Arbitrary-Oriented Scene Text Detection via Rotation Proposals

Jianqi Ma, Weiyuan Shao, Hao Ye, Li Wang, Hong Wang, Yingbin Zheng, and Xiangyang Xue

***Abstract*—This paper introduces a novel rotation-based framework for arbitrary-oriented text detection in natural scene images. We present the *Rotation Region Proposal Networks*, which are designed to generate inclined proposals with text orientation angle information. The angle information is then adapted for bounding box regression to make the proposals more accurately fit into the text region in terms of the orientation. The *Rotation Region-of-Interest* pooling layer is proposed to project arbitrary-oriented proposals to a feature map for a text region classifier. The whole framework is built upon a region-proposal-based architecture, which ensures the computational efficiency of the arbitrary-oriented text detection compared with previous text detection systems. We conduct experiments using the rotation-based framework on three real-world scene text detection datasets and demonstrate its superiority in terms of effectiveness and efficiency over previous approaches.**

***Index Terms*—Scene text detection, arbitrary oriented, rotation proposals.**

I. Introduction

TEXT detection aims to identify text regions of given images and is an important prerequisite for many multimedia tasks, such as visual classification [1], [2], video analysis [3], [4] and mobile applications [5]. Although there are a few commercial optical character recognition (OCR) systems for documentary texts or internet content, the detection of text in a natural scene image is challenging due to complex situations such as uneven lighting, blurring, perspective distortion, orientation, etc.

In recent years, much attention has been paid to the text detection task (e.g., [6]–[16]). Although these approaches have shown promising results, most of them rely on horizontal or nearly horizontal annotations and return the detection of horizontal regions. However, in real-world applications, a larger number of the text regions are not horizontal, and even applying non-horizontal aligned text lines as the axis-aligned proposals may not be accurate. Thus, the horizontal-specific methods cannot be widely applied in practice.

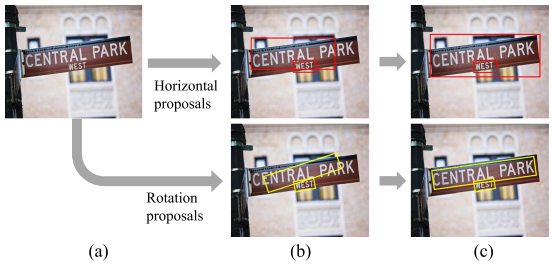


Fig. 1. Overview of text detection. First row: text detection based on horizontal bounding box proposal and bounding box regression of Faster-RCNN [20]. Second row: detection using rotation region proposal and bounding box regression with orientation step. (a) Input image; (b) Initial result; (c) Detection result after refinement.

Recently, a few works have been proposed to address arbitrary-oriented text detection [17]–[19]. In general, these methods mainly involve two steps, i.e., segmentation networks, such as the fully convolutional network (FCN), are used to generate text prediction maps, and geometric approaches are used for inclined proposals. However, prerequisite segmentation is usually time-consuming. In addition, some systems require several post-processing steps to generate the final text region proposals with the desired orientation and are thus not as efficient as those directly based on a detection network.

In this paper, we develop a rotation-based approach and an end-to-end text detection system for arbitrary-oriented text detection. Particularly, orientations are incorporated so that the detection system can generate proposals for arbitrary orientation. A comparison between the previous horizontal-based approach and ours is illustrated in Fig. 1. We present the *Rotation Region Proposal Networks* (RRPN), which are designed to generate inclined proposals with text orientation angle information. The angle information is then adapted for bounding box regression to make the proposals more accurately fit the text region. The *Rotation Region-of-Interest* (RRoI) pooling layer is proposed to project arbitrary-oriented proposals to a feature map. Finally, a two-layer network is deployed to classify the regions as either text or background. The main contributions of this paper include the following:

Manuscript received June 3, 2017; revised November 28, 2017 and January 31, 2018; accepted March 13, 2018. Date of publication March 23, 2018; date of current version October 15, 2018. This work was supported in part by the National Key R&D Program of China under Grant 2017YFC0803700, in part by the National Natural Science Foundation of China under Grants 61602459, 61572138, and U1611461, and in part by the Science and Technology Commission of Shanghai Municipality under Grants 17511101902 and 16JC1420400. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Hantao Liu. *(Jianqi Ma and Weiyuan Shao contributed equally to this work.) (Corresponding author: Yingbin Zheng.)*

J. Ma, L. Wang, and X. Xue are with Shanghai Key Lab of Intelligent Information Processing, School of Computer Science, Fudan University, Shanghai 200433, China (e-mail: majq16@fudan.edu.cn; wangli16@fudan.edu.cn; xyxue@fudan.edu.cn).

W. Shao, H. Ye, H. Wang, and Y. Zheng are with Shanghai Advanced Research Institute, Chinese Academy of Sciences, Shanghai 201210, China (e-mail: shaowy@sari.ac.cn; yeh@sari.ac.cn; wang_hong@sari.ac.cn; zhengyb@sari.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/TMM.2018.2818020

- Different from previous segmentation-based frameworks, ours has the ability to predict the orientation of a text line using a region-proposal-based approach; thus, the proposals can better fit the text region, and the ranged text region can be easily rectified and is more convenient for text reading. New components, such as the RRoI pooling layer and learning of the rotated proposal, are incorporated into the region-proposal-based architecture [20], which ensures the computational efficiency of text detection compared with segmentation-based text detection systems.
- We also propose novel strategies for the refinement of region proposals with arbitrary orientation to improve the performance of arbitrary-oriented text detection.
- We apply our framework to three real-world text detection datasets, i.e., MSRA-TD500 [21], ICDAR2013 [22] and ICDAR2015 [23], and find that it is more accurate and significantly efficient compared to previous approaches.

The rest of this paper is organized as follows. Section II introduces the background of scene text detection and related work. Section III briefly reviews the horizontal region proposal approach. Section IV discusses our framework in detail. In Section V, we demonstrate the quantitative study on three datasets. We conclude our work in Section VI.

II. Related Work

The reading of text in the wild has been studied over the last few decades; comprehensive surveys can be found in [24]–[27]. Methods based on the sliding window, connected components and the bottom-up strategy are designed to handle horizontal-based text detection. Sliding window-based methods [7], [10], [28]–[30] tend to use a sliding window of a fixed size to slide the text area and find the region most likely to include text. To consider more precise styles of text, [10], [31] apply multiple scales and ratios to the sliding window methods. However, the sliding window process leads to a large computational cost and inefficiency. Representative connected-component-based approaches such as the Stroke Width Transform (SWT) [32] and Maximally Stable Extremal Regions (MSER) [33] exhibited superior performances in the ICDAR 2011 [34] and ICDAR 2013 [22] robust text detection competitions. They mainly focus on the edge and pixel point of an image by detecting the character via edge detection or extreme region extraction and then combining the sub-MSER components into a word or text-line region. The capabilities of these methods are limited in some difficult situations involving multiple connected characters, segmented stroke characters and non-uniform illumination [35].

Scene text in the wild is usually aligned from any orientation in real-world applications, and approaches for arbitrary orientations are needed. For example, [36] uses mutual magnitude symmetry and gradient vector symmetry to identify text pixel candidates regardless of the orientation, including curves from natural scene images, and [37] designs a Canny text detector by taking the similarity between an image edge and text to detect text edge pixels and perform text localization. Recently, convolution-network-based approaches were proposed to perform text detection, e.g., Text-CNN [38], by first using an optimized MSER detector to find the approximate region of the text and then sending region features into a character-based horizontal text CNN classifier to further recognize the character region. In addition, the orientation factor is adopted in the segmentation models developed by Yao *et al.* [18]. Their model aims to predict more accurate orientations via an explicit manner of text segmentation and yields outstanding results on the ICDAR2013 [22], ICDAR2015 [23] and MSRA-TD500 [21] benchmarks.

A technique similar to text detection is generic object detection. The detection process can be made faster if the number of proposals is largely reduced. There is a wide variety of region proposal methods, such as Edge Boxes [39], Selective Search [40], and Region Proposal Networks (RPNs) [20]. For example, Jaderberg *et al.* [41] extends the region proposal method and applies the Edge Boxes method [39] to perform text detection. Their text spotting system achieves outstanding results on several text detection benchmarks. The Connectionist Text Proposal Network (CTPN) [42] is also a detection-based framework for scene text detection. It employs the image feature from the CNN network in LSTM to predict the text region and generate robust proposals.

This work is inspired by the RPN detection pipeline in regards to the dense-proposal based approach used for detection and RoI pooling operation used to further accelerate the detection pipeline. Detection pipelines based on RPN are widely used in various computer vision applications [43]–[45]. The idea is also similar to that of Spatial Transformer Networks (STN) [46], i.e., a neural network model can rectify an image by learning its affine transformation matrix. Here, we try to extend the model to multi-oriented text detection by injecting angle information. Perhaps the work most related to ours is [43], where the authors proposed an inception-RPN and made further text detection-specific optimizations to adapt the text detection. We incorporate the rotation factor into the region proposal network so that it is able to generate arbitrary-oriented proposals. We also extend the RoI pooling layer into the Rotation RoI (RRoI) pooling layer and apply angle regression in our framework to perform the rectification process and finally achieve outstanding results.

III. Horizontal Region Proposal

We begin with a brief review of RPN [20]. As mentioned in the previous section, an RPN is able to further accelerate the process of proposal generation. Part of VGG-16 [47] is employed as sharable layers, and the horizontal region proposals are generated by sliding over the feature map of the last convolutional layer. The features extracted from each sliding window are fed into two sibling layers (a box-regression (*reg*) layer and a box-classification (*cls*) layer), with 4 k (4 coordinates per proposal) outputs from the *reg* layer representing coordinates and 2k (2 scores per proposal) scores from the *cls* layer for k anchors of each sliding position.

To fit the objects to different sizes, the RPN uses two parameters to control the size and shape of anchors, i.e., scale and aspect ratio. The scale parameter determines the size of the anchor, and the aspect ratio controls the ratio of the width to the height for the anchor box. In [20], the authors set the scale as

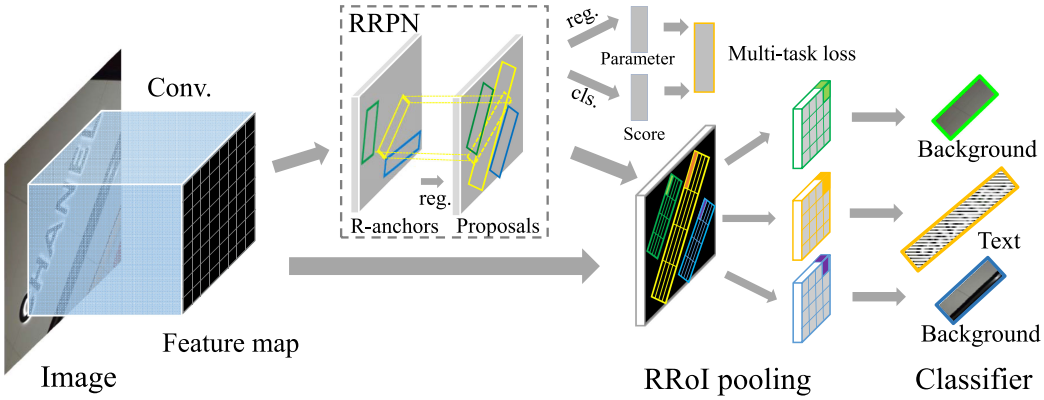


Fig. 2. Rotation-based text detection pipeline.

8, 16 and 32 and the ratio as 1:1, 1:2 and 2:1 for a generic object detection task. This anchor selection strategy can cover the shapes of nearly all natural objects and keep the total number of proposals low. However, in the text detection task, especially for scene images, texts are usually presented in an unnatural shape with different orientations; axis-aligned proposals generated by RPN are not robust for scene text detection. To make a network more robust for text detection and maintain its efficiency, we think that it is necessary to build a detection framework, which encodes the rotation information with the region proposals.

IV. Approach

We now elaborate the construction of the rotation-based framework; the architecture is illustrated in Fig. 2. We employ the convolutional layers of VGG-16 [47] in the front of the framework, which are shared by two sibling branches, i.e., the RRPN and a clone of the feature map of the last convolutional layer. The RRPN generates arbitrary-oriented proposals for text instances and further performs bounding box regression for proposals to better fit the text instances. The sibling layers branching out from the RRPN are the classification layer (*cls*) and the regression layer (*reg*) of the RRPN. The outputs from these two layers are the scores from the *cls* and proposal information from the *reg*, and their losses are computed and summed to form a multitask loss. Then, the RRoI pooling layer acts as a max pooling layer by projecting arbitrary-oriented text proposals from the RRPN onto the feature map. Finally, a classifier formed by two fully connected layers is used, and the region with the RRoI features is classified as either text or background.

A. Rotated Bounding Box Representation

In the training stage, the ground truth of a text region is represented as rotated bounding boxes with 5 tuples (x, y, h, w, θ). The coordinate (x, y) represents the geometric center of the bounding box. The height h is set as the short side of the bounding box, and the width w, as the long side. The orientation θ is the angle from the positive direction of the x-axis to the direction parallel to the long side of the rotated bounding box. Because of the special ability of scene text detection, the direction of reading and its opposite do not influence the detected region. Here, we simply maintain the orientation parameter θ such that it covers half the angular space. Suppose the orientation of a rotated box is θ; there exists one and only one integer k ensuring that $\theta + k\pi$ is within the interval $[-\frac{\pi}{4}, \frac{3\pi}{4})$, and we update $\theta + k\pi$ as θ. There are three benefits of the tuple representation (x, y, h, w, θ). First, it is easy to calculate the angle difference between two different rotated boxes. Second, this is a rotation-friendly representation for the angle regression of each rotated bounding box. Third, compared with the traditional 8-point representation $(x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4)$ of a rotated bounding box, this representation can be used to easily calculate the new ground truth after we rotate a training image.

Suppose the size of a given image is $I_H \times I_W$ and the original text region is represented as (x, y, h, w, θ). If we rotate the image by an angle $\alpha \in [0, 2\pi)$ around its center, the center of the anchor can be calculated as

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \mathbf{T}\left(\frac{I_W}{2}, \frac{I_H}{2}\right)\mathbf{R}(\alpha)\mathbf{T}\left(-\frac{I_W}{2}, -\frac{I_H}{2}\right)\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (1)$$

where $\mathbf{T}$ and $\mathbf{R}$ are the translation matrix and rotation matrix, respectively,

$$\mathbf{T}(\delta_x, \delta_y) = \begin{bmatrix} 1 & 0 & \delta_x \\ 0 & 1 & \delta_y \\ 0 & 0 & 1 \end{bmatrix} \quad (2)$$

$$\mathbf{R}(\alpha) = \begin{bmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3)$$

The width w' and height h' of the rotated bounding box do not change, and the orientation is $\theta' = \theta + \alpha + k\pi$ ($\theta' \in [-\frac{\pi}{4}, \frac{3\pi}{4})$). We employed this image rotation strategy for data augmentation during training.

B. Rotation Anchors

Traditional anchors, which use scale and aspect ratio parameters, are not sufficient for in-the-wild text detection. Therefore, we design the rotation anchors (R-anchors) by making several adjustments. First, an orientation parameter is added to control the orientation of a proposal. Six different orientations, i.e., $-\frac{\pi}{6}$,

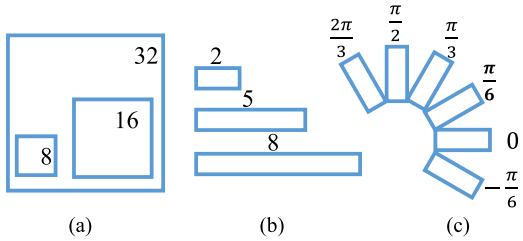


Fig. 3. Anchor strategy used in our framework. (a) Scale; (b) Ratio; (c) Angle.

0, $\frac{\pi}{6}$, $\frac{\pi}{3}$, $\frac{\pi}{2}$ and $\frac{2\pi}{3}$, are used, which are trade-offs between orientation coverage and computational efficiency. Second, as text regions usually have special shapes, the aspect ratio is changed to 1:2, 1:5 and 1:8 to cover a wide range of text lines. In addition, the scales of 8, 16 and 32 are kept. The anchor strategy is summarized in Fig. 3. Following our data representation step, a proposal is generated from the R-anchors with 5 variables (x, y, h, w, θ). For each point on the feature map, 54 R-anchors (6 orientations, 3 aspect ratios, and 3 scales) are generated, as well as 270 outputs (5×54) for the *reg* layer and 108 score outputs (2×54) for the *cls* layer at each sliding position. Then, we slide the feature map with the RRPN and generate $H \times W \times 54$ anchors in total for the feature map, with width W and height H.

C. Learning of Rotated Proposal

As the R-anchors are generated, a sampling strategy for the R-anchors is needed to perform network learning. We first define the intersection-over-union (IoU) overlap as the overlap between the skew rectangles of the ground truth and R-anchor. Then, positive R-anchors feature the following: (i) the highest IoU overlap or an IoU larger than 0.7 with respect to the ground truth, and (ii) an intersection angle with respect to the ground truth of less than $\frac{\pi}{12}$. Negative R-anchors are characterized by the following: (i) an IoU lower than 0.3, or (ii) an IoU larger than 0.7 but with an intersection angle with a ground truth larger than $\frac{\pi}{12}$. Regions that are not selected as either positive or negative are not used during training.

Our loss function for the proposal takes the form of multitask loss [48], which is defined as:

$$L(p, l, v^*, v) = L_{\text{cls}}(p, l) + \lambda l L_{\text{reg}}(v^*, v) \tag{4}$$

where l is the indicator of the class label ($l = 1$ for text and $l = 0$ for background; no regression for the background), the parameter $p = (p_0, p_1)$ is the probability over classes computed by the softmax function, $v = (v_x, v_y, v_h, v_w, v_\theta)$ denotes the predicted tuple for the text label, and $v^* = (v_x^*, v_y^*, v_h^*, v_w^*, v_\theta^*)$ denotes the ground truth. The trade-off between two terms is controlled by the balancing parameter λ. We define the classification loss for class l as:

$$L_{\text{cls}}(p, l) = -\log p_l \tag{5}$$

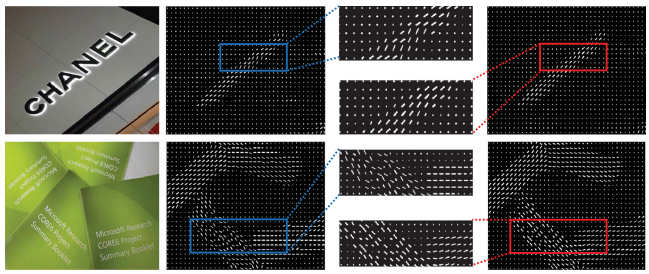


Fig. 4. Visualization of the impact on regression: input images (a); orientation and response of the anchors without regression term (b) and with regression (c). The orientation of the R-anchor is the direction of the white line at each point, with longer lines indicating a higher response score for text.

For the bounding box regression, the background RoIs are ignored, and we adopt smooth-L_1 loss for the text RoIs:

$$L_{\text{reg}}(v^*, v) = \sum_{i \in \{x,y,h,w,\theta\}} \text{smooth}_{L_1}(v_i^* - v_i) \tag{6}$$

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases} \tag{7}$$

The scale-invariant parameterizations tuple v and v^* are calculated as follows:

$$\begin{aligned} v_x &= \frac{x - x_a}{w_a}, v_y = \frac{y - y_a}{h_a} \\ v_h &= \log \frac{h}{h_a}, v_w = \log \frac{w}{w_a}, v_\theta = \theta \ominus \theta_a \end{aligned} \tag{8}$$

$$\begin{aligned} v_x^* &= \frac{x^* - x_a}{w_a}, v_y^* = \frac{y^* - y_a}{h_a} \\ v_h^* &= \log \frac{h^*}{h_a}, v_w^* = \log \frac{w^*}{w_a}, v_\theta^* = \theta^* \ominus \theta_a \end{aligned} \tag{9}$$

where x, x_a and x^* are for the predicted box, anchor and ground truth box, respectively; the same is for y, h, w and θ. The operation $a \ominus b = a - b + k\pi$, where $k \in Z$ to ensure that $a \ominus b \in [-\frac{\pi}{4}, \frac{3\pi}{4})$.

As described in the previous section, we give R-anchors fixed orientations within the range $[-\frac{\pi}{4}, \frac{3\pi}{4})$, and each of the 6 orientations can fit the ground truth that has an intersection angle of less than $\frac{\pi}{12}$. Thus, every R-anchor has its fitting range, which we call its fit domain. When an orientation of a ground truth box is in the fit domain of an R-anchor, this R-anchor is most likely to be a positive sample of the ground truth box. As a result, the fit domains of the 6 orientations divide the angle range $[-\frac{\pi}{4}, \frac{3\pi}{4})$ into 6 equal parts. Thus, a ground truth in any orientation can be fitted with an R-anchor of the appropriate fit domain. Fig. 4 shows a comparison of the utility of the regression terms. We can observe that the orientations of the regions are similar in a neighborhood region.

To verify the ability of a network to learn the text region orientation, we visualize the intermediate results in Fig. 5. For an input image, the feature maps of RRPN training after different iterations are visualized. The short white line on the feature

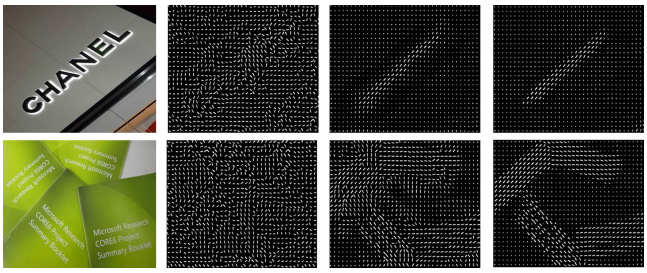


Fig. 5. Visualization of different multitask loss values. (a) Input images; (b) 0 iterations; (c) 15,000 iterations; (d) 150,000 iterations.

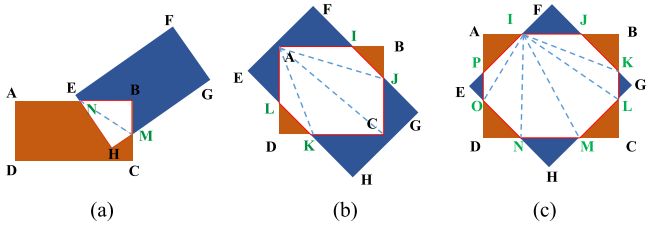


Fig. 6. Examples of skew IoU computation: (a) 4 points, (b) 6 points, (c) 8 points (vertices of rectangle are in black, while intersection points are in green). Considering example (b), first add intersection points I, J, L, and K and inner vertices A and C to $PSet$, sort $PSet$ to obtain convex polygon AIJCKL, and then calculate the intersection area $Area$(AIJCKL) = $Area$(ΔAIJ) + $Area$(ΔAJC) + $Area$(ΔACK) + $Area$(ΔAKL).

Algorithm 1: IoU Computation

1: **Input**: Rectangles $R_1, R_2, ..., R_N$
2: **Output**: IoU between rectangle pairs IoU
3: **for** each pair $\langle R_i, R_j \rangle$ $(i < j)$ **do**
4: Point set $PSet \leftarrow \emptyset$
5: Add intersection points of R_i and R_j to $PSet$
6: Add the vertices of R_i inside R_j to $PSet$
7: Add the vertices of R_j inside R_i to $PSet$
8: Sort $PSet$ into anticlockwise order
9: Compute intersection I of $PSet$ by triangulation
10: $IoU[i,j] \leftarrow \frac{Area(I)}{Area(R_i)+Area(R_j)-Area(I)}$
11: **end for**

map represents the R-anchor with the highest response to the text instance. The orientation of the short line is the orientation of this R-anchor, while the length of the short line indicates the level of confidence. We can observe that the brighter field of the feature map focuses on the text region, while the other region becomes darker after 150,000 iterations. Moreover, the orientations of the regions become closer to the orientation of the text instance as the number of iterations increases.

D. Accurate Proposal Refinement

Skew IoU Computation: The rotation proposals can be generated in any orientations. Thus, the IoU computation for axis-aligned proposals may lead to an inaccurate IoU of skew interactive proposals and further ruin the proposal learning. As shown in Algorithm 1, we design an implementation[1] for the skew IoU computation with consideration of the triangulation [49]; Fig. 6 shows the geometric principles. Given a set of skew rectangles $R_1, ..., R_n$, our goal is to compute the IoU for each pair $\langle R_i, R_j \rangle$. The first step is to generate the intersection point set $PSet$ of R_i and R_j (Lines 4-7 in Algorithm 1). The intersection points of the two rectangles and the vertices of one rectangle inside another rectangle are calculated and inserted into $PSet$. Then, the intersection area of $PSet$ is computed (Lines 8-10 in Algorithm 1). The points in $PSet$ are sorted into anticlockwise order according to their positions in the image, and a convex polygon is generated based on the ordered points. By the triangulation, we can obtain the triangle set (e.g., {ΔAIJ, ΔAJC, ΔACK, ΔAKL} in Fig. 6(b)). The area of the polygon is the sum of the areas of the triangles. Finally, the IoU value is computed.

Skew Non-Maximum Suppression (Skew-NMS): Traditional NMS takes only the IoU factor into consideration (e.g., the IoU threshold is 0.7), but it is insufficient for arbitrary-oriented proposals. For instance, an anchor with a ratio of 1:8 and an angle difference of $\frac{\pi}{12}$ has an IoU of 0.31, which is less than 0.7; however, it may be regarded as a positive sample. Therefore, the Skew-NMS consists of 2 phases: (i) keep the max IoU for proposals with an IoU larger than 0.7; (ii) if all proposals have an IoU in the range $[0.3, 0.7]$, keep the proposal with the minimum angle difference with respect to the ground truth (the angle difference should be less than $\frac{\pi}{12}$).

E. RRoI Pooling Layer

As presented for the Fast-RCNN [48], the RoI pooling layer extracts a fixed-length feature vector from the feature map for each proposal. Each feature vector is fed into fully connected layers that finally branch into the sibling *cls* and *reg* layers, and the outputs are the predicted localization and class of an object in an input image. As the feature map of image needs to be computed only once per image rather than computed for every generated proposal, the object detection framework is accelerated. The RoI pooling layer uses max pooling to convert the feature inside any valid RoI into a small feature map with a fixed spatial extent of $h_r \times w_r$, where h_r and w_r are layer hyperparameters that are independent of any RoI.

For the arbitrary-oriented text detection task, the traditional RoI pooling layer can only handle axis-aligned proposals. Thus, we present the rotation RoI (RRoI) pooling layer to adjust arbitrary-oriented proposals generated by RRPNs. We first set the RRoI layer hyperparameters to H_r and W_r for the RRoIs. The rotated proposal region can be divided into $H_r \times W_r$ subregions of $\frac{h}{H_r} \times \frac{w}{W_r}$ size for a proposal with height h and width w (as shown in Fig. 7(a)). Each subregion have the same orientation as that of the proposal. Fig. 7(b) displays an example with 4 vertices (A, B, C, and D) of the subregion on the feature map. The 4 vertices are calculated using a similarity transformation (shift, scale, and rotate) and grouped to range the border of the subregion. Then, max pooling is performed in every subregion,

[1]Here, we use the GPU to accelerate the computation speed.

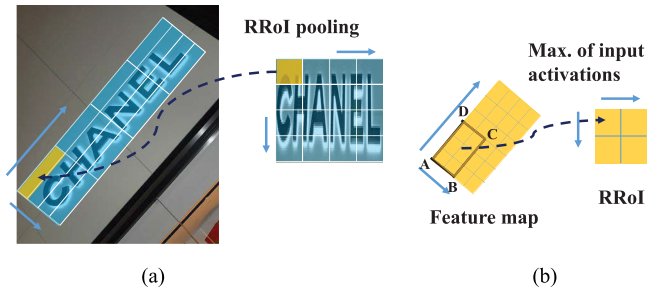


Fig. 7. RRoI pooling layer: (a) divide arbitrary-oriented proposal into subregions; (b) max pooling of a single region from an inclined proposal to a point in the RRoI.

Algorithm 2: RRoI Pooling

Input: Proposal (x, y, h, w, θ), pooled size (H_r, W_r), input feature map $InFeatMap$, spatial scale SS
Output: Output feature map $OutFeatMap$
$Grid_w, Grid_h \leftarrow \frac{w}{W_r}, \frac{h}{H_r}$
for $\langle i, j \rangle \in \{0, ..., H_r - 1\} \times \{0, ..., W_r - 1\}$ **do**
  $L, T \leftarrow x - \frac{w}{2} + jGrid_w, y - \frac{h}{2} + iGrid_h$
  $L_{rotate} \leftarrow (L - x)\cos\theta + (T - y)\sin\theta + x$
  $T_{rotate} \leftarrow (T - y)\cos\theta - (L - x)\sin\theta + y$
  $value \leftarrow 0$
  for $\langle k, l \rangle \in \{0, ..., \lfloor Grid_h \cdot SS - 1 \rfloor\} \times \{0, ..., \lfloor Grid_w \cdot SS - 1 \rfloor\}$ **do**
    $P_x \leftarrow \lfloor L_{rotate} \cdot SS + l\cos\theta + k\sin\theta + \frac{1}{2} \rfloor$
    $P_y \leftarrow \lfloor T_{rotate} \cdot SS - l\sin\theta + k\cos\theta + \frac{1}{2} \rfloor$
    if $InFeatMap[P_y, P_x] > value$ **then**
      $value \leftarrow InFeatMap[P_y, P_x]$
    end if
  end for
  $OutFeatMap[i, j] \leftarrow value$
end for

and max-pooled values are saved in the matrix of each RRoI; the pseudo-code for RRoI pooling is shown in Algorithm 2. Compared with RoI pooling, RRoI pooling can pool any regions, with various angles, aspect ratios, or scales, into a fixed-size feature map. Finally, the proposals are transferred into RRoIs and sent to the classifiers to give the result, i.e., either text or background.

V. Experiments

We evaluate the rotation-based framework on three popular text detection benchmarks: MSRA-TD500 [21], ICDAR2015 [23] and ICDAR2013 [22]. We follow the evaluation protocols of these benchmarks. The MSRA-TD500 dataset contains 300 training images and 200 testing images. Annotations of the images consist of both the position and orientation of each text instance, and the benchmark can be used to evaluate the text detection performance over the multi-oriented text instance. As the dataset of MSRA-TD500 is relatively smaller, its experiments are designed to exploit alternative settings. ICDAR2015 was released for the text localization of the incidental scene text

TABLE I
Effect of Data Augmentation

Data Augmentation	Precision	Recall	F-measure
Without rotation	44.5%	38.9%	41.5%
With rotation	**68.4%**	**58.9%**	**63.3%**

TABLE II
Runtime of Proposed Approach and of the Faster-RCNN

	Baseline	With border padding
Faster-RCNN	0.094 s	0.112 s
The work	0.214 s	0.225 s

These runtimes were achieved using a single Nvidia Titan X GPU.

Fig. 8. Comparison of rotation and horizontal region proposals. Left: original images; middle: text detection based on horizontal region proposal; right: text detection based on rotation region proposal.

challenge (Task 4.1) of the ICDAR 2015 Robust Reading Competition; it has 1,500 images in total. Different from previous ICDAR robust reading competitions, the text instance annotations have four vertices, which form an irregular quadrilateral bounding box with orientation information. We roughly generate an inclined rectangle to fit the quadrangle and its orientation. The ICDAR2013 dataset is from the ICDAR 2013 Robust Reading Competition. There are 229 natural images for training and 233 natural images for testing. All the text instances in this dataset are horizontally aligned, and we conduct experiments on this horizontal benchmark to determine the adaptability of our approach to specific orientations.

Implementation Details: Our network is initialized by pretraining a model for ImageNet classification [47]. The weights of the network are updated by using a learning rate of 10^{-3}

(a) (b) (c)

Fig. 9. Examples of failed detection on MSRA-TD500: (a) blur and uneven lighting situations; (b) extremely small text instances; (c) extremely long text line. The red boxes indicate instances of negative detection, i.e., either IoU < 0.5 or failed to detect; the yellow boxes indicate instances of positive detection with respect to the ground truth.

for the first 200,000 iterations and 10^{-4} for the next 100,000 iterations, with a weight decay of 5×10^{-4} and a momentum of 0.9. We use the rotation of an image with a random angle for the data augmentation, as their efficiency and measurements are improved when the augmentation is used (see Table I).

Due to our different R-anchor strategy, the total number of proposals for each image is nearly 6 times that of previous approaches such as the Faster-RCNN. To ensure efficient detection, we filter the R-anchors to remove those passing through the border of an image. Therefore, the speed of our system is similar to that of previous works in both the training and testing stages; a comparison with the state-of-the-art approaches on MSRA-TD500 is presented in Table VI-Left. Table II shows the runtime speed of our proposed framework and that of the original Faster-RCNN under the baseline settings and with border padding. We can observe that our approach takes two times as much as the Faster-RCNN approach.

A. Ablation Study

We first perform an ablation study on the smaller dataset, i.e., MSRA-TD500. The baseline system is trained using 300 images from the MSRA-TD500 training set; the input image is resized, with long side being 1,000 pixels. The evaluation result is a precision of 57.4%, recall of 54.5%, and F-measure of 55.9%, which reflects a much better performance compared to that of the original Faster-RCNN, the P, R and F of which were 38.7%, 30.4% and 34.0%, respectively. We make a comparison between rotation and horizontal region proposals, with some detection results illustrated in Fig. 8. The rotation-based approach is able to achieve accurate detection with less background area, which indicates the effectiveness of incorporating the rotation strategy.

Further analysis of the baseline results give us the following insights: (i) the difficult situations (e.g., blur and uneven lighting) in the image can hardly be detected; (ii) some text instances of extremely small size cannot be properly detected, resulting in a large recall loss regarding the performance; (iii) the extremely long text line, i.e., a height-width ratio of the bounding box larger than 1:10, cannot be correctly detected and is often split into several shorter proposals; hence, all the proposals become instances of false detection according to the evaluation of MSRA-TD500 (some failed detection instances are shown in Fig. 9). A few alternative strategies and settings from the baseline approach are tested; a summary is given in Table III.

TABLE III
EVALUATION ON MSRA-TD500 WITH DIFFERENT STRATEGIES AND SETTINGS

a.	b.	c.	d.	P	R	F	ΔF
Faster-RCNN [20]				38.7%	30.4%	34.0%	–
Baseline				57.4%	54.5%	55.9%	–
√				65.6%	58.4%	61.8%	5.9%
	√			63.3%	58.5%	60.8%	4.9%
		√		63.1%	55.4%	59.0%	3.1%
√	√	√		68.4%	58.9%	63.3%	7.4%
√	√	√	√	**71.8%**	**67.0%**	**69.3%**	13.4%

Experiments on Faster-RCNN are based on the original source code. P, R and F denote the precision, recall, and F-measure, respectively. ΔF is the improvement of the F-measure over the baseline. The strategies include the following: a. context of the text region; b. training dataset enlargement; c. border padding; and d. scale jittering.

TABLE IV
EXPLOITATION OF THE TEXT REGION CONTEXT BY ENLARGING THE TEXT BOUNDING BOX BY DIFFERENT FACTORS OF THE ORIGINAL SIZE

Factor	Precision	Recall	F-measure
1.0	57.4%	54.5%	55.9%
1.2	59.3%	57.0%	58.1%
1.4	**65.6%**	**58.4%**	**61.8%**
1.6	63.8%	56.8%	60.1%

Context of the Text Region: Incorporating the contextual information has been proven to be useful for the general object detection task (e.g., [50]), and we wonder whether it can promote a text detection system. We retain the center of the rotated bounding box and its orientation and enlarge both the width and height by a factor of $1.X$ in the data preprocessing step. During the testing phase, we divide the enlargement for every proposal. As shown in Table IV, all the experiments exhibit an obvious increase in the F-measure. The reason may be that as the bounding box becomes larger, more context information of the text instance is obtained, and the information regarding the orientation can be better captured. Thus, the orientation of the proposals can be more precisely predicted.

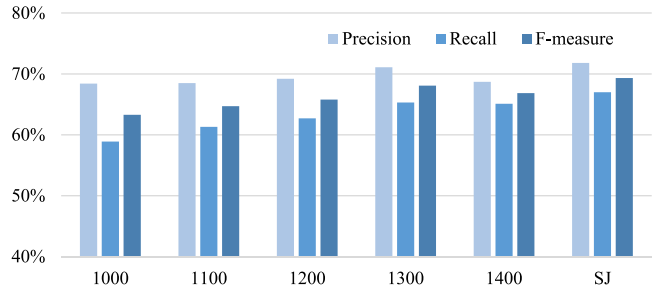


Fig. 10. Evaluation on MSRA-TD500 for different input scales. $1X00$ (X = 0, 1, 2, 3, 4) denotes those inputs with a long side of $1X00$ pixels, and SJ is the result with scale jittering. The experiments are conducted using strategies of context of the text region, training dataset enlargement, and border padding.

Training Dataset Enlargement: We adopt HUST-TR400 (contains 400 images, with text instances annotated using the same parameters as for MSRA-TD500) [51] as an additional dataset and form a training set of 700 images from both datasets. There is a significant improvement in all the measurements, and the F-measure is 60.8%, showing that the network is better trained and more robust when addressing noisy inputs.

Border Padding: Using our filtering strategy, most of the boundary breaking R-anchors are eliminated. However, as the bounding box is rotated by certain angles, it may still exceed the image border, especially when we enlarge the text region for the contextual information. Thus, we set a border padding of 0.25 times each side to reserve more positive proposals. The experiment shows that adding border padding to an image improves the detection results. The border padding increases the amount of computation for our approach by approximately 5% (Table II). In addition, combining border padding with enlargement of the text region and the training dataset yields a further improvement in the F-measure of 63.3%.

Scale Jittering: There are still a number of small text regions in both training datasets, and we would like to improve the robustness of our system. One approach is to rescale the input images to a fixed larger size, and another is to perform scale jittering, i.e., rescaling with a long side of a random size before sending the image into the network. Fig. 10 shows that the inputs with a long side of 1300 pixels outperform those with other fixed settings (precision: 71.1%, recall: 65.3%, and F-measure: 68.1%). When we apply scale jittering with a long side of a random size less than 1300 pixels, a better result is achieved compared to that of the experiment without jittering.

B. Performance on Benchmarks

MSRA-TD500: We use the best settings from the ablation study. The annotation of MSRA-TD500 prefers to label the region of a whole text line. Thus, the length of a text line does not have a fixed range, sometimes being very long. However, the ratios for the R-anchor are fixed and may not be large enough to cover all the lengths, which leads to several short bounding box results for a single text region. To address this extremely long text line issue, a post-processing step is incorporated by linking multiple short detection segments into a finer proposal,

Algorithm 3: Text-Linking Processing

Input: Proposal $P_1, ..., P_N$ $(P_k = x_k, y_k, h_k, w_k, \theta_k)$
Output: Merged Proposal Set $PSet$
Angle Threshold $T \leftarrow 10$
if $N == 1$ **then**
  $PSet \leftarrow \{P_1\}$
end if
for $k \in \{1, ..., N\}$ **do**
  Valid$[k] \leftarrow 1$
end for
for each pair $\langle P_i, P_j \rangle$ $(i < j)$ **do**
  if Valid$[i] == 0$ or Valid$[j] == 0$ **then**
    Continue
  end if
  MeanWidth $Width \leftarrow \frac{w_i + w_j}{2}$
  CenterDistance $Dis \leftarrow \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$
  CenterGrad $Grad \leftarrow |arctan(\frac{y_j - y_i}{x_j - x_i})|$
  if $Dis < Width$ and $|Grad - \theta_i| < T$ **then**
    $P_i \leftarrow \frac{x_i + x_j}{2}, \frac{y_i + y_j}{2}, \frac{h_i + h_j}{2}, w_i + w_j, \frac{\theta_i + \theta_j}{2}$
    Valid$[j] \leftarrow 0$
  end if
end for
$PSet \leftarrow \{P_k | \text{Valid}[k] == 1\}$

as detailed in Algorithm 3. With this post-processing, the performance is further boosted, with the F-measure being 74.2% and the time cost being only 0.3 s. We also conduct an experiment that incorporates the post-processing as well as the strategies presented in Section V-A on the Faster-RCNN [20]; the results are a precision of 42.7%, recall of 37.6%, and F-measure of 40.0%. The comparison verifies that using a rotation-based framework is necessary to achieve a more robust text detector. Note that the post-processing is applied for the text line detection benchmark, i.e., MSRA-TD500, only and that we do not apply the algorithm to the ICDAR benchmarks. The results (RRPN) on MSRA-TD500 are shown in the left-most column of Table VI.

ICDAR 2015: We train a baseline experiment on the ICDAR2015 benchmark using the same strategy used for MSRA-TD500. The evaluation result is a precision of 45.42%, recall of 72.56%, and F-measure of 55.87%. There are some differences between these two datasets. MSRA-TD500 tends to provide a text line ground truth, while ICDAR provides word-level annotations. Thus, the precision of our approach is lower than that of other methods that achieve the same F-measure. This issue may originate from three aspects. First, some of the incidental text regions are still too small for our detector to find. Second, there exist some small unreadable text instances (labeled '###') in the ICDAR2015 training set, which may lead to the false detection of text-like instances. Finally, our training set is insufficient (containing only 1,000 images) compared with those of previous approaches, such as [17], [18], [42]. Some of the detection results obtained based on the ICDAR2015 training set are shown in Fig. 12.

(a) (b) (c)

Fig. 11. Text detection results for different benchmarks. (a) MSRA-TD500; (b) ICDAR2015; (c) ICDAR2013.

Fig. 12. Text detection on ICDAR2015, with the model trained on the ICDAR2015 training set (including all unreadable text instances). The yellow areas denote instances of positive detection, with IoU > 0.5, while red areas represent text regions that were not detected.

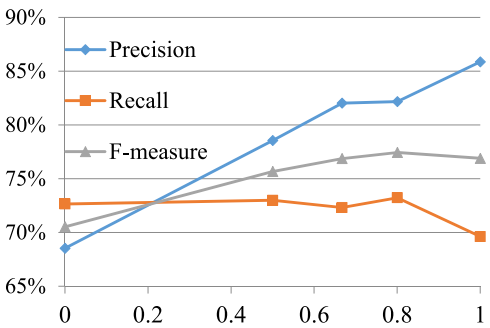


Fig. 13. Effect of unreadable text instance proportion using cross-validation on the training set. Horizontal axis represents the proportion of unreadable text instances removed; vertical axis represents the F-measure as a percentage.

To address the small text region issue, we create a larger scale by jittering the image patch with a long side of a random size less than 1,700 pixels before sending it into the network. We also check the impact of small unreadable text instances by randomly removing these instances from the training set. Fig. 13 displays the curves of the measurements. The recall rate remains the same, i.e., approximately 72%–73%, unless we remove all the unreadable instances, while the precision significantly increases with the proportion. Therefore, we randomly remove 80% of the unreadable text instances in the training set and keep the whole testing set. To further improve our detection system, we incorporate a few text datasets for training, i.e., ICDAR2013 [22], ICDAR2003 [52] and SVT [7]. As listed in Table V, the training images for different approaches are of the same order of magnitude, and ours achieves better performance.

TABLE V
TRAINING SET AND RESULTS FOR ICDAR2015

Approach	RRPN		[18]	[42]	[17]
# of images	2077	1229	1529	3000	1529
Training set	I13 I15 I03 SVT	I13 I15	I13 I15 M500	I13 CA	I13 I15 M500
Precision	**82.17%**	79.41%	72.26%	74.22%	70.81%
Recall	**73.23%**	70.00%	58.69%	51.56%	43.09%
F-measure	**77.44%**	74.41%	64.77%	60.85%	53.58%

I*XX* indicate ICDAR20*XX* training set, M500 indicates MSRA-TD500, SVT indicates SVT dataset [7], and CA indicates data collected by the authors of [42].

ICDAR 2013: To examine the adaptability of our approach, we also conduct experiments on the horizontal-based ICDAR2013 benchmark. We reuse the model trained for ICDAR2015, and the 5-tuple rotation proposals are fit into horizontal-aligned rectangles. The result is a precision of 90.22%, recall of 71.89%, and F-measure of 80.02% under the ICDAR 2013 evaluation protocol. As shown in Table VI, there is a 7% improvement compared with the Faster-RCNN, which confirms the robustness of our detection framework with the rotation factor.

C. More Results

The experimental results of our method compared with those of the state-of-the-art approaches are given in Table VI. As the RRPN models are trained separately for MSRA-TD500 and ICDAR, we also train a unified model (RRPN*) trained on all of the training sets to consider the generalization issue.

TABLE VI
COMPARISON WITH STATE-OF-THE-ART APPROACHES ON THREE BENCHMARKS

MSRA-TD500					ICDAR2015				ICDAR2013			
Approach	P	R	F	Time	Approach	P	R	F	Approach	P	R	F
Yin *et al.* [53]	71	61	65	0.8 s	CTPN [42]	74	52	61	Faster-RCNN [20]	75	71	73
Kang *et al.* [54]	71	62	66	–	Yao *et al.* [18]	72	59	65	Gupta *et al.* [55]	92	76	83
Yin *et al.* [56]	81	63	71	1.4 s	SCUT_DMPNet [57]	68	73	71	Yao *et al.* [18]	89	80	84
Zhang *et al.* [17]	**83**	67	74	2.1 s	UCSC_TextSpotter [58]	65	**79**	71	DeepText [43]	85	81	85
Yao *et al.* [18]	77	**75**	**76**	0.6 s	hust_orientedText [59]	77	75	76	CTPN [42]	93	83	88
RRPN	82	68	74	0.3 s	RRPN	82	73	77	RRPN	90	72	80
RRPN*	82	69	75	0.3 s	RRPN*	**84**	77	**80**	RRPN*	**95**	**88**	**91**

Faster-RCNN results based on ICDAR2013 are reported in [43]. Bold text denotes the top result, while underlined text corresponds to the second runner-up.

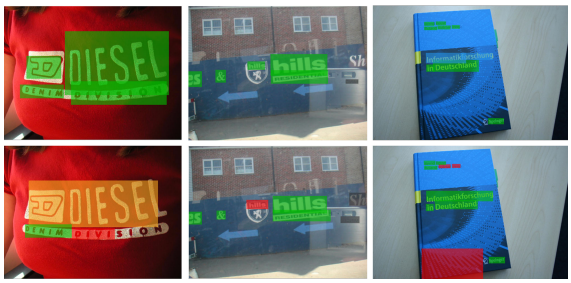


Fig. 14. Detection results of the proposed approach and DeepText [43], downloaded from the ICDAR evaluation website.[2] The green and red boxes indicate instances of positive and false detection, respectively, the orange box refers to "one box covering multiple instances", and the blue box indicates multiple occurrences of detection for one instance.

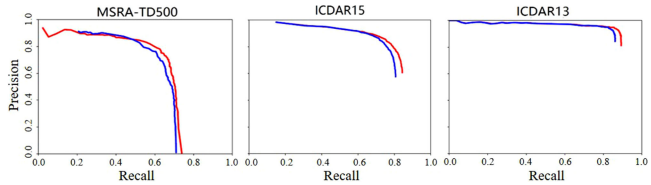


Fig. 15. Precision-recall curve of the benchmarks. The red and blue curves represent the results of RRPN and RRPN*, respectively.

The precision-recall curves of RRPN and RRPN* on the three datasets are illustrated in Fig. 15. For the MSRA-TD500 dataset, the performance of our RRPN reaches the same magnitude of that of the state-of-the-art approaches, such as [18] and [17]. When our system achieves text detection, it is more efficient than others, requiring a processing time of only 0.3 s per testing image. For the ICDAR benchmarks, the substantial performance gains over the published works confirm the effectiveness of using a rotation region proposal and rotation RoI for the text detection task. The recently developed DeepText [43] is also a detection-based approach, but it is based on the Inception-RPN structure. Both our approach and DeepText are evaluated on the ICDAR2013 benchmark. The evaluation results in Table VI and detection examples in Fig. 14 demonstrate that our approach performs better in terms of different evaluation measurements. We believe that our rotation-based framework is also complementary to the Inception-RPN structure, as they both focus on different levels of information. Some detection results obtained on the benchmarks are illustrated in Fig. 11, and we have released the code and trained models for future research.[3]

VI. CONCLUSION

In this paper, we introduced a rotation-based detection framework for arbitrary-oriented text detection. Inclined rectangle proposals were generated with the text region orientation angle information from higher convolutional layers of network, resulting in the detection of text with multiple orientations. A novel RRoI pooling layer was also designed and adapted to the rotated RoIs. Experimental comparisons with the state-of-the-art approaches on MSRA-TD500, ICDAR2013 and ICDAR2015 showed the effectiveness and efficiency of our proposed RRPN and RRoI for the text detection task.

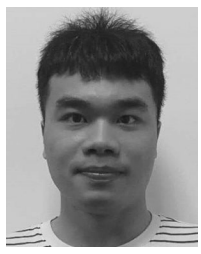
REFERENCES

[1] S. Karaoglu, R. Tao, T. Gevers, and A. W. M. Smeulders, "Words matter: Scene text for image classification and retrieval," *IEEE Trans. Multimedia*, vol. 19, no. 5, pp. 1063–1076, May 2017.
[2] X. Bai, M. Yang, P. Lyu, and Y. Xu, "Integrating scene text and visual appearance for fine-grained image classification with convolutional neural networks," *arXiv:1704.04613*, 2017.
[3] X.-C. Yin, Z.-Y. Zuo, S. Tian, and C.-L. Liu, "Text detection, tracking and recognition in video: A comprehensive survey," *IEEE Trans. Image Process.*, vol. 25, no. 6, pp. 2752–2773, Jun. 2016.
[4] X. Liu and W. Wang, "Robustly extracting captions in videos based on stroke-like edges and spatio-temporal analysis," *IEEE Trans. Multimedia*, vol. 14, no. 2, pp. 482–489, Apr. 2012.
[5] K. L. Bouman, G. Abdollahian, M. Boutin, and E. J. Delp, "A low complexity sign detection and text localization method for mobile applications," *IEEE Trans. Multimedia*, vol. 13, no. 5, pp. 922–934, Oct. 2011.
[6] X. Chen and A. L. Yuille, "Detecting and reading text in natural scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2004, pp. 366–373.
[7] K. Wang and S. Belongie, "Word spotting in the wild," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 591–604.
[8] L. Neumann and J. Matas, "A method for text localization and recognition in real-world images," in *Proc. 10th Asian Conf. Comp. Vis.*, LNCS, vol. 6494, 2010, pp. 770–783.
[9] A. Bissacco, M. Cummins, Y. Netzer, and H. Neven, "Photoocr: Reading text in uncontrolled conditions," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 785–792.

[2]RRPN: http://rrc.cvc.uab.es/?ch=2&com=evaluation&view=method_samples&task=1&m=15904&gtv=1 ; DeepText: http://rrc.cvc.uab.es/?ch=2&com=evaluation&view=method_samples&task=1&m=8665&gtv=1

[3]https://github.com/mjq11302010044/RRPN

[10] M. Jaderberg, A. Vedaldi, and A. Zisserman, "Deep features for text spotting," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 512–528.
[11] W. Huang, Y. Qiao, and X. Tang, "Robust scene text detection with convolution neural network induced MSER trees," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 497–511.
[12] L. Wu, P. Shivakumara, T. Lu, and C. L. Tan, "A new technique for multi-oriented scene text line detection and tracking in video," *IEEE Trans. Multimedia*, vol. 17, no. 8, pp. 1137–1152, Aug. 2015.
[13] S. Tian *et al.*, "Text flow: A unified text detection system in natural scene images," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 4651–4659.
[14] D. Bazazian *et al.*, "Improving text proposals for scene images with fully convolutional networks," *arXiv:1702.05089*, 2017.
[15] X. Ren *et al.*, "A convolutional neural network-based chinese text detection algorithm via text structure modeling," *IEEE Trans. Multimedia*, vol. 19, no. 3, pp. 506–518, Mar. 2017.
[16] 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.
[17] Z. Zhang *et al.*, "Multi-oriented text detection with fully convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4159–4167.
[18] C. Yao *et al.*, "Scene text detection via holistic, multi-channel prediction," *arXiv:1606.09002*, 2016.
[19] T. He, W. Huang, Y. Qiao, and J. Yao, "Accurate text localization in natural image with cascaded convolutional text network," *arXiv:1603.09423*, 2016.
[20] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks." *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, 2016.
[21] 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.*, 2012, pp. 1083–1090.
[22] D. Karatzass, F. Shafait, and S. Uchida, "Icdar 2013 robust reading competition," in *Proc. Int. Conf. Document Anal. Recognit.*, 2013, pp. 1484–1493.
[23] D. Karatzas, L. Gomez-Bigorda, and A. Nicolaou, "Icdar 2015 competition on robust reading," in *Proc. Int. Conf. Document Anal. Recognit.*, 2015, pp. 1156–1160.
[24] D. Chen and J. Luettin, "A survey of text detection and recognition in images and videos," Tech. Rep. IDIAP-RR. 00-38, 2000.
[25] K. Jung, K. I. Kim, and A. K. Jain, "Text information extraction in images and video: A survey," *Pattern Recognit.*, vol. 37, no. 5, pp. 977–997, 2004.
[26] S. Uchida, "Text localization and recognition in images and video," in *Handbook of Document Image Processing and Recognition*, New York, USA: Springer, 2014, pp. 843–883.
[27] 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.
[28] K. I. Kim, K. Jung, and J. H. Kim, "Texture-based approach for text detection in images using support vector machines and continuously adaptive mean shift algorithm," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 12, pp. 1631–1639, Dec. 2003.
[29] X. Chen and A. L. Yuille, "Detecting and reading text in natural scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2004, vol. 2, pp. 366–373.
[30] L. Neumann and J. Matas, "Scene text localization and recognition with oriented stroke detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 97–104.
[31] T. Wang, D. J. Wu, A. Coates, and A. Y. Ng, "End-to-end text recognition with convolutional neural networks," in *Proc. Int. Conf. Pattern Recognit.*, 2012, pp. 3304–3308.
[32] B. Epshtein, E. Ofek, and Y. Wexler, "Detecting text in natural scenes with stroke width transform," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 2963–2970.
[33] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image Vis. Comput.*, vol. 22, no. 10, pp. 761–767, 2004.
[34] A. Shahab, F. Shafait, and A. Dengel, "Icdar 2011 robust reading competition challenge 2: Reading text in scene images," in *Proc. Int. Conf. Document Anal. Recognit.*, 2011, pp. 1491–1496.
[35] S. Zhang, M. Lin, T. Chen, L. Jin, and L. Lin, "Character proposal network for robust text extraction," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2016, pp. 2633–2637.
[36] A. Risnumawan, P. Shivakumara, C. S. Chan, and C. L. Tan, "A robust arbitrary text detection system for natural scene images," *Expert Syst. Appl.*, vol. 41, no. 18, pp. 8027–8048, 2014.
[37] 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.*, 2016, pp. 3566–3573.
[38] 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.
[39] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 391–405.
[40] J. R. R. Uijlings, K. E. A. V. D. Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," *Int. J. Comput. Vis.*, vol. 104, no. 2, pp. 154–171, 2013.
[41] 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, 2014.
[42] 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.
[43] Z. Zhong, L. Jin, S. Zhang, and Z. Feng, "Deeptext: A unified framework for text proposal generation and text detection in natural images," *arXiv:1605.07314*, 2016.
[44] H. Jiang and E. G. Learned-Miller, "Face detection with the faster R-CNN," *arXiv:1606.03473*, 2016.
[45] L. Wang *et al.*, "Evolving boxes for fast vehicle detection," in *Proc. IEEE Int. Conf. Multimedia Expo.*, 2017, pp. 1135–1140.
[46] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial transformer networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 2017–2025.
[47] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent.*, arXiv preprint arXiv: 1409.1556, 2014. [Online]. Available: https://arxiv.org/abs/1409.1556
[48] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1440–1448.
[49] D. A. Plaisted and J. Hong, "A heuristic triangulation algorithm," *J. Algorithms*, vol. 8, no. 3, pp. 405–437, 1987.
[50] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016.
[51] C. Yao, X. Bai, and W. Liu, "A unified framework for multioriented text detection and recognition." *IEEE Trans. Image Process.*, vol. 23, no. 11, pp. 4737–4749, Nov. 2014.
[52] S. M. Lucas *et al.*, "ICDAR 2003 robust reading competitions," in *Proc. 17th Int. Conf. Document Anal. Recognit.*, 2003, pp. 682–687.
[53] 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.
[54] L. Kang, Y. Li, and D. Doermann, "Orientation robust text line detection in natural images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 4034–4041.
[55] A. Gupta, A. Vedaldi, and A. Zisserman, "Synthetic data for text localisation in natural images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2315–2324.
[56] 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.
[57] Y. Liu and L. Jin, "Deep matching prior network: Toward tighter multi-oriented text detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 3454–3461.
[58] S. Qin and R. Manduchi, "Cascaded segmentation-detection networks for word-level text spotting," in *Proc. Int. Conf. Document Anal. Recognit.*, 2017, pp. 1275–1282.
[59] B. Shi, X. Bai, and S. Belongie, "Detecting oriented text in natural images by linking segments," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 3482–3490.



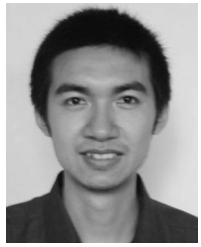
Jianqi Ma received the B.S. degree in software engineering from the School of Software Engineering, Fudan University, Shanghai, China, in 2015. He is currently working toward the M.S. degree with the School of Computer Science, Fudan University, Shanghai, China. His research interests include computer vision, signal processing, and machine learning.

Weiyuan Shao received the B.S. degree from Shanghai Normal University, Shanghai, China, in 2012, and the M.S. degree in computer science from Fudan University, Shanghai, China, in 2016. He joined Shanghai Advanced Research Institute, Chinese Academy of Sciences, in July 2016. His research interests include computer vision, especially in scene text detection and face recognition.

Hao Ye received the Ph.D. degree in computer science from Fudan University, Shanghai, China, in 2016. He is currently an Associate Professor with the Shanghai Advanced Research Institute, Chinese Academy of Sciences, Shanghai, China. His research interests include computer vision, multimedia information processing, and deep learning.

Li Wang received the B.S. degree in software engineering from Hubei University, Wuhan, China, in 2010. She is currently working toward the M.S. degree with the School of Computer Science, Fudan University, Shanghai, China. Her research interests include computer vision and machine learning.

Hong Wang received the B.S. degree from the University of Science and Technology of China, Hefei, China, in 2005, and the Ph.D. degree from the University of Chinese Academy of Sciences, Beijing, China, in 2011. He is currently an Associate Professor with the Shanghai Advanced Research Institute, Chinese Academy of Science, Shanghai, China. His research interests include computer vision and signal processing.



Yingbin Zheng received the B.S. and Ph.D. degrees in computer science from Fudan University, Shanghai, China, in 2008 and 2013, respectively. He was a Research Scientist with the SAP Labs, Shanghai Shi, China from 2013 to 2015. Since 2015, he has been with the Shanghai Advanced Research Institute, Chinese Academy of Sciences, Shanghai, China, where he is currently an Associate Professor. His research interests include computer vision, especially in scene understanding and video data analysis.

Xiangyang Xue received the B.S., M.S., and Ph.D. degrees in communication engineering from Xidian University, Xi'an, China, in 1989, 1992, and 1995, respectively. He is currently a Professor of computer science with the Fudan University, Shanghai, China. His research interests include computer vision and multimedia information processing.