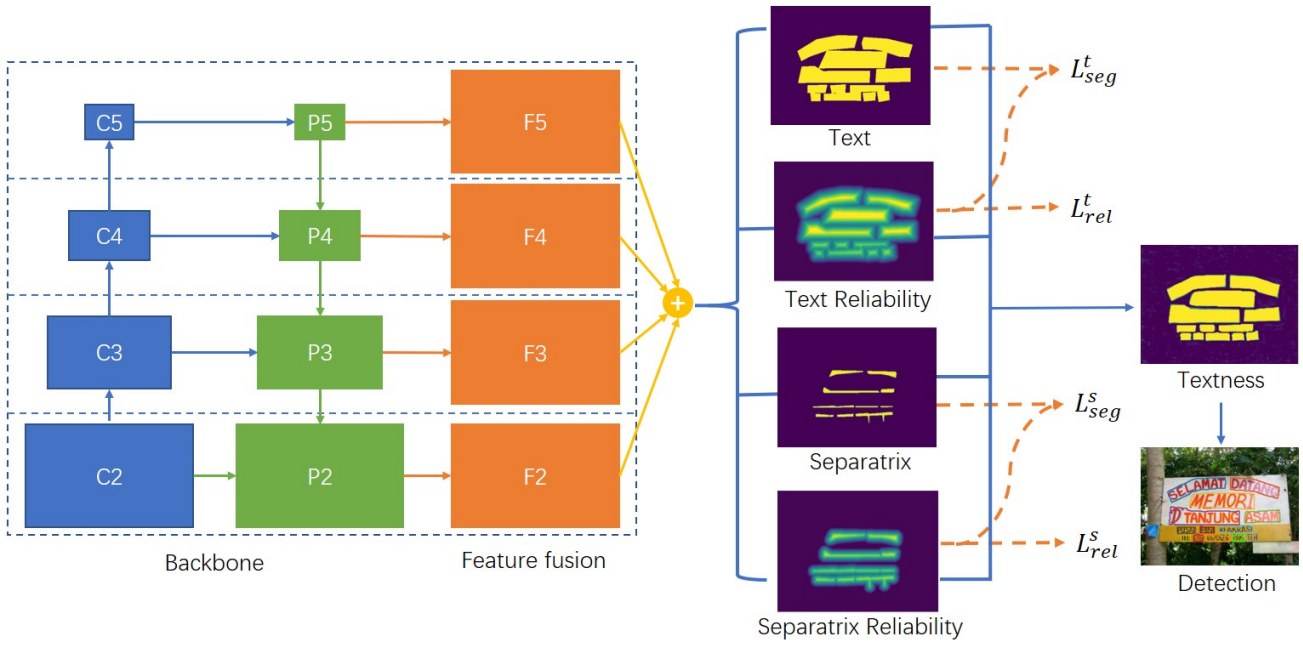


Fig. 4. Overall architecture of our framework. Four parallel branches conduct text segmentation, text reliability regression, separatrix segmentation and separatrix reliability regression, respectively. Orange dotted lines indicate training strategy, while blue lines depict the inference pipeline.

the binary mask. The confidence score of a detection is the average text segmentation score of all its inside pixels.

D. Optimization

Our method adopt joint optimization among the four segmentation and regression tasks. The overall loss function of our framework is as follows:

$$L = L^t_{seg} + L^s_{seg} + L^t_{rel} + L^s_{rel}, \tag{6}$$

where L^t_{seg} and L^s_{seg} are cross-image normalized focal losses with the guidance of their associated reliability, which will be described below. L^t_{rel} and L^s_{rel} are smooth L_1 losses [31], which optimizes the reliability regression tasks:

$$L_{rel}(\bar{R}_i, R_i) = \begin{cases} \frac{1}{2\beta}(\bar{R}_i - R_i)^2, & \text{if } |\bar{R}_i - R_i| < \beta; \\ |\bar{R}_i - R_i| - \frac{\beta}{2}, & \text{otherwise,} \end{cases} \tag{7}$$

where β is the threshold of the piece-wise smooth L_1 function and is set to $\frac{1}{9}$ in the experiments.

Cross-image normalized focal loss with guidance. Firstly, we revisit the standard focal loss [32], and then introduce the proposed cross-image normalized focal loss with the guidance of reliability utilized in our text and separatrix segmentation tasks.

The standard focal loss [32] is defined as:

$$FL(p_r) = -(1 - p_r)^\gamma \log(p_r), \tag{8}$$

where $p_r = p$ for positive samples and $p_r = 1 - p$ for negative ones. p is the predicted probability value of being the target class, the focusing parameter γ is set to 2 in the experiments.

Focal loss is often used to deal with the imbalance between positive and negative samples by reducing the total weights of background which includes a lot of easy negatives. The total weights of positive and negative samples are:

$$V^+ = \sum_{i \in \mathcal{P}} (1 - p_i)^\gamma; \quad V^- = \sum_{i \in \mathcal{N}} p_i^\gamma, \tag{9}$$

where p_i is the predicted probability at pixel i, $\mathcal{P}$ and $\mathcal{N}$ are sets of all positive and negative samples, respectively. The ideal value of $\frac{V^-}{V^+}$ should around 1, but it is still a large number when negative samples are far more than positive samples such as in the cases of text/non-text and separatrix/non-separatrix segmentation tasks. Moreover, the standard focal loss will decrease the overall weights of all the samples, which leads to lower optimization efficiency. For weighting ratio adjustment, [32] proposes an α-balanced variant of the focal loss which changes the weighting ratio to $\frac{1-\alpha_t}{\alpha_t}\frac{V^-}{V^+}$ with an extra hyper parameter α_t. To compensate for the overall weight decrease, using focal loss requires to empirically increase the loss weight or learning rate.

Instead of tuning the balancing parameter or loss weight, we propose a cross-image normalized focal loss to adaptively adjusting the value of weighting ratio $\frac{V^-}{V^+}$ during the training process. For simpler notations we take positive location i as an example, and the case for negative locations is deducible. The cross-image normalized focal loss at a positive location i is defined as:

$$NFL(p_i) = -\frac{U}{2V_f}(1 - p_i)^\gamma \log(p_i), \quad i \in \mathcal{P}, \tag{10}$$

where U is the number of all pixel samples in current image, and V_f is the cross-image running mean of positive weight summation computed from all seen images at time step f:

$$V_f = mV_{f-1} + (1 - m)V^+, \tag{11}$$

where V_{f-1} is the running mean at the last time step, $m = 0.9$ is the momentum, V^+ is the weights summation of positive samples in current image. In this way, the weights summation will be normalized to close to $\frac{U}{2}$ for both positive and negative samples.

As described in Section III-B, we additionally introduce the guidance of reliability into the cross-image normalized focal loss:

$$GNFL(p_i) = -\frac{U}{2V_f'}(1+\mu|R_i|)(1-p_i)^\gamma \log(p_i), \quad i \in \mathcal{P}, \tag{12}$$

where R_i is the current reliability, $\mu = 0.5$ in the experiments, V_f' is the running mean of V'^+:

$$V'^+ = \sum_{i\in\mathcal{P}} (1+|R_i|)(1-p_i)^\gamma. \tag{13}$$

IV. Experiments

A. Datasets

We evaluate our method on three standard arbitrary-shaped scene text detection datasets: SCUT-CTW1500 [33], Total-Text [34] and ICDAR-ArT [35].

SCUT-CTW1500. SCUT-CTW1500 [33] is an arbitrary-shaped scene text detection dataset which consists of 1000 training images and 500 test images. It includes multi-oriented, curved and irregular-shaped textline-level instances in the form of simple polygons annotated by 14 vertices.

TotalText. TotalText [34] is a challenging arbitrary text dataset that includes horizontal, multi-oriented, and curved texts. It consist of 1255 training images and 300 test images. Different from SCUT-CTW1500, the text instances in Total-Text are labeled at word level with variable number of vertices.

ICDAR-ArT. ICDAR-ArT [35] is a large-scale arbitrary-shaped text detection dataset which collects 5603 training images and 4563 test images. It is a comprehensive dataset extended from SCUT-CTW1500 and TotalText. The whole sets of SCUT-CTW1500 and TotalText are included in the training set of ICDAR-ArT with random changes in annotation and color. Similar to TotalText, the word-level text instances in ICDAR-ArT are labeled with adaptive number of vertices.

ICDAR2015. ICDAR2015 [36] is an incidental scene text dataset which contains 1000 training images and 500 test images. It is a challenging because the text instances in the images appear in random scale, orientation, location, viewpoint and blurring. The annotations of ICDAR2015 are provided at word-level in the form of quadrilateral bounding-boxes represented by 8 coordinates of four clock-wise corners.

MSRA-TD500. MSRA-TD500 [37] contains 300 training images and 200 test images of multi-oriented texts. It is also a multi-lingual dataset including English and Chinese. Unlike ICDAR2015, the annotations of MSRA-TD500 are at line-level which are represented by aligned horizontal rectangles and their orientations. We follow the previous works [1], [8] to include the 400 images from HUST-TR400 [38] as training data.

We adopt the same standard evaluation metric as the ICDAR challenges for all experiments.

B. Implementation Details

We adopt ResNet18 and ResNet50 [28] pre-trained on ImageNet [39] as our backbone and employ an FPN [29] neck for multi-level feature maps. $\{P_2, P_3, P_4, P_5\}$ of the FPN are fused to a single scale of 4×4 down-sampling of the input image. The feature fusion layers and convolutional blocks in the four branches are trained from scratch with Kaiming initialization [40], while the logits prediction layers are initialized with Gaussian distribution of 0 mean and 0.01 standard deviation. Our framework is implemented on PyTorch [41] and MMDetection [42], and is optimized by stochastic gradient descent (SGD) with a momentum of 0.99 and weight decay of 0.0005. We adopt the "cosine" policy to adjust the learning rate. We adopt online augmentations including random color jittering, random rotation in the range of $[-10°, 10°]$ and random resize while preserving the original aspect ratios of input images. In particular, we conduct random cropping on ICDAR2015 for the targets are relatively sparse and small. We train the models with batch size 16 on 4 Titan X (Pascal). The weighting parameter λ in Equation 2-3 is empirically set to 0.1.

C. Ablation Study

We conduct a series of ablation studies on SCUT-CTW1500 and TotalText to demonstrate the effectiveness of our fuzzy semantics exploration, reliability analysis and cross-image normalized focal loss in arbitrary-shaped scene text detection. All ablation experiments adopt ResNet50 as backbone and are without extra pre-training. The training size of both datasets are between the ranges of $[1024, 480]$ and $[1024, 800]$. In the inference stage, the input size of SCUT-CTW1500 and TotalText are $[1200, 800]$ and $[1333, 800]$, respectively. The initial learning rate is 0.001 and training schedule is 500 epochs on both SCUT-CTW1500 and TotalText. Detections whose confidence scores are over 0.85 will be taken as the final detection results. Other implementation details are as described in Section IV-B. As shown in Table I, we design five different experiments to study the effectiveness of our method. The controlled variables are the utilization of separatrix segmentation, reliability analysis and cross-image normalized focal loss.

a) ***Effectiveness of separatrix segmentation:*** The first line in Table I is our baseline which only incorporates the text region segmentation branch and optimizes with cross-image normalized focal loss defined in Equation 10. The output text segmentation map is taken as the final textness map. To demonstrate the effectiveness of exploring the fuzzy semantics of instance separatrix, we conduct multi-label categorization for each pixel location by incorporating separatrix segmentation branch, as shown in the second line in Table I. In this experiment we generate textness map by directly apply Equation 4 without reliability analysis. The third line in Table I represents the experiment with text segmentation and text reliability regression. And the fourth line is our proposed method. Detection candidates are generated by binarizing the textness score maps and extracting contours of connected components as described in Equation 5.

TABLE I
ABLATION STUDY ON SCUT-CTW1500 AND TOTALTEXT.

Controlled Variables				SCUT-CTW1500			TotalText		
Text	Separatrix	Reliability	NormFocal	Precision	Recall	F-measure	Precision	Recall	F-measure
✓	–	–	✓	79.8	65.1	71.7	84.8	69.2	76.2
✓	✓	–	✓	84.4	78.7	81.4	87.4	76.7	81.7
✓	–	✓	✓	79.8	65.2	71.8	85.0	70.6	77.1
✓	✓	✓	✓	**84.7**	**78.9**	**81.7**	**87.5**	**77.9**	**82.5**
✓	✓	–	–	84.4	76.9	80.5	85.2	76.7	80.7

According to the quantitative results listed in the first and second lines of Table I, the exploration of instance separatrixes brings salient improvements of 9.7% and 5.5% in F-measure on SCUT-CTW1500 and TotalText, respectively. And as shown in the third and fourth lines of Table I, separatrix segmentation and separatrix reliability analysis brings about 9.9% and 5.4% improvements in F-measure against sole text segmentation and text reliability analysis on SCUT-CTW1500 and TotalText, respectively. The above two comparisons demonstrate the effectiveness of separatrix exploration. As shown in the (1) and (6) rows of Figure 5, text segmentation alone are very fragile to false positive pixels around the semantic boundaries. And adjacent instances are connected by a very small amount of false positives in the separatrix area, resulting in merged contours in the initial detections. As shown in the (2) row of Figure 5, instance separatrixes are successfully segmented, which proves that separatrix is a unique and recognizable semantic category in natural scene images. And they play a critical role in separating cluttered text instances as shown in the (5) and (7) rows of Figure 5, where the merged text instances in the initial detections are effectively separated by removing separatrix pixels.

*b) **Effectiveness of reliability analysis**:* Except for the intrinsically fuzzy semantic boundaries, the coarse pixel labeling for both text regions and instance separatrixes introduces a certain amount of noisy data. To alleviate the negative influence of the confusing data, we adopt additional branches to predict the reliability of each pixel location on both text and separatrix segmentation. The fourth line in Table I shows the experiment with four branches conducting text segmentation, text reliability regression, separatrix segmentation and separatrix reliability regression, which is our proposed method. The reliability analysis is involved in both training and inference stages as described in Section III-B and III-D.

As shown in the first and third lines of Table I, reliability analysis on text brings 0.1% and 0.9% improvements against sole text segmentation on SCUT-CTW1500 and TotalText. And according to the second and fourth lines of Table I, reliability analysis on both text and separatrix brings 0.3% and 0.8% improvements against text and separatrix segmentation baselines in F-measure. And as shown in the (3) and (4) rows of Figure 5, the predicted reliability maps are visually reasonable and successfully model the fuzzy semantic boundaries of both text and separatrix. The experimental results demonstrate the rationality and effectiveness of our proposed reliability analysis.

TABLE II
COMPARISONS ON TOTALTEXT

Method	Venue	Backbone	P	R	F	FPS
TextSnake [1]	ECCV'18	VGG16	82.7	74.5	78.4	–
Wang *et al.* [15]	CVPR'19	VGG16	80.9	76.2	78.5	–
SAST [3]	MM'19	Res50	83.8	76.9	80.2	–
CSE [11]	CVPR'19	Res34	81.4	79.1	80.2	2.4
TextDragon [26]	ICCV'19	VGG16	85.6	75.7	80.3	–
TextField [4]	TIP'19	VGG16	81.2	79.9	80.6	–
PSENet-1s [5]	CVPR'19	Res50	84.0	78.0	80.9	3.9
PSENet-4s [5]	CVPR'19	Res50	84.5	75.2	79.6	8.4
LOMO [6]	CVPR'19	Res50	88.6	75.7	81.6	–
CRAFT [7]	CVPR'19	VGG16	87.6	79.9	83.6	–
PAN [8]	ICCV'19	Res18	**89.3**	81.0	**85.0**	39.6
CRNet [9]	MM'20	Res50	85.8	**82.5**	84.1	4.6
TextRay [17]	MM'20	Res50	83.5	77.9	80.6	–
DB-R18 [10]	AAAI'20	Res18	88.3	77.9	82.8	50
DB-R50 [10]	AAAI'20	Res50	87.1	**82.5**	84.7	32
Ours-R18	–	Res18	85.8	77.0	81.1	33.5
Ours-R50	–	Res50	88.7	79.9	84.1	24.3

TABLE III
COMPARISONS ON SCUT-CTW1500

Method	Venue	Backbone	P	R	F	FPS
TextSnake [1]	ECCV'18	VGG16	67.9	**85.3**	75.6	–
Wang *et al.* [15]	CVPR'19	VGG16	80.1	80.2	80.1	–
Tian *et al.* [2]	CVPR'19	Res50	82.7	77.8	80.1	–
SAST [3]	MM'19	Res50	85.3	77.1	81.0	27.6
CSE [11]	CVPR'19	Res34	81.1	76.0	78.4	2.6
TextField [4]	TIP'19	VGG16	83.0	79.8	81.4	–
PSENet-1s [5]	CVPR'19	Res50	84.8	79.7	82.2	3.9
PSENet-4s [5]	CVPR'19	Res50	82.1	77.8	79.9	8.4
LOMO [6]	CVPR'19	Res50	**89.2**	69.6	78.4	4.4
CRAFT [7]	CVPR'19	VGG16	86.0	81.1	83.5	–
PAN [8]	ICCV'19	Res18	86.4	81.2	83.7	39.8
CRNet [9]	MM'20	Res50	87.0	80.9	83.8	4.6
TextRay [17]	MM'20	Res50	82.8	80.4	81.6	–
DB-R18 [10]	AAAI'20	Res18	84.8	77.5	81.0	55
DB-R50 [10]	AAAI'20	Res50	86.9	80.2	83.4	22
Ours-R18	–	Res18	84.6	77.7	81.0	35.2
Ours-R50	–	Res50	85.3	82.5	**83.9**	25.1

*c) **Effectiveness of cross-image normalized focal loss**:* The two fuzzy semantic categories we explore in the proposed method suffer from extremely biased negative and positive sample ratios. As one-class segmentation tasks, their foreground pixels are far less than background pixels. To deal with this problem, we propose a cross-image normalized focal loss defined in Equation 10 which adaptively balances the training weights of positive and negative samples. To demonstrate its effectiveness, we conduct an experiment in which we replace the proposed cross-image normalized focal loss in the second line experiment with standard focal loss. For fair comparison, we increase the loss weight of the standard focal loss to 100.

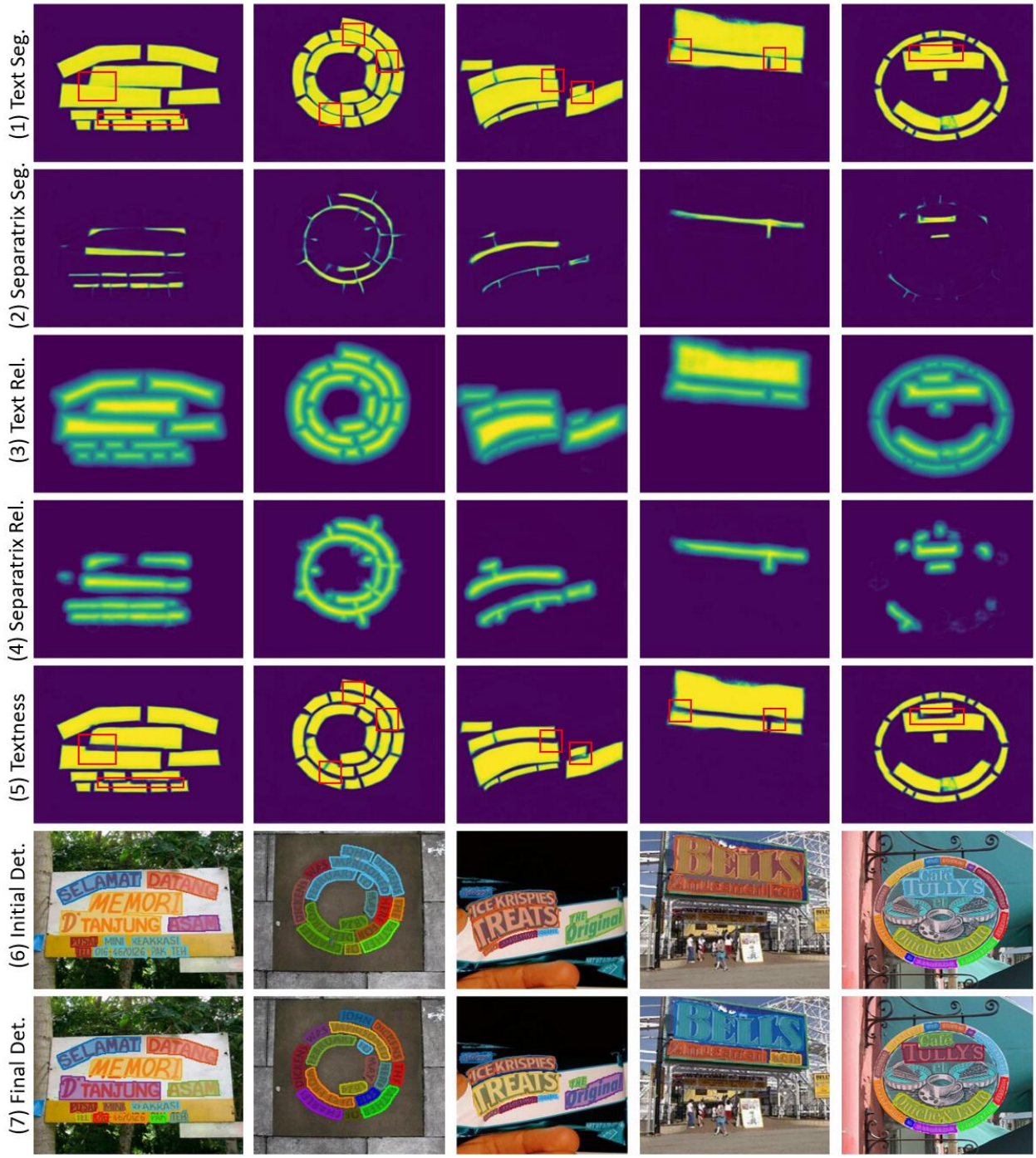


Fig. 5. Outputs demonstration on TotalText. Rows from top to bottom are text segmentation maps, separatrix segmentation maps, text reliability predictions, separatrix reliability predictions, fused textness maps, initial detections extracted from text segmentation maps and final detections extracted from textness maps. The merged instances in the initial detections are visualized by a single color area, while in the final detections the instances are separated and visualized by different colors.

TABLE IV
RESULTS ON ICDAR-ART

Method	Venue	Backbone	P	R	F	FPS
TextRay [17]	MM'20	Res50	76.0	58.6	66.2	–
Ours-R50	–	Res50	**78.5**	**60.6**	**68.4**	24.5

This experiment is shown in the fifth line in Table I. According to the quantitative results, cross-image normalized focal loss achieves 0.9% and 1.0% improvement against standard focal loss, which proves its effectiveness.

D. Comparisons with State-of-the-Art Methods

a) ***Curve text detection****:* To evaluate the effectiveness of our method in detecting curve texts, we conduct experiments on three benchmark datasets: TotalText, SCUT-CTW1500 and ICDAR-ArT. In the training stage, input images are randomly resized between the ranges of $[1024, 480]$ and $[1024, 800]$.

TotalText. The TotalText model is pretrained on a selected subset of ICDAR-ArT, which excludes all the test images

TABLE V
COMPARISONS ON MSRA-TD500

Method	Venue	Backbone	P	R	F	FPS
TextSnake [1]	ECCV'18	VGG16	83.2	73.9	78.3	1.1
Wang *et al.* [15]	CVPR'19	VGG16	85.2	82.1	83.6	–
Tian *et al.* [2]	CVPR'19	Res50	84.2	81.7	82.9	–
TextField [4]	TIP'19	VGG16	87.4	75.9	81.3	–
CRAFT [7]	CVPR'19	VGG16	88.2	78.2	82.9	8.6
PAN [8]	ICCV'19	Res18	84.4	**83.8**	84.1	30.2
CRNet [9]	MM'20	Res50	86.0	82.0	84.0	4.6
DB-R18 [10]	AAAI'20	Res18	90.4	76.3	82.8	62
DB-R50 [10]	AAAI'20	Res50	**91.5**	79.2	84.9	32
Ours-R18	–	Res18	90.0	80.4	84.9	35.5
Ours-R50	–	Res50	89.3	81.6	**85.3**	25.4

TABLE VI
COMPARISONS ON ICDAR2015

Method	Venue	Backbone	P	R	F	FPS
TextSnake [1]	ECCV'18	VGG16	84.9	80.4	82.6	1.1
Tian *et al.* [2]	CVPR'19	Res50	88.3	85.0	86.6	–
SAST [3]	MM'19	Res50	86.7	**87.1**	86.9	–
TextField [4]	TIP'19	VGG16	84.3	80.5	82.4	1.8
PSENet-1s [5]	CVPR'19	Res50	86.9	84.5	85.7	1.6
PSENet-4s [5]	CVPR'19	Res50	86.1	83.8	84.9	3.8
LOMO [6]	CVPR'19	Res50	91.3	83.5	87.2	–
CRAFT [7]	CVPR'19	VGG16	89.8	84.3	86.9	8.6
PAN [8]	ICCV'19	Res18	84.0	81.9	82.9	26.1
CRNet [9]	MM'20	Res50	88.3	84.5	86.4	4.6
DB-R18 [10]	AAAI'20	Res18	86.8	78.4	82.3	48
DB-R50 [10]	AAAI'20	Res50	**91.8**	83.2	**87.3**	12
Ours-R18	–	Res18	88.1	78.8	83.2	15.3
Ours-R50	–	Res50	89.8	82.7	86.1	12.1

of TotalText. The pretraining schedule is 300 epochs with initial learning rate of 0.001, and finetune on TotalText for 500 epochs with initial learning rate of 0.0001. The test scale of TotalText is $[1333, 800]$. Detections with confident scores over 0.83 are taken as final results. The quantitative results are shown in Table II. Comparing with the representative seed expansion strategies like PSENet [5] and CSE [11], our

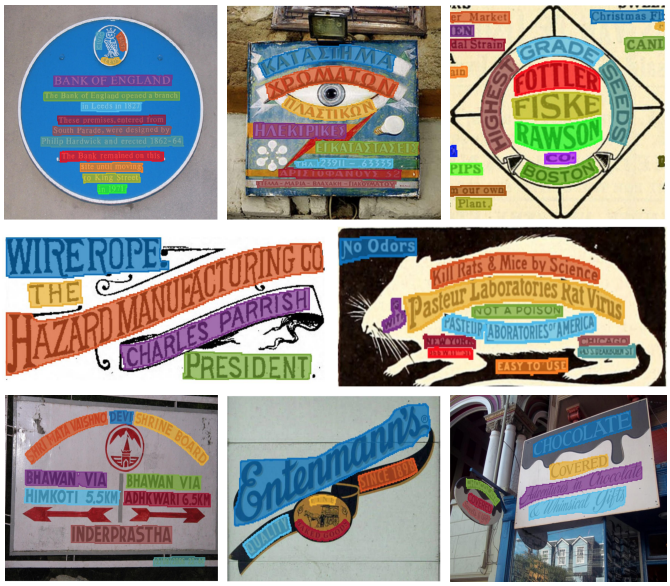


Fig. 6. Qualitative results on SCUT-CTW1500.

TABLE VII
EFFICIENCY AND TIME CONSUMPTION OF OUR METHOD ON CTW1500

Method	F	Time Consumption(ms)			FPS
		Backbone	Head	Post	
Ours-800	**83.9**	18.9	8.1	12.8	25.1
Ours-640	83.0	11.2	5.1	7.9	41.3
Ours-512	80.7	7.1	3.0	4.7	67.6
Ours-320	74.4	5.7	2.1	3.5	**88.5**

redundancy removal strategy achieves favorable results. Also, our simple and straightforward method outperforms several multi-stage processing methods such as TextDragon [26] and CRAFT [7]. The last row in Figure 5 shows some qualitative results on TotalText. As demonstrated, our method successfully detects arbitrary-shaped text instances at word-level and achieves very competitive results on TotalText dataset.

SCUT-CTW1500. Similar to TotalText, the pretraining data of SCUT-CTW1500 model is obtained by excluding the test set of SCUT-CTW1500 from the training set of ICDAR-ArT. The pretraining schedule is 500 epochs with initial learning rate of 0.001, and finetune on SCUT-CTW1500 for 500 epochs with initial learning rate of 0.0001. The test scale of SCUT-CTW1500 is $[1200, 800]$. We take detections whose confident scores are over 0.81 as final results. According to the quantitative results listed in Table III, our method surpasses several typical seed expansion strategies like PSENet [5], CSE [11], PAN [8] and the benchmark segmentation-based method DB [10], achieving state-of-the-art performances. The qualitative results on SCUT-CTW1500 are shown in Figure 6. As we can see, our method is robust in accurately detecting arbitrary-shaped and long textlines.

ICDAR-ArT. ICDAR-ArT is a challenging large-scale dataset which includes Chinese and English text instances. The model is trained on the whole training set without extra pretraining, and the training schedule is 300 epochs with initial learning rate of 0.001. The test scale of ICDAR-ArT is $[1440, 960]$. Detections with confident scores over 0.5 are taken as final results. Table IV shows the quantitative results of our method and Figure 7 shows some qualitative results on ICDAR-ArT. Our method surpasses the regression-based method TextRay [17] and extracts tight and accurate text areas. The experimental results demonstrate the robustness of our method in detecting arbitrary-shaped multi-lingual texts.

*b) **Multi-oriented text detection**:* To evaluate the effectiveness of our method in detecting multi-oriented texts, we conduct experiments on two benchmark datasets: MSRA-TD500 and ICDAR2015.

MSRA-TD500. MSRA-TD500 is a small scale line-level dataset which includes only 300 training images, so we use the line-level SCUT-CTW1500 model as the pretrain model and borrow the 400 images in HUST-TR400 as training data. In the training stage, input images are randomly resized between the ranges of $[1024, 480]$ and $[1024, 800]$. The test scale is $[1024, 1024]$. The finetuning schedule is 500 epochs with initial learning rate of 0.0001. We take detections whose scores are over 0.8 as final results. According to the quantitative results listed in Table V, our method achieves 85.3 in F-measure which is the state-of-the-art performance. The

Fig. 7. Qualitative results on ICDAR-ArT.

ResNet18 version of our method also achieves 84.9 in F-measure, which is well-matched with the ResNet50 version of DB [10] while running at a higher FPS. As we can see, our method is robust in accurately and efficiently detecting multi-oriented textlines.

ICDAR2015. The text targets in ICDAR2015 are commonly sparse and small, so we conduct an extra random crop procedure in data augmentation stage. Input images are randomly rescaled to $[0.5, 3]$ times of the original scale and cropped into $[640, 640]$ image patches. The word-level TotalText model is adopted as the pretrain model. The finetuning schedule is 500 epochs with initial learning rate of 0.0001. The test scale of ICDAR2015 is $[2048, 1152]$. Detections with confident scores over 0.7 are taken as final results. Table VI shows the quantitative results of our method. As shown in the table, the ResNet18 version of our method surpasses PAN [8] and DB-R18 [10] which are based on the same backbone. And the ResNet50 version of our method surpasses many benchmark segmentation-based methods including PSENet [5] and PAN and achieves comparable performance with other state-of-the-art methods. The experimental results demonstrate the robustness of our method in detecting multi-oriented word-level texts.

*c) **Speed analysis**:* The running efficiency of our method and the comparing methods on the evaluation datasets are listed in Table II to Table VI. The top three efficient methods are marked with colored numbers: red, green and blue represents first, second and third rank, respectively. According to the statistics, our method runs at about 25FPS with ResNet50 and 35FPS with ResNet18 on TotalText, SCUT-CTW1500 and MSRA-TD500 datasets. The inference scales of the above three datasets are $[1200, 800]$, $[1333, 800]$ and $[1024, 1024]$. For the larger input size $[2048, 1152]$ which we adopted on ICDAR2015, the running speeds for the ResNet50 and ResNet18 versions of our method are 15.1FPS and 15.3FPS, respectively. In comparison, the three ResNet18-based methods, DB-R18, PAN and Ours-R18, are commonly the top three efficient methods among all the comparing methods. Besides, among ResNet50-based methods, Ours-R50 surpasses DB-R50 on SCUT-CTW1500 and ICDAR2015 datasets in FPS and achieves the most efficient method. Comprehensively, as demonstrated by the above experiments, our method is both effective and efficient among the state-of-the-arts.

To study the trade-off between efficiency and performance under different input sizes and investigate the detailed time consumption of our model, we scale the shorter side of the the input images into four fixed values, e.g. [800, 640, 512, 320], while preserving the original aspect ratio to test our ResNet50-based model on SCUT-CTW1500 dataset. The according performance, detailed time consumption and the overall efficiency of our method is presented in Table VII. As shown in Table VII, the time consumption of the backbone and head takes about 2/3 of the overall processing time. Besides, the shrinking of input size incurs efficiency increase and performance drop. Comprehensively, our model is capable of fast text detection under different input sizes while maintaining a satisfactory performance.

V. Conclusion

In this work, we propose a segmentation-based arbitrary-shaped scene text detection method which explores the fuzzy semantics of text region and instance separatrix in natural scene images. To fully explore the fuzzy semantics, we model the ambiguous semantic boundaries with reliability analysis, and acquire text instances with intact and clear separatrixes from a redundancy removal perspective. We also design a cross-image normalized focal loss with the guidance of reliability to balance the extremely imbalanced positive and negative pixel samples and alleviate the impact of noisy data. Experiments demonstrate the uniqueness of instance separatrixes as a recognizable semantic category in natural scene images and the effectiveness of our fuzzy semantics exploration.

Acknowledgment

The authors would like to thank...

References

[1] S. Long, J. Ruan, W. Zhang, X. He, W. Wu, and C. Yao, "Textsnake: A flexible representation for detecting text of arbitrary shapes," in *Proceedings of the European Conference on Computer Vision (ECCV)*, vol. 11206. Springer, 2018, pp. 19–35.

[2] Z. Tian, M. Shu, P. Lyu, R. Li, C. Zhou, X. Shen, and J. Jia, "Learning shape-aware embedding for scene text detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 2019, pp. 4234–4243.

[3] P. Wang, C. Zhang, F. Qi, Z. Huang, M. En, J. Han, J. Liu, E. Ding, and G. Shi, "A single-shot arbitrarily-shaped text detector based on context attended multi-task learning," in *Proceedings of the ACM International Conference on Multimedia (MM)*. ACM, 2019, pp. 1277–1285.

[4] Y. Xu, Y. Wang, W. Zhou, Y. Wang, Z. Yang, and X. Bai, "Textfield: Learning a deep direction field for irregular scene text detection," *IEEE Transactions on Image Processing (TIP)*, vol. 28, no. 11, pp. 5566–5579, 2019.

[5] W. Wang, E. Xie, X. Li, W. Hou, T. Lu, G. Yu, and S. Shao, "Shape robust text detection with progressive scale expansion network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 2019, pp. 9336–9345.

[6] C. Zhang, B. Liang, Z. Huang, M. En, J. Han, E. Ding, and X. Ding, "Look more than once: An accurate detector for text of arbitrary shapes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 2019, pp. 10 552–10 561.

[7] Y. Baek, B. Lee, D. Han, S. Yun, and H. Lee, "Character region awareness for text detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 2019, pp. 9365–9374.

[8] W. Wang, E. Xie, X. Song, Y. Zang, W. Wang, T. Lu, G. Yu, and C. Shen, "Efficient and accurate arbitrary-shaped text detection with pixel aggregation network," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. IEEE Computer Society, 2019, pp. 8439–8448.

[9] Y. Zhou, H. Xie, S. Fang, Y. Li, and Y. Zhang, "Crnet: A center-aware representation for detecting text of arbitrary shapes," in *Proceedings of the ACM International Conference on Multimedia (MM)*. ACM, 2020, pp. 2571–2580.

[10] M. Liao, Z. Wan, C. Yao, K. Chen, and X. Bai, "Real-time scene text detection with differentiable binarization," in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*. AAAI Press, 2020.

[11] Z. Liu, G. Lin, S. Yang, F. Liu, W. Lin, and W. L. Goh, "Towards robust curve text detection with conditional spatial expansion," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 2019, pp. 7269–7278.

[12] J. Duan, Y. Xu, Z. Kuang, X. Yue, H. Sun, Y. Guan, and W. Zhang, "Geometry normalization networks for accurate scene text detection," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. IEEE Computer Society, 2019, pp. 9136–9145.

[13] M. Liao, Z. Zhu, B. Shi, G. Xia, and X. Bai, "Rotation-sensitive regression for oriented scene text detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 2018, pp. 5909–5918.

[14] F. Wang, L. Zhao, X. Li, X. Wang, and D. Tao, "Geometry-aware scene text detection with instance transformation network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 2018, pp. 1381–1389.

[15] X. Wang, Y. Jiang, Z. Luo, C. Liu, H. Choi, and S. Kim, "Arbitrary shape scene text detection with adaptive text region representation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 2019, pp. 6449–6458.

[16] Y. Liu, H. Chen, C. Shen, T. He, L. Jin, and L. Wang, "Abcnet: Real-time scene text spotting with adaptive bezier-curve network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 2020, pp. 9806–9815.

[17] F. Wang, Y. Chen, F. Wu, and X. Li, "Textray: Contour-based geometric modeling for arbitrary-shaped scene text detection," in *Proceedings of the ACM International Conference on Multimedia (MM)*. ACM, 2020, pp. 111–119.

[18] Z. Zhang, C. Zhang, W. Shen, C. Yao, W. Liu, and X. Bai, "Multi-oriented text detection with fully convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 2016, pp. 4159–4167.

[19] Z. Liu, G. Lin, S. Yang, J. Feng, W. Lin, and W. L. Goh, "Learning markov clustering networks for scene text detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 2018, pp. 6936–6944.

[20] P. Lyu, M. Liao, C. Yao, W. Wu, and X. Bai, "Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes," in *Proceedings of the European Conference on Computer Vision (ECCV)*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds., vol. 11218. Springer, 2018, pp. 71–88.

[21] Y. Wang, H. Xie, Z. Zha, M. Xing, Z. Fu, and Y. Zhang, "Contournet: Taking a further step toward accurate arbitrary-shaped scene text detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 2020, pp. 11 750–11 759.

[22] D. Deng, H. Liu, X. Li, and D. Cai, "Pixellink: Detecting scene text via instance segmentation," in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, S. A. McIlraith and K. Q. Weinberger, Eds. AAAI Press, 2018, pp. 6773–6780.

[23] P. Lyu, C. Yao, W. Wu, S. Yan, and X. Bai, "Multi-oriented scene text detection via corner localization and region segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 2018, pp. 7553–7563.

[24] B. Shi, X. Bai, and S. J. Belongie, "Detecting oriented text in natural images by linking segments," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 2017, pp. 3482–3490.

[25] S. Zhang, X. Zhu, J. Hou, C. Liu, C. Yang, H. Wang, and X. Yin, "Deep relational reasoning graph network for arbitrary shape text detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 2020, pp. 9696–9705.

[26] W. Feng, W. He, F. Yin, X. Zhang, and C. Liu, "Textdragon: An end-to-end framework for arbitrary shaped text spotting," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. IEEE Computer Society, 2019, pp. 9075–9084.

[27] S. J. Osher and R. Fedkiw, *Level set methods and dynamic implicit surfaces*, ser. Applied mathematical sciences. Springer, 2003, vol. 153.

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 2016, pp. 770–778.

[29] T. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 2017, pp. 936–944.

[30] X. Wang, R. Zhang, T. Kong, L. Li, and C. Shen, "Solov2: Dynamic and fast instance segmentation," in *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[31] R. B. Girshick, "Fast R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. IEEE Computer Society, 2015, pp. 1440–1448.

[32] T. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. IEEE Computer Society, 2017, pp. 2999–3007.

[33] Y. Liu, L. Jin, S. Zhang, C. Luo, and S. Zhang, "Curved scene text detection via transverse and longitudinal sequence connection," *Pattern Recognition*, vol. 90, pp. 337–345, 2019.

[34] C. K. Chng and C. S. Chan, "Total-text: A comprehensive dataset for scene text detection and recognition," in *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR)*. IEEE Computer Society, 2017, pp. 935–942.

[35] C. K. Chng, E. Ding, J. Liu, D. Karatzas, C. S. Chan, L. Jin, Y. Liu, Y. Sun, C. C. Ng, C. Luo, Z. Ni, C. Fang, S. Zhang, and J. Han, "ICDAR2019 robust reading challenge on arbitrary-shaped text - rrc-art," in *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR)*. IEEE Computer Society, 2019, pp. 1571–1576.

[36] D. Karatzas, L. Gomez-Bigorda, A. Nicolaou, S. K. Ghosh, A. D. Bagdanov, M. Iwamura, J. Matas, L. Neumann, V. R. Chandrasekhar, S. Lu, F. Shafait, S. Uchida, and E. Valveny, "ICDAR 2015 competition on robust reading," in *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR)*. IEEE Computer Society, 2015, pp. 1156–1160.

[37] C. Yao, X. Bai, W. Liu, Y. Ma, and Z. Tu, "Detecting texts of arbitrary orientations in natural images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 2012, pp. 1083–1090.

[38] C. Yao, X. Bai, and W. Liu, "A unified framework for multioriented text detection and recognition," *IEEE Transactions on Image Processing (TIP)*, vol. 23, no. 11, pp. 4737–4749, 2014.

[39] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and F. Li, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[40] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. IEEE Computer Society, 2015, pp. 1026–1034.

[41] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2019, pp. 8024–8035.

[42] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu, Z. Zhang, D. Cheng, C. Zhu, T. Cheng, Q. Zhao, B. Li,

X. Lu, R. Zhu, Y. Wu, J. Dai, J. Wang, J. Shi, W. Ouyang, C. C. Loy, and D. Lin, "Mmdetection: Open mmlab detection toolbox and benchmark," *CoRR*, vol. abs/1906.07155, 2019.