Multi-Oriented and Multi-Lingual Scene Text Detection With Direct Regression

Wenhao He, Xu-Yao Zhang, Fei Yin, and Cheng-Lin Liu, *Fellow, IEEE*

***Abstract*— Multi-oriented and multi-lingual scene text detection plays an important role in computer vision area and is challenging due to the wide variety of text and background. In this paper, first, we point out the two key tasks when extending convolutional neural network (CNN)-based object detection frameworks to scene text detection. The first task is to localize the text region by a downsampled segmentation-based module, and the second task is to regress the boundaries of text region determined by the first task. Second, we propose a scene text detection framework based on fully convolutional network with a bi-task prediction module, in which one is a pixel-wise classification between the text and non-text and the other is pixel-wise regression to determine the vertex coordinates of quadrilateral text boundaries. Post-processing for word-level detection is based on non-maximum suppression, and for the line-level detection, we design a heuristic line segments grouping method to localize long text lines. We evaluated the proposed framework on various benchmarks, including multi-oriented and multi-lingual scene text data sets, and achieved the state-of-the-art performance on most of them. We also provide abundant ablation experiments to analyze several key factors in building high performance CNN-based scene text detection systems.**

***Index Terms*— Fully convolutional network, scene text detection, multi-oriented, multi-task.**

I. Introduction

TEXT in scene images plays a critical role in information extraction and scene understanding. Therefore, scene text detection is needed in many practical applications such as automatic driving, reading assistance, mobile OCR and translation. However, scene text detection is also challenging. Unlike scanned documents, scene text owns much more variations in fonts, scales, orientations and perspective distortion. In scene images, background region could be hardly distinguished from text without context information. Other factors like varying illumination and occlusions also complicate the problem.

Approaches for scene text detection can be roughly divided into two groups as shown in Fig. 1 according to the composition of scene texts. An intuition definition is to regard text as a character composite [44], and follows a "grouping characters to line" strategy by firstly localizing characters and then grouping them into a word or text line. In this group, most traditional methods [2], [6], [29] extract connected components of characters. However, these methods adopt manually designed features that are not robust enough to be adaptive to complex scenes. Recent works [33], [39] based on deep convolutional neural network (CNN) [20] generalize the concept of characters, where text blocks rather than characters within a text line, are detected and then these text blocks are grouped into text lines. These methods give excellent performance since "characters" are extracted by a more robust and learnable way instead of heuristic processing.

The second group of methods takes text as object regardless of the constituent characters. Methods [12], [21], [49], [50] in this group mostly take the CNN based generic object detection frameworks such as Faster-RCNN [32], SSD [23], YOLO [31], R-FCN [4]. Without the need of grouping characters or text blocks, these methods give better performance. These methods have two main issues to deal with: to localize text regions in a segmentation task, and to determine text region boundaries in a regression task. The segmentation task mentioned here has a more generalized definition covering the conventional semantic segmentation. Take Faster-RCNN as instance, the Region Proposal Network (RPN) in this framework can be regarded as a shape-based segmentation module, where objects with similar shapes are highlighted under the same anchor prior. To save computation, down-sampled instead of full resolution feature maps are often exported.

For the regression task, there are usually two approaches to determine the boundaries as illustrated in [12]. The first one is indirect regression which is widely adopted in frameworks like Faster-RCNN and SSD. For indirect regression, the network learns the offset from a proposal to the corresponding ground truth. The second approach is direct regression which straightforwardly learns the offset from pixels within an object to the box corners. A visualized explanation of indirect and direct regression is displayed in Fig. 2. The superiority of direct regression is in three folds: 1) complex proposals or anchors to describe long and highly inclined scene text are not necessary;

Manuscript received November 5, 2017; revised May 21, 2018; accepted June 26, 2018. Date of publication July 12, 2018; date of current version August 14, 2018. This work was supported by the National Natural Science Foundation of China (NSFC) under Grant 61721004, Grant 61411136002, Grant 61733007, Grant 61633021 and NVIDIA NVAIL Program. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Junsong Yuan. *(Corresponding author: Cheng-Lin Liu.)*

W. He, X.-Y. Zhang, and F. Yin are with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: wenhao.he@nlpr.ia.ac.cn; xyz@nlpr.ia.ac.cn; fyin@nlpr.ia.ac.cn).

C.-L. Liu is with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, also with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China, and also with the Center for Excellence of Brain Science and Intelligence Technology, Chinese Academy of Sciences, Beijing 100190, China (e-mail: liucl@nlpr.ia.ac.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2018.2855399

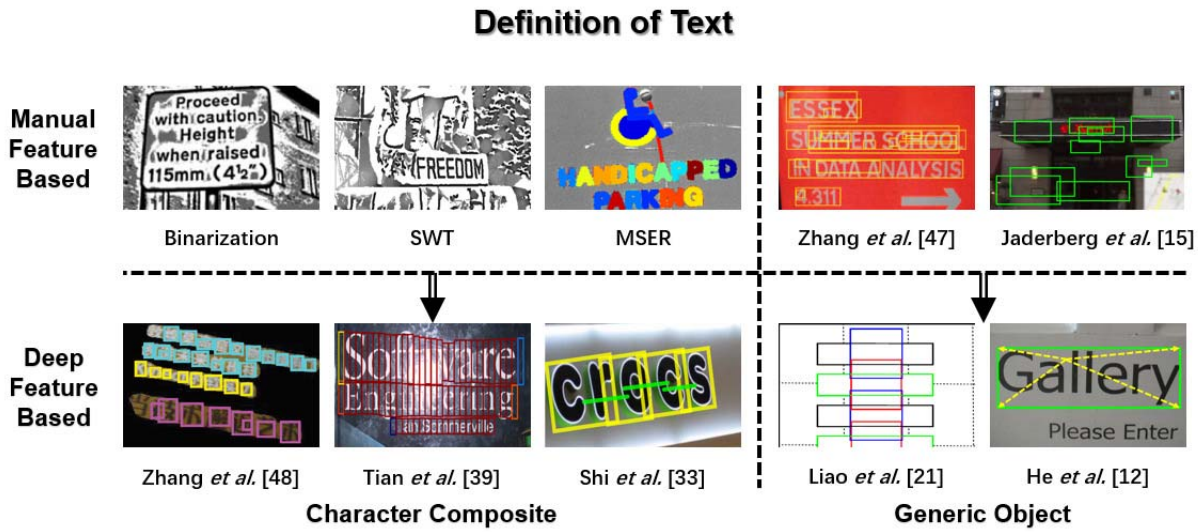
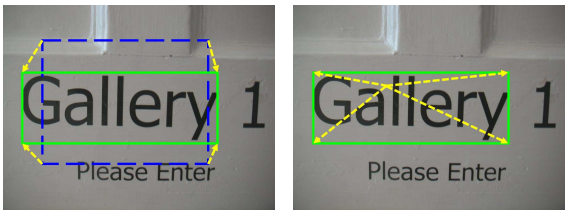


Fig. 1. Two types of definition for text. Left: text is composed by characters. Right: text is a special type of object. Top: text detectors based on manual features. Bottom: text detectors based on deep features.



(a) (b)

Fig. 2. Visualized explanation of indirect and direct regression. The solid green lines are boundaries of text "Gallery", the dashed blue lines are boundaries of text proposal, and the dashed yellow vectors are the ground truths of regression task. (a) The indirect regression predicts the offset from a proposal. (b) The direct regression predicts the offset from a point.

2) regression without anchor mechanism could be more efficient in computation. In [26], 54 types of anchors are related to 378 channels, which cost much more time compared with our 9-channel output. 3) direct regression could avoid quadrilateral matching. The method in [24] and [26] are both based on indirect regression, and spend efforts in matching proposals to ground truths.

In this paper, we propose a scene text detection method by regarding scene text as generic object. Our method uses a fully convolutional network (FCN) [25] with bi-task predictions for extracting features and locating text line boundaries. The first classification task performs down-sampled segmentation between text and non-text, and the second direct regression task learns offset from each text region pixel to the corresponding quadrilateral text boundaries. The direct regression of quadrilaterals facilitates multi-oriented scene text description without any orientation information. For word-level detection, a newly proposed Recalled Non-Maximum Suppression (R-NMS) [12] is implemented for post-processing, while for line-level detection as for Chinese texts, the post-processing is based on a line-segments grouping method to form long text lines. Our detection framework achieves state-of-the-art performance on many benchmarks which contain multi-oriented and multi-lingual texts, as well as cluttered scenes. This verifies the effectiveness of the proposed method. This paper is an extension of our previous conference paper [12], and the major contribution falls in four respects:

1) We extend our previous method designed for English scene text detection such that it can achieve superior performance on multi-lingual texts;
2) A new post-processing method for line-level detection, as for languages like Chinese, Japanese and Korean, is proposed. By this, our detection method can give both line-level and word-level annotations;
3) We stress the importance of ground truth design by introducing the concepts of positive text scale and transition boundary. The necessity of these settings was justified in experiments;
4) We provide more ablation experiments to evaluate the influence of feature extraction architecture, hard example mining, sufficient receptive field and sigmoid normalization in direct regression task.

The rest of this paper is organized as follows. Section. II gives a brief review of recent works on scene text detection. Section. III introduces the details of the proposed method. Section. IV presents the experimentation details and results. Section. V concludes this paper.

II. Related Work

Depending on the definition of text as shown in Fig. 1, scene text detection methods can be roughly partitioned into four categories according to two dimensions: character-level v.s. word/line-level, manual feature v.s. deep feature.

Case 1 (Character-Level With Manual Feature): Most previous methods extract connected components as character candidates. In early works connected components were extracted by Niblack's local binarization method [30], Stroke Width Transformation (SWT) [6], [42], among other image segmentation techniques. Maximally Stable Extremal Regions (MSER) based methods [2], [3], [46] were widely studied in last few years, and some recent works [40], [41] also adopt super-pixel based segmentation.

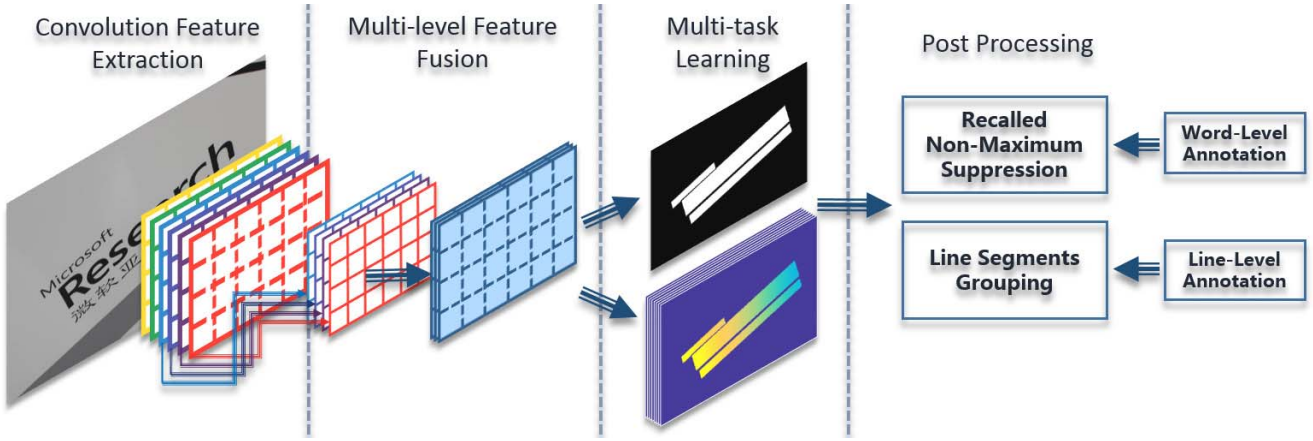


Fig. 3. Overview of the proposed text detection method.

Case 2 (Character-Level With Deep Feature): Recent CNN based methods promote character-level text detection by learning efficient features. The method of [11] combines FCN and multi-task module for precise character candidate extraction, and can be viewed as a transition work from traditional framework to CNN based one. Zhang *et al.* [48] and Yao *et al.* [43] both extract characters by segmenting character-level regions based on FCN. Tian *et al.* [39] uses the RPN module in Faster-RCNN to detect slices rather than real characters within each text word/line. The text slices here can be regarded as characters in another form, and are grouped into text words/lines in a post-processing procedure. Shi *et al.* [33] improves the method of [39] by judging whether two text slices are connected. Consequently, in the post-processing, text slices are separated into different groups according to the linking information, and slices within the same group are linked into a text line. This method fails to detect text lines with wide character space where linking information is obscure.

Case 3 (Word/Line-Level With Manual Feature): The method in [47] is one of the early works that treat text detection as object detection by taking advantages of the local binary pattern (LBP) feature [28] and the symmetric characteristic of horizontal scene text lines. Jaderberg *et al.* [15] extracts word region proposals with high recall rate based on manual feature based detectors like Edge Boxes [51] and aggregate channel feature (ACF) detector [5], and then filter or refine these proposals by a random forest [1] based classifier.

Case 4 (Word/Line-Level With Deep Feature): Recent scene text detection methods in [9], [21], and [49] are based on Faster-RCNN, SSD and YOLO which are widely applied in generic object detection. However, these methods predict the bounding boxes of texts in rectangular shape, and as a result, they can only handle horizontal scene text detection. To cope with multi-oriented texts, Liu and Jin [24] modifies the anchor mechanism proposed in Faster-RCNN into rotated form and learn the offset from the best matching anchor to the ground truth text boundary. Furthermore, He *et al.* [12] and Zhou *et al.* [50] abandon the anchors for boundary regression and learn to describe the text boundaries directly, and both two methods achieve significant progress surpassing previous methods by a large margin. This indicates the superiority of describing quadrilateral text boundaries directly rather than learning the offset from rectangular candidates.

III. The Proposed Method

The proposed text detection method is diagrammed in Fig. 3. It consists of four major modules: the first three modules, namely convolutional feature extraction, multi-level feature fusion, multi-task learning, together constitute the network part, and the last post-processing module is switched depending on whether word-level or line-level text is required. For word-level detection, post-processing is based on Recalled NMS, and for line-level detection, post-processing is based on heuristic line segments grouping. In the following, we will introduce details of each module, ground truth design, loss functions and data augmentation, respectively.

A. Feature Extraction

To ensure that the regression task could "see" long texts and give more accurate boundary measurement, the convolutional feature extraction part is designed such that the maximum receptive field is larger than the input image size S (here $S = 320$). In this work, we adopt three types of feature extraction structure for comparison and analysis. The first two are VGG-16 [36] and ResNet-50 [10] which are originally designed for classification task on ImageNet and widely adapted to other computer vision domains. For VGG-16, we use the sub-network by removing the fully connected layers and appending additional 3×3 convolutional layers to enlarge the receptive field until it is larger than S. For ResNet-50, we also remove the average pooling and fully connected layers. The third network called S-VGG is the architecture used in [12] which follows the design of VGG-16 but adopts less kernels and inserts batch normalization [14] to boost the loss convergence. The S-VGG network is taken as a baseline network for better comparison. Visualized network structures are shown in Fig. 4.

B. Multi-Level Feature Fusion

Architectures adopting top-down and skip connection are widely used in recent works [13], [19], [22], [50] to produce

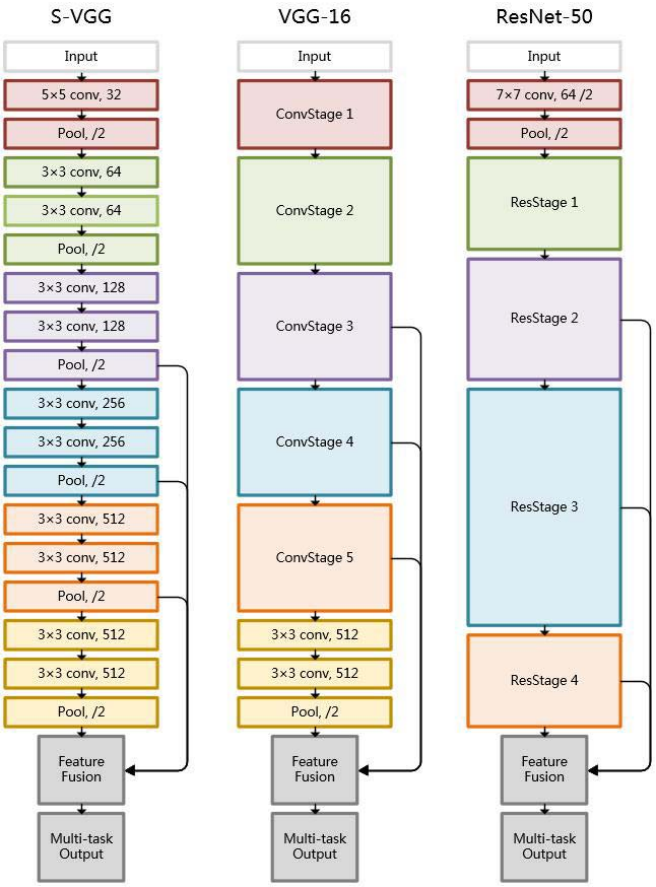


Fig. 4. Three types of feature extraction architecture. **Left**: S-VGG network designed by ourselves. **Middle**: VGG-16 network with additional convolutional layers. **Right**: Residual network with 50 parameter layers.

feature maps of finer-resolution with multi-level features fusion. Features extracted by this fusion manner have two merits: firstly, multi-level features could assist the model to handle multi-scale objects, which brings efficiency in test stage; secondly, lower-level features with higher resolution stress local information which is beneficial for accurate text boundaries localization, while higher-level features with lower resolution stress global information which is beneficial for rough text size estimation. Fusing multi-level features could improve both classification and regression.

The multi-level fusion module is realized by an iterative process. For each step, the top most two feature layers are merged by modules shown in Fig. 5.a. Feature maps of different sizes are firstly sent through a convolutional layer with 1×1 kernels to normalize channel size, and then the lower-resolution feature map is up-sampled by a factor of 2 using a deconvolutional layer. After that, the up-sampled map is merged with the higher-resolution map by element-wise addition. Finally, we up-sample the merged feature maps to be $\frac{1}{4}$ size of the input image.

C. Multi-Task Output

The spatial size of feature maps for each task is also $\frac{1}{4}$ size of the input image to reduce computation since there is no need to get pixel-level information for detection task.

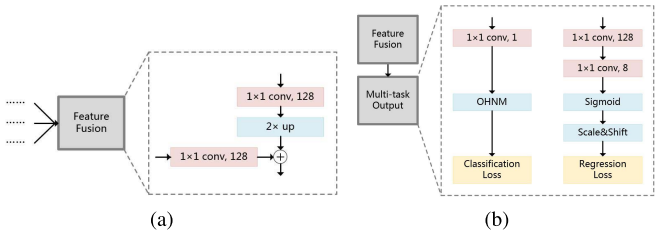


Fig. 5. Structure of Feature Fusion and Multi-task Output. (a) Feature Fusion structure. (b) Multi-task Output structure.

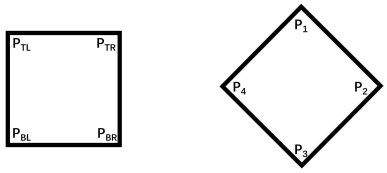


Fig. 6. Vertex order confusion for a quadrilateral. The left quadrilateral has obvious top-left, top-right, bottom-right, bottom-left vertexes, however, there is no obvious top-left or other three vertexes for the right quadrilateral.

The classification task output $\mathcal{M}_{cls}$ is a $\frac{S}{4} \times \frac{S}{4}$ 2nd-order tensor and it can be roughly taken as down-sampled segmentation between text and non-text for input images. $\mathcal{M}_{cls}$ is generated by forwarding the fused feature from a convolutional layer with 1×1 kernel of one channel. Elements in $\mathcal{M}_{cls}$ with higher value are more likely to be text, otherwise non-text;

The regression task output $\mathcal{M}_{loc}$ is a $\frac{S}{4} \times \frac{S}{4} \times 8$ 3rd-order tensor. The channel size of $\mathcal{M}_{loc}$ indicates that 8 coordinates, corresponding to the four quadrilateral vertexes (from top-left to bottom-left in clock-wise order) of a text boundary, are predicted. The value at (w, h, c) in $\mathcal{M}_{loc}$ is denoted as $L_{(w,h,c)}$, and it means the offset from coordinate of a quadrilateral vertex to that of the point at $(4w, 4h)$ in input image, and therefore, the quadrilateral $\mathcal{B}(w, h)$ on the input image can be formulated as

$$\mathcal{B}(w,h) = \left\{ L_{(w,h,2n-1)} + 4w,\ L_{(w,h,2n)} + 4h \middle| n \in \{1,2,3,4\} \right\}. \tag{1}$$

One concern in designing $\mathcal{M}_{loc}$ is to arrange the order of quadrilateral vertexes. In other words, we should find the top-left vertex first and then arrange four vertexes in clock-wise manner. To solve this problem, the key step is to find top-left vertex P_{TL}, and we regard P_{TL} as the top-left vertex if the next vertex P_{TR} in clock-wise order satisfies that

$$\begin{cases} P_{TL} \cdot x \leq P_{TR} \cdot x, \\ P_M = \dfrac{P_{TL} - P_{TR}}{\|P_{TL} - P_{TR}\|_2}, \\ \arccos \dfrac{P_M \cdot y}{P_M \cdot x} \in \left[-\dfrac{\pi}{4}, \dfrac{\pi}{4} \right]. \end{cases} \tag{2}$$

There could be more than one vertex satisfying the above conditions in extreme cases, and for simplicity we choose the first vertex that meets these constraints as P_{TL}. Take the quadrilateral on the right in Fig. 6 as instance, if we begin to check from P_1, then P_1 is P_{TL}, otherwise P_4 is P_{TL}. Since such extreme condition is rarely taken place, the negative

influence of vertex order confusion on regression task could be neglected.

By combining outputs of two tasks, we could predict a scored quadrilateral for each 4×4 region of the input image. Detailed structure of multi-task output module is shown in Fig. 5.b.

D. Ground Truth and Loss Function

For training the network, the full multi-task loss $\mathcal{L}$ can be represented as

$$\mathcal{L} = \mathcal{L}_{cls} + \lambda_{loc} \cdot \mathcal{L}_{loc}, \tag{3}$$

where $\mathcal{L}_{cls}$ and $\mathcal{L}_{loc}$ represent loss for classification task and regression task, respectively. The balance between two losses is controlled by the hyper-parameter λ_{loc}.

1) Classification Task: The ground truth for classification task can be roughly deemed as a down-sampled segmentation between text and non-text, which are usually denoted strictly as positive and negative category [48]. However, in this paper we only regard pixels around the text center line within a distance R as positive and enclose positive region with "NOT CARE" boundary named as **transition boundary** (shown in Fig. 10.d). The parameter R is proportional to the shortest side of text boundaries by ratio of 0.2.

"NOT CARE" region means the backward gradients for this region are forced to be zero. The transition boundary is designed to promote the model to focus more on discriminative regions rather than border areas between text and non-text.

a) Positive text scale: In previous FCN based scene text detection methods [33], [48]–[50], all text regions in training images are taken as positive, and there are few works lay emphasis on constraining text scale. In our work, a text line is taken as a positive sample only when its shortest side length ranges in $\left[32 \times 2^{-1}, 32 \times 2^{1}\right]$ (positive text scale). If the shortest side length falls in $\left[32 \times 2^{-1.5}, 32 \times 2^{-1}\right) \cup \left(32 \times 2^{1}, 32 \times 2^{1.5}\right]$ ("NOT CARE" text scale), we take this text line as "NOT CARE", otherwise negative.

The lower boundary of positive text scale is determined by supposing that both R and transition boundary occupy 1p (pixel) on the ground truth map, which is 4p in total referring to the shortest side length of 16p in original image. The lower boundary of "NOT CARE" text scale is determined similarly by supposing that R occupies 0.5p and transition boundary occupy 1p, which is 3p on ground truth map and 12p in original image. Upper boundaries of positive and "NOT CARE" text scale are determined in a symmetric way.

The main reason to restrict positive scale range is that for huge text, the CNN could only see simple strokes and suffers to recognize text through more context. While for tiny text, it could lose much detailed information after the early down-sampling layers.

The loss function $\mathcal{L}_{cls}$ chosen for classification task is the squared hinge loss. Denote the label of text as 1, non-text as 0, and the ground truth for a given pixel as $y_i^* \in \{0, 1\}$ and predicted value as $\hat{y}_i$, $\mathcal{L}_{cls}$ is formulated as

$$\mathcal{L}_{cls} = \frac{1}{S^2} \sum_{i \in \mathcal{L}_{cls}} \max\left(0, \operatorname{sign}\left(0.5 - y_i^*\right) \cdot \left(\hat{y}_i - y_i^*\right)\right)^2. \tag{4}$$

Algorithm 1 Online Hard Negative Mining

Input: P – positive pixel index set
N – negative pixel index set
R_n – negative ratio
R_{hn} – hard negative ratio
Output: N_s – output selected negative pixel index set

for each iteration **do**
  $N_s = \emptyset$
  if $|P| < |N|$ **then**
    $[\sim, N_d] = sort(\hat{y}_i, \text{'}descend\text{'})$
    $C_n = (|P| + |N|) \cdot R_n$
    $C_h = C_n \cdot R_{hn}$
    $N_s\left(1 : C_h\right) = N_d\left(1 : C_h\right)$
    $N_l = Randomize\left(N \cap \overline{N_s}\right)$
    $N_s\left(C_h + 1 : C_n\right) = N_l\left(1 : C_n - C_h\right)$
  else
    $N_s = N$
return N_s

b) Online hard negative mining: The proportion between text and non-text is imbalanced, where most regions contain none text, and this brings the imbalanced classification problem. Moreover, non-text class has a large within-class variance because most non-text regions have simple texture, leading the model to have less opportunity to be trained on more complex areas.

To compensate the above two issues, we resort to online hard negative mining (OHNM) which derives from online hard example mining (OHEM) [35]. In simplicity, all positive text region pixels are taken for training, while only hard negative (non-text) regions are retained. Details of OHNM algorithm is illustrated in Algorithm. 1.

2) Regression Task: We use a $Scale\&Shift$ module (shown in Fig. 5.b) for fast loss convergence to cope with the wide range of regression value. $Scale\&Shift$ takes the value z from a sigmoid neuron as input and then stretches z into $\hat{z}$ by

$$\hat{z} = 800 \cdot z - 400, \quad z \in (0, 1), \tag{5}$$

where

$$z = \frac{1}{1 + e^{-z_0}}. \tag{6}$$

Here we assume that the maximum positive text size is less than 400, which is larger than S. Normally, z_0 before $Sigmoid$ layer is assumed as the prediction for regression task, however, this will bring problems like unstable or unable of loss convergence. Passing a sigmoid layer could relieve these problems and this is further analyzed in Section .IV.D.7.

Following [7], the loss function $\mathcal{L}_{loc}$ used in regression task is defined as Eq. 8. Denote the ground truth for a given pixel as z_i^* and predicted value as $\hat{z}_i$, $\mathcal{L}_{loc}$ is formulated as

$$\mathcal{L}_{loc} = \sum_{i \in \mathcal{L}_{loc}} \left[y_i^* > 0\right] \cdot \operatorname{smooth}_{L_1}\left(z_i^* - \hat{z}_i\right), \tag{7}$$

$$\operatorname{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1, \\ |x| - 0.5 & \text{otherwise.} \end{cases} \tag{8}$$

Fig. 7. Three steps in Recalled NMS. Left: results of traditional NMS (quadrilaterals in red are false detection). Middle: recalled high score quadrilaterals. Right: merging results by closeness.

We choose smooth L_1 loss here because it is less sensitive to outliers compared with L_2 loss. During training stage, smooth L_1 loss requires less careful tuning of learning rate and decreases steadily.

E. Post-Processing

After getting the outputs produced by multi-task learning, each point of the prediction map is related with a scored quadrilateral. To filter the non-text region, we only preserve points with score above a threshold in classification task. However, there will be still densely overlapped quadrilaterals for a word or text line, which requires a post-processing step to eliminate redundancy.

Two kinds of post-processing are adopted in our work depending on the annotation manner. For word-level annotation, where most ground truth quadrilaterals are smaller than the receptive field, we choose the Recalled Non-Maximum Suppression as the post-precessing. While for line-level annotation, where many text lines are much longer than the receptive field, we adopt a Line Segments Grouping method.

1) Word-Level Post-Processing: The Recalled NMS is a trade-off solution for two problems: (i) when texts are close, quadrilaterals between two words are often retained because of the difficulty in classifying pixels within word space, (ii) if we solve problem (i) by simply retaining quadrilaterals with higher score, text region with relative lower confidence will be discarded and the overall recall will be sacrificed a lot. The Recalled NMS could both remove quadrilaterals within text spaces and maintain the text region with low confidence. The Recalled NMS has three steps as shown in Fig. 7.

- First, we get suppressed quadrilaterals $\mathcal{B}_{sup}$ from densely overlapped quadrilaterals $\mathcal{B}$ by traditional NMS with overlap threshold 0.5;
- Second, each quadrilateral in $\mathcal{B}_{sup}$ is switched to the one with highest score in $\mathcal{B}$. After this step, quadrilaterals within word space are changed to those of higher score and low confidence text region are preserved as well;
- Third, after the second step we may get dense overlapped quadrilaterals again, and instead of suppression, we merge quadrilaterals in $\mathcal{B}_{sup}$ which are highly overlapped with each other.

2) Line-Level Post-Processing: The line-level post-processing is inspired from character grouping in [37]. However, unlike character-level candidates, the line segments here are arbitrary quadrilaterals instead of rectangles. So before introducing the line-level post-processing, we firstly redeclare attributes like height, aspect ratio, orientation for a quadrilateral. As shown in Fig. 8, $P_1 \cdots P_4$ are the coordinates of four vertexes, $l_1 \cdots l_4$ are the lengths of four edges. The height h, short side pair (s_1, s_2) and aspect ratio r of a quadrilateral are defined by

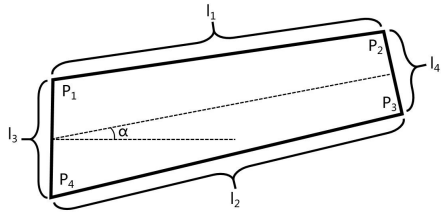


Fig. 8. Quadrilateral attributes. $P_1 \cdots P_4$ are the four vertex coordinates of the quadrilateral, $l_1 \cdots l_4$ are the lengths of four edges, and α is the angle between horizontal line and quadrilateral center line.

Algorithm 2 Conditions of Quadrilateral Shape Category

Input: r – aspect ratio of quadrilateral
α – orientation of quadrilateral
Output: S – shape category of quadrilateral

```
if r ∈ [1, 3/2] then
    return character;
else if α ∈ [−π/4, π/4] then
    return horizontal;
else
    return vertical;
```

$$\begin{cases} h = \min\left\{\dfrac{l_1 + l_2}{2}, \dfrac{l_3 + l_4}{2}\right\}, \\ (s_1, s_2) = \arg\min\left\{\dfrac{l_1 + l_2}{2}, \dfrac{l_3 + l_4}{2}\right\}, \\ r = \dfrac{\sum_{i=1}^{4} l_i}{2h} - 1. \end{cases} \tag{9}$$

Take the quadrilateral in Fig. 8 as instance,

$$\begin{cases} h = \dfrac{l_3 + l_4}{2}, \\ (s_1, s_2) = (l_3, l_4), \\ \alpha = \operatorname{atan}\left(\dfrac{v_y}{v_x}\right). \end{cases} \tag{10}$$

where,

$$(v_x, v_y) = \frac{P_2 + P_3}{2} - \frac{P_1 + P_4}{2}. \tag{11}$$

Quadrilaterals are separated into three shape categories, which are horizontal, vertical and character, based on the value of r and α. Conditions to distinguish categories are listed Algorithm. 2.

After introducing quadrilateral attributes, the line-level post-processing which contains Redundancy Suppression, Segments Grouping, Vertexes Determination and NMS, are illustrated below.

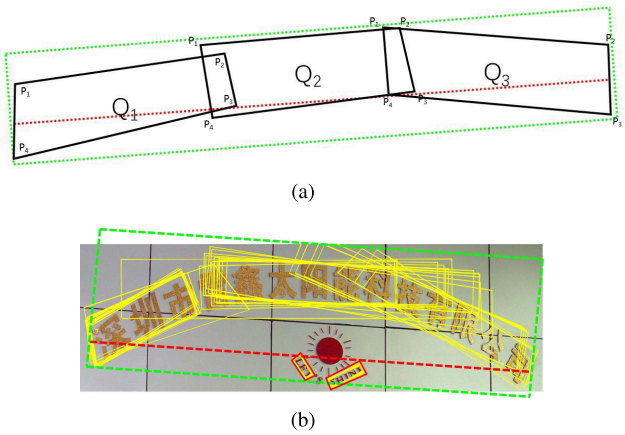


Fig. 9. (a) Line segments grouping and vertexes determination. (b) Grouping results on curved text line. Yellow quadrilaterals are the line-segments predicted by the network, green dashed quadrilaterals are correctly detected text lines, and red quadrilaterals are false positive. The grouping performance relies on the line-segments predicted by the network.

a) Redundancy suppression: Considering the severe redundancy of line-segments and high complexity in computing overlapped area between quadrilaterals, we firstly perform a fast-NMS based on the distance within vertexes. Here we define the relative distance between two quadrilaterals Q_1 and Q_2 as

$$\mathcal{D}(Q_1, Q_2) = \frac{\sum_{i=1}^{4} \|Q_1.P_i - Q_2.P_i\|_2}{\sum_{i=1}^{4} (Q_1.l_i + Q_2.l_i)}. \quad (12)$$

If $\mathcal{D}(Q_1, Q_2) < 0.01$, we assume Q_1 is highly overlapped with Q_2, and if the score of Q_1 is higher than that of Q_2, we remove Q_2, and vice versa. Nearly 95% quadrilateral candidates could be removed by fast-NMS.

b) Segments grouping: This step is to cluster quadrilaterals into different groups based on spatial relationships. If the shape category of a quadrilateral is character, it won't be grouped with any other quadrilaterals. Otherwise, only quadrilaterals that have identical shape category could be grouped together. Suppose Q_1 and Q_2 are both horizontal, and $\frac{\sum_{i=1}^{4} Q_1.P_i}{4}$ is on the left side of $\frac{\sum_{i=1}^{4} Q_2.P_i}{4}$ as shown in Fig. 9.a. Other conditions that should be satisfied to group two quadrilateral Q_1 and Q_2 are listed below, and vertical case is similar by rotating Q_1 and Q_2 by 90°.

- $Q_1.h/Q_2.h \in [4/5, 5/4]$
- $Q_2.P_1.x + Q_2.P_4.x > Q_1.P_1.x + Q_1.P_4.x$
- $Q_2.P_1.x + Q_2.P_4.x < Q_1.P_2.x + Q_1.P_3.x$
- $\frac{\min\{\|Q_1.P_2 - Q_2.P_4\|_2, \|Q_1.P_3 - Q_2.P_1\|_2\}}{\max\{\|Q_1.P_2 - Q_2.P_4\|_2, \|Q_1.P_3 - Q_2.P_1\|_2\}} > 0.8$

c) Vertexes determination: After line-segments grouping, we suppose each group contains only one text line, and then the four vertexes of this text line should be determined. Take Fig. 9.a as instance. Firstly, link the line between center points of sides at either end and denote it as the orientation line. Secondly, translate the orientation line vertically (horizontally for vertical cases) until it has no intersection point with the line-segments in group. And then we determine the upper and lower (left and right for vertical cases) boundaries of the final text line. Thirdly, make up the rest two text line boundaries which are vertical with those in the second step. It should be noticed that our line grouping method could also deal with slightly curved text lines as shown in Fig. 9.b if line-segments are predicted properly.

d) NMS: The final step is a traditional NMS with 0.5 overlap threshold, and the confidence score of text line is the mean value of quadrilateral scores in the same group.

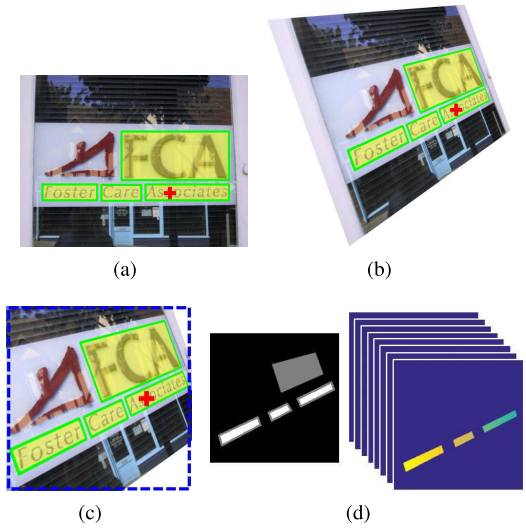


Fig. 10. Region of Interest sampling. (a) PoI (red cross) in text region is selected. (b) Perspective distortion for original image. (c) Image patch containing interest point is cropped out. (d) Ground truths are generated. The gray region that encloses white region is the transition boundary.

F. Data Augmentation

Data augmentation is essential to improve both robustness and generalization of deep neural models, as well as save much human effort in data collection. To repeatedly and effectively make use of each training sample, we propose a data augmentation strategy called region of interest (RoI) sampling. Given an image in the training dataset, first randomly select a point of interest (PoI) in text region. Text that contains this PoI is called text of interest (ToI). Then perform perspective distortion to the original image and guarantee that the shortest side length of ToI falls in $[32 \times 2^{-0.8}, 32 \times 2^{0.8}]$. Finally a patch of 320×320 is cropped from the whole image with the ToI randomly placed in it. A visualized RoI sampling procedure is displayed in Fig. 10. For each iteration, samples containing texts are disposed in this manner, and thus training samples could own abundant variation.

IV. Experiments

A. Implementation Details

In training stage, the network is optimized by stochastic gradient descent (SGD) with back-propagation, and the max iteration is 3×10^5. We adopt the "multistep" strategy in Caffe [16] to adjust learning rate. For the first 3×10^4 iterations the learning rate is fixed to be 10^{-2}, and after that it is reduced to 10^{-3} until the 10^5th iteration. For the rest 10^5 iterations,

the learning rate keeps 10^{-4}. Apart from adjusting learning rate, the hard negative ratio in OHNM is increased from 0.2 to 0.7, and the task balance index λ_{loc} is raised from 0.01 to 0.5 at the 3×10^4th iteration. The weight decay, momentum, positive and negative ratio are fixed to be 4×10^{-4}, 0.9 and 1, respectively. All layers are initialized by "xavier" [8]. In test stage, we adopt a multi-scale sliding window strategy in which the window size is 640×640, the sliding stride is 480, and the multi-scale set is $\{2^{-3}, 2^{-2}, \cdots, 2^1\}$. Pixels on $\mathcal{M}_{cls}$ are deemed as text if their values are higher than 0.9.

The model is optimized on training datasets of ICDAR2013 and ICDAR2015 for word-level detection model. As for Chinese, we adopt training data from RCTW-17 and CASIA-10K. To better explore the training data, we also cover all the text regions to generate pure non-text samples for training. The whole experiments are conducted on Caffe and run on a workstation with 2.9GHz 12-core CPU, 256G RAM, GTX Titan X and Ubuntu 64-bit OS.

B. Datasets

We evaluated the proposed method on various datasets containing multi-lingual texts, including our own dataset of Chinese text images.

1) ICDAR2015 Incidental Text Dataset: This dataset contains 1000 training images and 500 test images. This dataset contains texts with various scales, blurring, orientations and viewpoints. The annotation of bounding box (actually quadrilateral) has 8 coordinates of four corners in a clock-wise manner. In evaluation stage, word-level predictions are required.

2) MSRA-TD500: This dataset contains 300 training images and 200 test images, where there are many multi-oriented text lines. Texts in this dataset are stably captured with high resolution and are bi-lingual of both English and Chinese. In evaluation stage, line-level predictions are required.

3) ICDAR2013 Focused Text Dataset: This dataset lays more emphasis on horizontal scene texts. It contains 229 training images and 233 test images which are well captured and clear. The evaluation protocol is introduced in [12].

4) RCTW-17: Reading Chinese Text in the Wild (RCTW-17) is a large-scale dataset that contains various types of images, including street views, posters, menus, indoor scenes and screen-shots. There are 8346 training images and 4229 test images. Like ICDAR2015, for each text line, 8 coordinates of a quadrilateral are annotated. In evaluation stage, line-level predictions are required.

5) CASIA-10K: CASIA-10K is a Chinese scene text dataset provided by ourselves. This dataset contains 10000 images under various scenarios, in which 7000 images are for training and 3000 images are for testing. For each text line, 8 coordinates of a quadrilateral are annotated. In evaluation stage, line-level predictions are required.

6) MLT-17: Multi-lingual scene text detection (MLT-17)[1] deals with various scripts and languages in natural scenes. The dataset is composed of complete scene images which come from 9 languages representing 6 different scripts. There are 7200 training images, 1800 validation images and 9000 test images. In evaluation stage, languages like English, French, Arabic and Bangla are annotated in word-level, while languages like Chinese, Japanese and Korean are annotated in line-level.

TABLE I

COMPARISON OF METHODS ON ICDAR2015 INCIDENTAL SCENE TEXT DATASET. RESULTS OF THE PROPOSED METHOD WITH THREE FEATURE EXTRACTION STRUCTURES ARE LISTED

Algorithm	Precision	Recall	F-measure
Proposed (VGG-16)	0.85	**0.80**	**0.82**
Proposed (S-VGG)	0.84	0.79	0.81
Zhou *et al.* [50]	0.83	0.78	0.81
Proposed (ResNet-50)	**0.89**	0.73	0.80
Shi *et al.* [33]	0.73	0.77	0.75
Liu *et al.* [24]	0.73	0.68	0.71
Tian *et al.* [39]	0.74	0.52	0.61
Zhang *et al.* [48]	0.71	0.43	0.54
StradVision2 [18]	0.77	0.37	0.50
StradVision1 [18]	0.53	0.46	0.50
NJU-Text [18]	0.70	0.36	0.47
AJOU [18]	0.47	0.47	0.47
HUST_MCLAB [18]	0.44	0.38	0.41

C. Experimental Results

1) ICDAR2015 Incidental Text Dataset: The proposed method reaches the state-of-the-art performance on this dataset as shown in Tab. I. From Tab. I we can see the results from VGG-16 and S-VGG are both better than that from ResNet-50. One reason is that ResNet-50 may be too deep that the model gets worse generalization on scene text detection. In Section. IV.D, more analysis will be provided. Part of our detection results are shown in Fig. 11.

2) MSRA-TD500: We provide two groups of results on this dataset as shown in Tab. II, with comparisons to other representative methods. The first group of results which surpasses previous ones by a large margin are gotten by adding samples from CASIA-10K for training. The second group of results are gotten by only using training samples in MSRA-TD500, and for better fine-tuning on limited training data, we set the learning rate to be 10^{-5} and stop the model from learning regression task. Since our method treats text lines as an object, text lines with wide character space could also be detected. Part of our detection results are shown in Fig. 12.

3) ICDAR2013 Focused Text Dataset: Our method reaches the state-of-the-art performance on ICDAR2013 dataset as shown in Tab. III. Failed cases are mainly caused by single character text and overlapped text lines, both of which are rarely seen in training data. Part of our detection results are shown in Fig. 13.

4) RCTW-17: The detection results of our method on RCTW-17 dataset are shown in Tab. IV [34].[2] Our method ranks the second place, but is still competitive to the state-of-the-art with a gap of 0.35% under F-measure. Failed cases

[1] http://rrc.cvc.uab.es/?ch=8&com=introduction

[2] http://mclab.eic.hust.edu.cn/icdar2017chinese/result.html

Fig. 11. Detection examples of our model on ICDAR2015 Incidental Scene Text benchmark. Green region means true positive and red region means false positive.

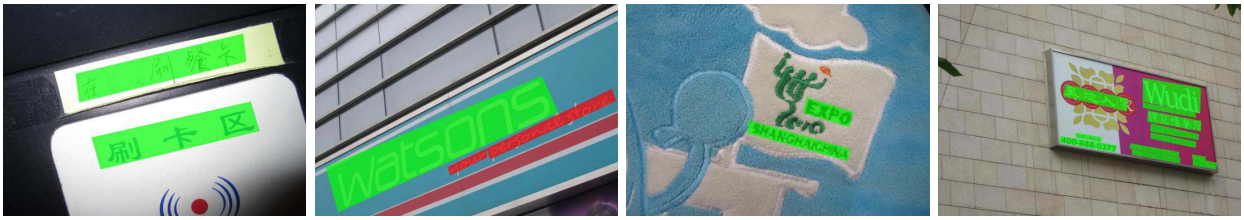


Fig. 12. Detection examples of our model on MSRA-TD500. Green region means true positive and red region means false positive. Our model could handle various fonts and long multi-oriented texts. text lines with wide character space could also be localized well. False cases are mainly caused by line segments grouping.

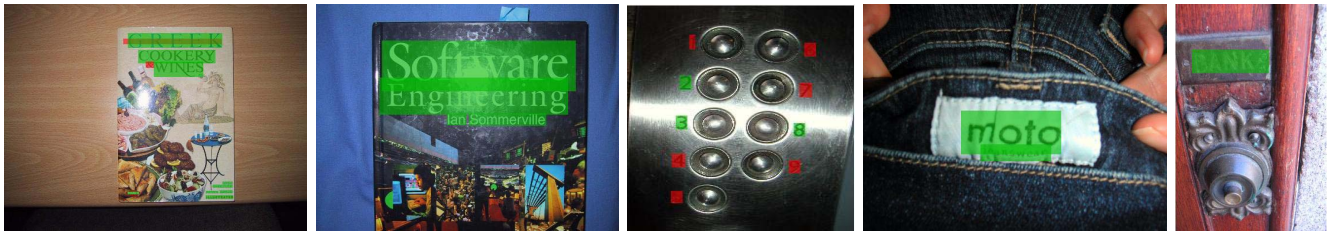


Fig. 13. Detection examples of our model on ICDAR2013. Green region means true positive and red region means false negative. Most failed cases are single characters and overlapped text lines.

TABLE II

COMPARISON OF METHODS ON MSRA-TD500 DATASET. PROPOSED* MEANS USING DATA FROM CASIA-10K FOR TRAINING

Algorithm	Precision	Recall	F-measure
Proposed*	**0.91**	**0.81**	**0.86**
Shi *et al.* [33]	0.86	0.70	0.77
Proposed	0.85	0.70	0.76
Zhou *et al.* [50]	0.87	0.67	0.76
He *et al.* [12]	0.77	0.70	0.74
Zhang *et al.* [48]	0.83	0.67	0.74
Yin *et al.* [45]	0.81	0.63	0.71
Kang *et al.* [17]	0.71	0.62	0.66
Yao *et al.* [42]	0.63	0.63	0.60

TABLE III

COMPARISON OF METHODS ON ICDAR2013 FOCUSED SCENE TEXT DATASET

Algorithm	Precision	Recall	F-measure
Proposed	**0.95**	**0.89**	**0.91**
He *et al.* [12]	0.92	0.81	0.86
Shi *et al.* [33]	0.88	0.83	0.85
Liao *et al.* [21]	0.88	0.83	0.85
Zhang *et al.* [48]	0.88	0.78	0.83
He *et al.* [11]	0.93	0.73	0.82
Tian *et al.* [38]	0.85	0.76	0.80

are mainly caused by incorrect text line partition and curved text lines. It should be noticed that, Chinese text lines are separated by semantic meanings rather than word space, and thus recognition information is inevitable for more precise partition. Part of our detection results are shown in Fig. 14 together with results for CASIA-10K.

5) CASIA-10K: Our method on the newly proposed CASIA-10K dataset is taken as the baseline. For better

Fig. 14. Detection examples of our model on RCTW-17 and CASIA-10K. Green region means true positive, red region means false positive and yellow region (quadrilateral) means ground truth. The first row displays the capability of our model in dealing with complex background, long and vertical text lines. The rest images display false cases in text line partition and localizing curved text lines.

TABLE IV
COMPARISON OF METHODS ON ICDAR2017 READING CHINESE TEXT IN THE WILD

Team Name	Precision	Recall	F-measure
Foo & Bar	0.7439	**0.5948**	**0.6611**
NLPR_PAL (Proposed)	0.7717	0.5729	0.6576
gmh	0.7064	0.5784	0.6360
SCUT_MBCNN	0.7361	0.5184	0.6084
IVA	0.6610	0.5522	0.6017
CCFLAB	0.7406	0.4713	0.5760
CAS_HotEye	**0.7915**	0.4417	0.5670
Baseline [33]	0.7603	0.4044	0.5278
XMU_SuperLab	0.7222	0.4133	0.5258
Image Search Team	0.6544	0.3996	0.4962
SCUT_DLVC	0.7058	0.3656	0.4817

TABLE V
COMPARISON OF METHODS ON CASIA-10K DATASET

Algorithm	Precision	Recall	F-measure
Proposed	**0.8128**	**0.7048**	**0.7550**
SegLink	0.7275	0.6967	0.7118
EAST	0.7771	0.5327	0.6321

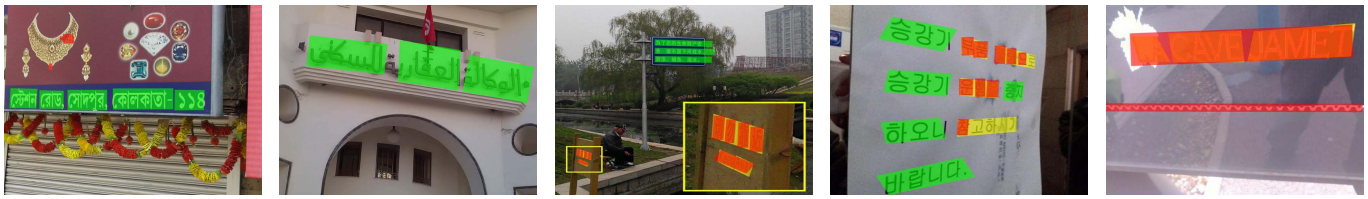
TABLE VI
RESULTS OF VALIDATION AND TEST SETS ON ICDAR2017 COMPETITION OF MULTI-LINGUAL SCENE TEXT DETECTION AND SCRIPT IDENTIFICATION. OTHER METHODS ARE ALL EVALUATED ON TEST SET

Method	Precision	Recall	F-measure
Proposed (Validation)	0.8266	0.7253	0.7726
Proposed (Test)	0.7669	0.5794	**0.6601**
SCUT_DLVClab	**0.8028**	0.5454	0.6496
Sensetime_OCR	0.5693	**0.6943**	0.6256
SARI_FDU_RRPN_v1	0.7117	0.5550	0.6237
TH-DL	0.6775	0.3478	0.4597
linkage-ER-Flow	0.4448	0.2559	0.3249
IDST_CV	0.3181	0.2602	0.2863

comparison, we also test the performance of other two recent text detection methods which are EAST [50] and SegLink [33] respectively on this dataset. For the EAST, we adopt the post-processing in this paper which is more adaptive for line-level annotation. As for SegLink, we re-implement it by introducing transition boundary and positive text scale to the ground truth design rather than directly using the source code on Github.[3] The results on CASIA-10K are listed in Tab. V, and our method gets the best F-measure score mainly due to two reasons. The first one is that our method depicts the whole text boundaries rather than part shape of the text line, and thus gives more precise boundary than SegLink does. The second reason is that our work lays more emphasis on robust text feature extraction as shown Section. IV.D, and thus both our method and SegLink outperform EAST.

6) MLT-17: The results on both validation and test dataset of MLT-17 are listed in Tab. VI [27]. The test set is more

[3]https://github.com/bgshih/seglink



(a) (b) (c) (d) (e)

Fig. 15. Detection examples of our model on MLT-17. Green region means true positive, red region means false positive and yellow region means ground truth.

complicated than validation set, and thus gives a lower F-measure score. Since it is a multi-lingual dataset, the annotation contains both word-level (Latin, Arabic, et cl.) and line-level (Chinese, Japanese, Korean, et cl.). Considering that the line-level post-processing could also realize word partition if candidate quadrilaterals are separated well, here we adopt line-level post-processing for the MLT-17 dataset. By comparing with the other results, our method achieves the state-of-the-art performance. Part of our detection results are shown in Fig. 15.

Our method could deal with both word-level and line-level annotation, however, two new problems arise for multi-lingual scene text detection. Firstly, mixing word-level and line-level annotation could lead confusion in word partition. In Fig. 15.c and Fig. 15.d, characters are detected separately, which is rarely encountered in RCTW-17 and CASIA-10K. In Fig. 15.e, Latin words which should be separated are grouped together. The second problem for multi-lingual text detection is that mixed lingual data leads models prone to detecting text-alike regions as shown in Fig. 15.a and Fig. 15.e. This problems is mainly caused by the increasing inner-class variance since text features are extracted from multiple rather than single language. An intuitive solution for both problems is to add a script identification task and perform specific post-processing strategies for each language.

D. Ablation Experiments

We run a number of ablations to analyze several factors in designing scene text detection models. All the ablation experiments are performed on ICDAR2015 Incidental Scene Text dataset, and for simplicity we encode each experiment in $X_1X_2X_3$ form. The meaning of each X_i is explained below:

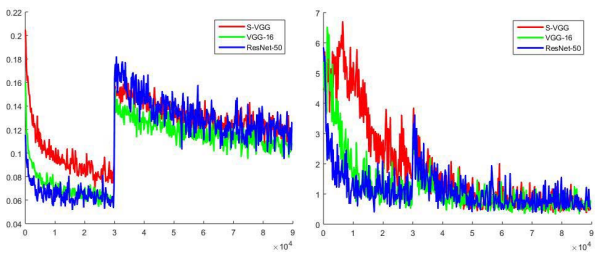
- $X_1 \in \{A, B, C\}$, where A is S-VGG, B is VGG-16, C is ResNet-50;
- $X_2 \in \{0, 1\}$, where 0 means no data augmentation, 1 means using data augmentation;
- $X_3 \in \{0, 1\}$, where 0 means not using OHNM, 1 means using OHNM;

1) Feature Extraction Architecture: Under the same setting for X_2, X_3, we compare the performance under three kinds of architectures illustrated in Section. III.A. From Tab. VII, we note that deeper model which achieves higher performance on ImageNet may not perform better. ResNet-50 gives lower F-measure scores than S-VGG and VGG-16. We assume that this might be caused by the over-fitting problem, since text and non-text classification is a much easier problem or lays emphasis on other factors like text recognition information which a single classification task could not handle well with. The performance gap between VGG-16 and S-VGG is also not large, and false cases are similar indicating that scene text detection may require more reflection on other aspects instead of stacking layers.

TABLE VII

RESULTS OF ABLATION EXPERIMENTS ON ICDAR2015 INCIDENTAL SCENE TEXT DATASET

Mode	Precision	Recall	F-measure
A11	0.8534	0.7540	0.8006
B11	0.8538	0.7959	0.8238
C11	0.8776	0.7285	0.7961
A01	0.8373	0.7877	0.8117
B01	0.8120	0.8089	0.8104
C01	0.8272	0.7516	0.7876
A10	0.6879	0.6548	0.6709
B10	0.6423	0.5715	0.6048
C10	0.5701	0.7737	0.6565



(a) (b)

Fig. 16. Multi-task loss convergence of different feature extraction structures. (a) Classification loss curves. (b) Regression loss curves.

One obvious difference between random initialization and fine-tuning is that the losses for both tasks converge faster under fine-tuning mode, even though all the losses would be similar finally. Loss convergence of different architectures is shown in Fig. 16.

2) Data Augmentation: The data augmentation here refers RoI sampling, and if data augmentation is not adopted, image patches should be cropped beforehand, which introduces much human effort in sorting training samples and also limits the using of training data. From Tab. VII, when fixing X_1X_3, most results given by using data augmentation are better. Although the improvement is not remarkable, the RoI sampling strategy still facilitates the training sample generation.

TABLE VIII

RESULTS ON LARGER POSITIVE SCALE. "X-LG" REFERS MODEL TRAINED ON LARGER POSITIVE SCALE

Mode	Precision	Recall	F-measure	Gap
AA11	0.8545	0.7540	0.8006	-
AA11-LG	0.8297	0.6052	0.6999	0.1007 ↓
BA11	0.8538	0.7959	0.8238	-
BA11-LG	0.8471	0.7121	0.7737	0.0501 ↓
CA11	0.8776	0.7285	0.7961	-
CA11-LG	0.8563	0.6172	0.7174	0.0787 ↓

TABLE IX

RESULTS WITHOUT TRANSITION BOUNDARY. "X-NN" REFERS MODELS HAVING NO TRANSITION BOUNDARY

Mode	Precision	Recall	F-measure	Gap
AA11	0.8545	0.7540	0.8006	-
AA11-NN	0.8323	0.7145	0.7689	0.0317 ↓
BA11	0.8538	0.7959	0.8238	-
BA11-NN	0.7887	0.8016	0.7951	0.0287 ↓
CA11	0.8776	0.7285	0.7961	-
CA11-NN	0.8471	0.7097	0.7723	0.0238 ↓

3) Effectiveness of OHNM: The OHNM is intended to deal with the long tail distribution exhibited in non-text region where the majority of non-text could be easily classified. To test the essence of OHNM, we fix X_2X_3 to be 10, and compare the performances on three types of feature extraction structures. As shown in Tab. VII, results given by models without OHNM are far from satisfactory, and consequently OHNM is indispensable in training the scene text detection framework.

4) Adequate Positive Text Scale: In our work, text (positive) regions are determined not only by their physical attribute, but also by their scales. Text regions whose shortest side size falls in $\left[32 \times 2^{-1}, 32 \times 2^{1}\right]$ are taken as positive, otherwise as "NOT CARE" or negative. This restriction differs from previous FCN based methods [43], [48] which take all text regions as positive regardless of text scale.

To demonstrate the essence of positive text scale, we extend the positive scale range from $\left[32 \times 2^{-1}, 32 \times 2^{1}\right]$ to $\left[32 \times 2^{-1}, 32 \times 2^{2}\right]$. The network architecture we use is $X_1 11$. Results listed in Tab. VIII show that extending the positive upper limit pulls down the performance obviously and therefore adequate positive text scale should be carefully given.

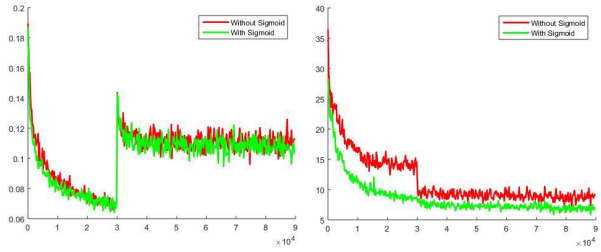
5) Transition Boundary: In our work, positive text region is enclosed by "NOT CARE" boundary named transition boundary. Our explanation is that transition boundary plays a role of transition from positive to negative urging the model to pay attention to worthier non-text regions. To demonstrate the essence of this design, we adopt the architecture $X_1 11$ and replace transition boundary with negative region in training stage for comparison. Results listed in Tab. IX demonstrate that removing transition boundary could cause obvious negative effect on performance.

6) Receptive Field and Input Size: In theory, the receptive field size determines how large the network could "see", and therefore the maximum length that the regression task could measure should be less than the receptive field size. Experimentally we adopt the A11 as the baseline structure and fix the input size to be 320×320. Then we gradually remove the top convolutional layers to shrink the receptive field. Tab. X displays the performance on different receptive field sizes and there is an obvious drop as the network becomes shallower which indicates that it is essential to build a deep enough network to ensure the model could catch global information of the text line.

TABLE X

RESULTS ON DIFFERENT RECEPTIVE FIELD

Receptive Field	Precision	Recall	F-measure
320	0.8545	0.7540	0.8006
224	0.8061	0.7525	0.7784
128	0.6560	0.6996	0.6771



(a) (b)

Fig. 17. The loss convergence for network without sigmoid normalization. (a) Classification loss curves. (b) Regression loss curves.

TABLE XI

RESULTS ON SIGMOID NORMALIZATION. X11-NOSIG INDICATES MODEL WITHOUT SIGMOID NORMALIZATION

Receptive Field	Precision	Recall	F-measure	Gap
A11	0.8534	0.7540	0.8006	-
A11-nosig	0.7845	0.8151	0.7995	0.0001 ↓
B11	0.8538	0.7959	0.8238	-
B11-nosig	0.8064	0.8382	0.8220	0.0018 ↓
C11	0.8776	0.7285	0.7961	-
C11-nosig	0.7596	0.8397	0.7976	0.0015 ↑

7) Sigmoid Normalization: To verify the effectiveness of sigmoid normalization, we analyze the loss convergence by removing the Sigmoid and Scale&Shift layers. The loss convergence for regression task without sigmoid normalization is shown in Fig. 17, the network configuration we used here is A11. The sigmoid normalization, which is embedded in regression task, has little influence on the classification task because of the identical decreasing for both classification losses. While for the regression task, there is an obvious gap between networks with and without sigmoid normalization, and moreover the sigmoid normalization generates more steady losses.

The detection performance without sigmoid normalization is shown in Tab. XI where A11, B11 and C11 are taken into comparison. Results in Tab. XI indicate that the sigmoid normalization has little influence on the final detection performance when models are well trained, while for the sake

of steady loss convergence, sigmoid normalization could be a better choice.

V. Conclusion

In this paper, we presented a FCN based framework for multi-oriented and multi-lingual scene text detection by taking text word or line as an object. The proposed framework involves two prediction tasks: the first classification task performs down-sampled segmentation to roughly localize text regions, and the second regression task determines the text boundaries. Direct regression strategy is chosen for efficient quadrilateral boundary regression. To improve the performance, we designed RoI sampling for data augmentation, introduced positive text scale and transition boundary in ground truth design, on-line hard negative mining to deal with class imbalance, and sigmoid normalization for fast and steady regression loss convergence. Post-processing methods for both word-level and line-level annotation are proposed as well. These strategies were demonstrated effective in experiments. We achieved state-of-the-art performance on multiple datasets, including ICDAR2015, ICDAR2013, MSRA-TD500, MLT-17, and a new large dataset CASIA-10K mainly containing Chinese texts. The ablation experiments show that hard non-text sample mining, positive text scale and transition boundary influence significantly on the text detection performance, while the network structure and fine-tuning are less influential.

A promising future work would be to combine word-level and line-level methods to overcome the deficiency of either group of approaches. Line-level methods may fail to detect characters at either end of long text lines, while character-level methods ignore global information of text lines and require more delicate post-processing. The combination of two groups of methods is expected to yield better performance on scene text detection.

References

[1] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[2] H. Chen, S. S. Tsai, G. Schroth, D. M. Chen, R. Grzeszczuk, and B. Girod, "Robust text detection in natural images with edge-enhanced maximally stable extremal regions," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2011, pp. 2609–2612.

[3] H. Cho, M. Sung, and B. Jun, "Canny text detector: Fast and robust scene text localization algorithm," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 3566–3573.

[4] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object detection via region-based fully convolutional networks," in *Proc. Neural Inf. Process. Syst.*, 2016, pp. 379–387.

[5] P. Dollár, R. Appel, S. Belongie, and P. Perona, "Fast feature pyramids for object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 8, pp. 1532–1545, Aug. 2014.

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

[7] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 1440–1448.

[8] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. Int. Conf. Artif. Intell. Stat.*, 2010, pp. 249–256.

[9] A. Gupta, A. Vedaldi, and A. Zisserman, "Synthetic data for text localisation in natural images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 2315–2324.

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

[11] T. He, W. Huang, Y. Qiao, and J. Yao, "Text-attentional convolutional neural network for scene text detection," *IEEE Trans. Image Process.*, vol. 25, no. 6, pp. 2529–2541, Jun. 2016.

[12] W. He, X.-Y. Zhang, F. Yin, and C.-L. Liu, "Deep direct regression for multi-oriented scene text detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2017, pp. 745–753.

[13] L. Huang, Y. Yang, Y. Deng, and Y. Yu. (2015). "DenseBox: Unifying landmark localization with end to end object detection." [Online]. Available: https://arxiv.org/abs/1509.04874

[14] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

[15] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, "Reading text in the wild with convolutional neural networks," *Int. J. Comput. Vis.*, vol. 116, no. 1, pp. 1–20, 2016.

[16] Y. Jia *et al.*, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. ACM Int. Conf. Multimedia*, 2014, pp. 675–678.

[17] L. Kang, Y. Li, and D. Doermann, "Orientation robust text line detection in natural images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 4034–4041.

[18] D. Karatzas *et al.*, "ICDAR 2015 competition on robust reading," in *Proc. Int. Conf. Document Anal. Recognit.*, 2015, pp. 1156–1160.

[19] T. Kong, A. Yao, Y. Chen, and F. Sun, "HyperNet: Towards accurate region proposal generation and joint object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 845–853.

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[21] M. Liao, B. Shi, X. Bai, X. Wang, and W. Liu, "TextBoxes: A fast text detector with a single deep neural network," in *Proc. AAAI Conf. Artif. Intell.*, 2017, pp. 4161–4167.

[22] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2117–2125.

[23] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, and S. Reed, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, in Lecture Notes in Computer Science, vol. 9905. Amsterdam, The Netherlands: Springer, Oct. 2016, pp. 21–37.

[24] Y. Liu and L. Jin, "Deep matching prior network: Toward tighter multi-oriented text detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 1962–1969.

[25] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 3431–3440.

[26] J. Ma *et al.* (2017). "Arbitrary-oriented scene text detection via rotation proposals." [Online]. Available: https://arxiv.org/abs/1703.01086

[27] N. Nayef *et al.*, "ICDAR2017 robust reading challenge on multi-lingual scene text detection and script identification—RRC-MLT," in *Proc. Int. Conf. Document Anal. Recognit.*, vol. 1, 2017, pp. 1454–1459.

[28] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, Jul. 2002.

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

[30] Y.-F. Pan, C.-L. Liu, and X. Hou, "Fast scene text localization by learning-based filtering and verification," in *Proc. Int. Conf. Image Process.*, 2010, pp. 2269–2272.

[31] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, Jun. 2016, pp. 779–788.

[32] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[33] B. Shi, X. Bai, and S. Belongie, "Detecting oriented text in natural images by linking segments," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2017, pp. 2550–2558.

[34] B. Shi *et al.*, "ICDAR2017 competition on reading chinese text in the wild (RCTW-17)," in *Proc. Int. Conf. Document Anal. Recognit.*, vol. 1, pp. 1429–1434, 2017.

[35] A. Shrivastava, A. Gupta, and R. Girshick, "Training region-based object detectors with online hard example mining," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 761–769.

[36] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent.*, 2015. [Online]. Available: https://arxiv.org/abs/1409.1556
[37] L. Sun, Q. Huo, W. Jia, and K. Chen, "A robust approach for text detection from natural scene images," *Pattern Recognit.*, vol. 48, no. 9, pp. 2906–2920, 2015.
[38] S. Tian, Y. Pan, C. Huang, S. Lu, K. Yu, and C. Lim Tan, "Text flow: A unified text detection system in natural scene images," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 4651–4659.
[39] Z. Tian, W. Huang, T. He, P. He, and Y. Qiao, "Detecting text in natural image with connectionist text proposal network," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 56–72.
[40] C. Wang, F. Yin, and C.-L. Liu, "Scene text detection with novel superpixel based character candidate extraction," in *Proc. Int. Conf. Document Anal. Recognit.*, vol. 1, 2017, pp. 929–934.
[41] X. Wang, Y. Song, Y. Zhang, and J. Xin, "Natural scene text detection with multi-layer segmentation and higher order conditional random field based analysis," *Pattern Recognit. Lett.*, vol. 60, pp. 41–47, Aug. 2015.
[42] C. Yao, X. Bai, W. Liu, Y. Ma, and Z. Tu, "Detecting texts of arbitrary orientations in natural images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 1083–1090.
[43] C. Yao, X. Bai, N. Sang, X. Zhou, S. Zhou, and Z. Cao. (2016). "Scene text detection via holistic, multi-channel prediction." [Online]. Available: https://arxiv.org/abs/1609.03605
[44] Q. Ye and D. Doermann, "Text detection and recognition in imagery: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 7, pp. 1480–1500, Jul. 2015.
[45] X. C. Yin, W. Y. Pei, J. Zhang, and H. W. Hao, "Multi-orientation scene text detection with adaptive clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1930–1937, Sep. 2015.
[46] X.-C. Yin, X. Yin, K. Huang, and H.-W. Hao, "Robust text detection in natural scene images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 5, pp. 970–983, May 2014.
[47] Z. Zhang, W. Shen, C. Yao, and X. Bai, "Symmetry-based text line detection in natural scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 2558–2567.
[48] Z. Zhang, C. Zhang, W. Shen, C. Yao, W. Liu, and X. Bai, "Multi-oriented text detection with fully convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 4159–4167.
[49] Z. Zhong, L. Jin, S. Zhang, and Z. Feng. (2016). "DeepText: A unified framework for text proposal generation and text detection in natural images." [Online]. Available: https://arxiv.org/abs/1605.07314
[50] X. Zhou *et al.*, "EAST: An efficient and accurate scene text detector," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 5551–5560.
[51] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, in Lecture Notes in Computer Science, vol. 8693. Zurich, Switzerland: Springer, Sep. 2014, pp. 391–405.

Wenhao He received the B.S. degree in communication engineering from Beihang University, Beijing, China, in 2013. He is currently pursuing the Ph.D. degree in pattern recognition and intelligent systems with the Institute of Automation, Chinese Academy of Sciences, Beijing. He was a Visiting Researcher at the University of La Rochelle, in 2017. His research interests include pattern recognition, deep learning, computer vision, and scene text detection.

Xu-Yao Zhang received the B.S. degree in computational mathematics from Wuhan University, Wuhan, China, in 2008, and the Ph.D. degree in pattern recognition and intelligent systems from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2013. He was a Visiting Researcher with CENPARMI, Concordia University, in 2012. From 2015 to 2016, he was a Visiting Scholar with the Montreal Institute for Learning Algorithms, University of Montreal, Canada. He is currently an Associate Professor with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences. His research interests include machine learning, pattern recognition, handwriting recognition, and deep learning.

Fei Yin received the B.S. degree in computer science from the Xidian University of Posts and Telecommunications, Xi'an, China, in 1999, the M.E. degree in pattern recognition and intelligent systems from the Huazhong University of Science and Technology, Wuhan, China, in 2002, and the Ph.D. degree in pattern recognition and intelligent systems from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2010. He is currently an Associate Professor with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences. He has authored over 50 papers at international journals and conferences. His research interests include document image analysis, handwritten character recognition, and image processing.

Cheng-Lin Liu (F'15) received the B.S. degree in electronic engineering from Wuhan University, Wuhan, China, in 1989, the M.E. degree in electronic engineering from Beijing Polytechnic University, Beijing, China, in 1992, the Ph.D. degree in pattern recognition and intelligent control from the Chinese Academy of Sciences, Beijing, in 1995. He was a Post-Doctoral Fellow with the Korea Advanced Institute of Science and Technology and later with the Tokyo University of Agriculture and Technology from 1996 to 1999. From 1999 to 2004, he was a Research Staff Member and later a Senior Researcher with the Central Research Laboratory, Hitachi, Ltd., Tokyo, Japan. He is currently a Professor and the Director with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences. He has authored over 200 technical papers at prestigious international journals and conferences. His research interests include pattern recognition, image processing, neural networks, machine learning, and especially the applications to character recognition and document analysis. He is a fellow of the IAPR. He received the IAPR/ICDAR Young Investigator Award of 2005. He is an Associate Editor-in-Chief of *Pattern Recognition* journal and an Associate Editor of *Image and Vision and Computing*, the *International Journal on Document Analysis and Recognition*, and *Cognitive Computation*.