Arbitrary-Oriented Scene Text Detection via Rotation Proposals

Jianqi Ma[1]*, Weiyuan Shao[2]*, Hao Ye[2], Li Wang[1], Hong Wang[2], Yingbin Zheng[2]†, Xiangyang Xue[1]
[1]School of Computer Science, Fudan University, Shanghai, China
[2]Shanghai Advanced Research Institute, Chinese Academy of Sciences, Shanghai, China

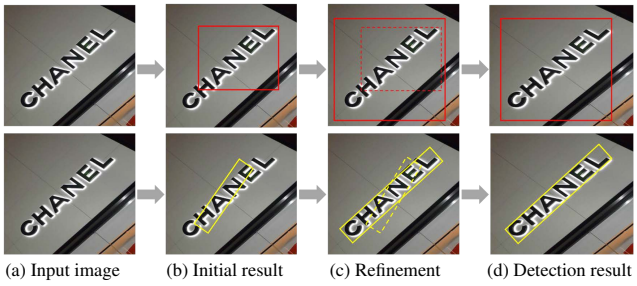
Abstract

This paper introduces a novel rotation-based framework for arbitrary-oriented text detection in natural scene images. We present the Rotation Region Proposal Networks (RRPN), which is 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 orientation. The Rotation Region-of-Interest (RRoI) pooling layer is proposed to project arbitrary-oriented proposals to the feature map for a text region classifier. The whole framework is built upon region proposal based architecture, which ensures the computational efficiency of the arbitrary-oriented text detection comparing 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.

1. Introduction

Text detection is an important prerequisite for many computer vision and document analysis problems, which aims to identify text regions of the given images. Although there are a few commercial OCR systems for documental texts or internet contents, detect text in natural scene image is challenge due to hard situations such as uneven lighting, burring, perspective distortion, orientation, and *etc*.

In the past years there have been many attentions in the text detection task [3, 22, 7, 35, 23, 1, 15, 12, 31]. 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 great number of the text regions are not horizontal, and even applying the non-horizontal aligned text lines as the axis-aligned proposals may not be accurate, so the horizontal-specific methods cannot be widely applied in practice.



(a) Input image (b) Initial result (c) Refinement (d) Detection result

Figure 1: Overview for text detection. First row: text detection based on horizontal bounding box. Second row: detection using rotation region proposal.

Recently, a few works have been proposed to address the arbitrary-oriented text detection [44, 39, 10]. In general, there are mainly two steps included in the task of these methods, *i.e*., segmentation networks such as Fully Convolutional Network (FCN) to generate the text prediction maps and geometric approaches for inclined proposals. However, prerequisite segmentation is usually time-consuming. Besides, some systems need several post-processing steps to generate final text region proposal with orientation, which are not as efficient as those based on detection network directly.

In this paper, we come up with 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. Comparison between the previous horizontal-based approach and ours are illustrated in Figure 1. We present the Rotation Region Proposal Networks (RRPN), which is designed to generate inclined proposals with text orientation angle information. The angle information are 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 the feature map. Finally, a two-layer network is deployed to classify the regions as text or background. The main contributions of this paper include:

*These authors contributed equally to this work.
†Corresponding author. E-mail: zhengyb@sari.ac.cn

- Different from previous segmentation-based framework, ours has ability to predict the orientation of a text line using region proposal based approach, so that the proposals can better fit the text region and the ranged text region can be easily rectified and more convenient for text reading.
- The framework incorporates new components into region proposal based architecture [8], which ensures the computational efficiency for text detection comparing with segmentation-based text detection systems.
- We also proposed novel strategies for select region proposals with arbitrary orientation to improve the performance on arbitrary-oriented text detection.
- We perform the rotation-based framework on three real-world text detection datasets, *i.e*., MSRA-TD500 [38], ICDAR2013 [19] and ICDAR2015 [18], and find that it is accurate and significantly more efficient than previous approaches.

2. Related Work

Reading text in the wild has been studied during the last few decades and the comprehensive surveys can be found at [2, 16, 33, 40]. Methods based on the sliding window, connected component and bottom-up strategy are designed to handle the horizontal-based text detection. Sliding window-based methods [20, 4, 35, 24, 15] tend to use a sliding window of a fixed size to slide the text area and find the region most likely to be the presence of text. In order to take more presence styles of text into account, [15, 36] applies multi-scale and multi-ratio to the sliding window methods. However, the process of sliding window leads to a large computational cost and leads to inefficiency. Representative approaches of connected component-based like Stroke Width Transform (SWT) [7] and Maximally Stable Extremal Regions (MSER) [22] were with leading performance in ICDAR 2011 [29] and ICDAR 2013 [19] robust text detection competitions. They mainly focus on the edge and pixel point of the image by detecting the character by edge detection or extreme region extraction and then combining the sub MSER components into a word or text-line region. The efforts of these methods are limited in some hard situations such as multiple connected characters, segmented stroke characters and non-uniform illumination [43].

Scene text in the wild is usually aligned from any orientation in real-world applications and approaches for arbitrary orientation are needed. For example, [28] use Mutual Magnitude Symmetry and Gradient Vector Symmetry to identify text pixel candidates regardless of any orientation including curves from natural scene image, and [5] design the Canny Text Detector by taking the similarity between image edge and text to detect text edge pixels and text localization. Recently, convolution network based approaches are proposed to perform text detection, *e.g*., TextCNN [11] by first using 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. And the orientation factor is adopted in segmentation models by Yao et al. [39]. Their model tries to predict more accurate orientation with an explicit manner of text segmentation, and performs outstanding results on ICDAR2013 [19], ICDAR2015 [18] and MSRA-TD500 [38] benchmarks.

A technique similar to text detection is generic object detection. The detection process can be faster if the number of proposals is largely reduced. There is a wide variety of region proposal methods such as Edge Boxes [46], Selective Search [34], and Region Proposal Network (RPN) [27]. For example, Jaderberg *et al*. [13] extend the region proposal method and apply the Edge Boxes method [46] to perform text detection. Their text spotting system achieves outstanding result on several text detection benchmarks. The Connectionist Text Proposal Network (CTPN) [32] is also a detection based framework for scene text detection by employing the image feature from CNN network in LSTM to predict the text region and generate robust proposals.

This work is inspired by the Fast-RCNN [8] and RPN detection pipeline. The RPN can be combined with Fast-RCNN framework to further accelerate the proposal process and achieve the state-of-the-art detection performance on competitions such as ImageNet [6]. The idea is also inspired by Spatial Transformer Networks (STN) [14], which suggests a neural network model can learn to transform an image with an angle degree; here we try to add angle information into the model for multi-oriented text detection. Perhaps the work most related to ours is [45], where the authors proposed Inception-RPN and made further text detection-specific optimization to adapt the text detection. Comparing with previous works, 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 a very good result.

3. Approach

We now elaborate the construction of the rotation-based framework. The architecture is illustrated in Figure 2; details of each phase will be described in the following parts.

3.1. Horizontal Region Proposal

As mentioned in previous section, RPN [27] is able to further accelerate the process of proposal generation. Part of VGG-16 [30] is employed as sharable layers, and the horizontal region proposals are generated by sliding over the

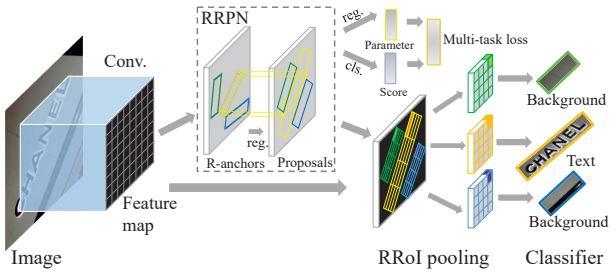


Figure 2: Rotation based text detection pipeline.

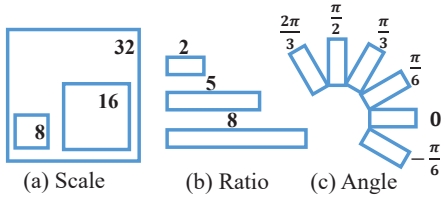


Figure 3: Anchor strategy in our framework.

feature map of the last convolutional layer. The feature extracted from each sliding window are fed into two sibling layers (a box-regression (reg) layer and a box-classification (cls) layer), where there are $4k$ (4 coordinates per proposal) outputs from reg layer representing coordinates and $2k$ (2 scores per proposal) scores from cls layer for k anchors of each sliding position.

To fit the objects in different sizes, RPN uses two parameters to control the size and shape of anchors, *i.e.*, scale and aspect ratio. Scale parameter decides the size of the anchor, and aspect ratio controls the ratio of width and height for anchor box. In [27], the authors set the scale as 8, 16 and 32 and the ratio as 1:1, 1:2 and 2:1 for generic object detection task. This anchor selection strategy can cover almost shapes of all natural objects and keep the total proposals at a small number. However, in the text detection task, especially with the 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 the network more robust for text detection and keep its efficiency, we think that it is necessary to build a detection framework, which encodes the rotation information with the region proposals.

3.2. Proposed Architecture

As shown in Figure 2, we employ the convolutional layers of VGG16 [30] in the front of the framework, which are shared by two sibling branches, *i.e.*, the RRPN and a clone of feature map of last convolutional layer. 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 branched from the RRPN are the classification layer (cls) and regression layer (reg) of RRPN, respectively. Outputs from these two layers are the scores from cls and proposal information from reg and their losses are computed and summed to form multi-task loss. Then RRoI pooling layer acts as a max pooling layer by projecting arbitrary-oriented text proposals from 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 text or background.

3.3. Data Preprocessing

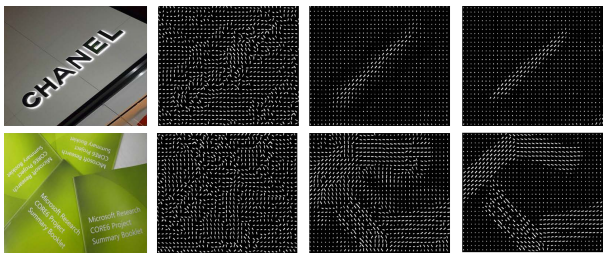
In the training stage, the ground truth of a text region is represented as a rotated bounding box with 5 tuples (x, y, h, w, θ). The coordinate (x, y) represents geometric center of the bounding box. The height h is set as short side of bounding box and the width w as the long side. The orientation θ is the orientation of long side. We fix the range to $[-\frac{\pi}{4}, \frac{3\pi}{4})$ by shifting to the opposite direction if the angle is out of this range.

3.4. Rotation Anchors

Traditional anchors using scale and aspect ratio parameters are not enough for in-the-wild text detection, therefore we design the rotation anchors (R-anchors) with several adjustments. First, orientation parameter is added to control the orientation of a proposal. Six different orientations, *i.e.*, $-\frac{\pi}{6}$, 0, $\frac{\pi}{6}$, $\frac{\pi}{3}$, $\frac{\pi}{2}$ and $\frac{2\pi}{3}$, are used, which are trade-off between orientation coverage and computational efficiency. Second, as text regions are usually with special shapes, the aspect ratio is changed to 1:2, 1:5 and 1:8 to cover a wide range of text line. Besides, the scale of 8, 16 and 32 are kept. Summary of the anchor strategy is in Figure 3. Following our data preprocess step, a proposal is generated from the R-anchors with 5 variables (x, y, h, w, θ). For each point on feature map, 54 R-anchors ($6\times3\times3$) are generated, and respectively 270 outputs (5×54) for reg layer and 108 scores output (2×54) for cls layer on each sliding position. Then, we slide the feature map with RRPN and generates $H \times W \times 54$ anchors in total for the feature map with width W and height H.

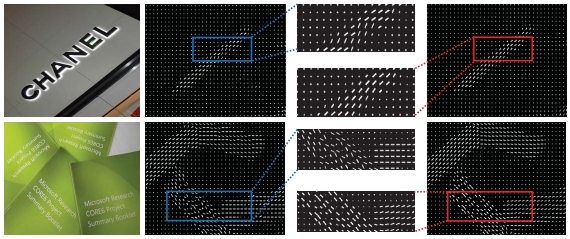
3.5. Learning of Rotated Proposal

As the R-anchors are generated, the sampling strategy for R-anchors is needed for network learning. We first define the Intersection-over-Union (IoU) overlap as the overlap between the skew rectangles of ground truth and R-anchor. Then Positive R-anchors are featured with: (i) the highest IoU overlap or IoU larger than 0.7 with ground truth, and (ii) intersection angle with the ground truth less than $\frac{\pi}{12}$. Negative R-anchors are defined as: (i) IoU lower than 0.3, or (ii) IoU larger than 0.7 but the intersection angle with ground truth larger than $\frac{\pi}{12}$. Regions which are not positive or negative are not used during training.



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

Figure 4: Visualization of different multi-task loss values. The orientation is the direction of the white line on each point, and longer lines indicate higher response score for text. (a) Input images; (b) 0 iteration; (c) 15,000 iterations; (d) 150,000 iterations.



(a) input image (b) w/o. reg. (c) with reg.

Figure 5: Visualization of the impact for regression. (a) Input images; (b)(c) orientation and response of the anchors without regression (b) and with regression (c).

Our loss function for the proposal uses the form of multi-task loss [8], which is defined as:

$$L(p, l, v^*, v) = L_{\text{cls}}(p, l) + \lambda l L_{\text{reg}}(v^*, v) \tag{1}$$

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

$$L_{\text{cls}}(p, l) = -\log p_l \tag{2}$$

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

$$L_{\text{reg}}(v^*, v) = \sum_{i \in \{x,y,h,w,\theta\}} \text{smooth}_{L_1}(v_i^* - v_i) \tag{3}$$

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases} \tag{4}$$

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

$$\begin{aligned} v_x &= \tfrac{x - x_a}{w_a}, v_y = \tfrac{y - y_a}{h_a} \\ v_h &= \log \tfrac{h}{h_a}, v_w = \log \tfrac{w}{w_a}, v_\theta = \theta \ominus \theta_a \end{aligned} \tag{5}$$

$$\begin{aligned} v_x^* &= \tfrac{x^* - x_a}{w_a}, v_y^* = \tfrac{y^* - y_a}{h_a} \\ v_h^* &= \log \tfrac{h^*}{h_a}, v_w^* = \log \tfrac{w^*}{w_a}, v_\theta^* = \theta^* \ominus \theta_a \end{aligned} \tag{6}$$

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

As we describe in the prior 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 less than $\frac{\pi}{12}$, so that every R-anchor has its fitting range, which we call it fit domain. When 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 6 orientations divide the angle range $[-\frac{\pi}{4}, \frac{3\pi}{4})$ into 6 equal parts. Thus ground truth in any orientation can be fitted with an R-anchor of appropriate fit domain. Figure 5 shows a comparison of the utility of regression term. We can observe that the orientations of the regions are similar in a neighborhood region.

In order to verify the ability of the network for learning text region orientation, we visualize the intermediate results in Figure 4. For an input image, the feature maps of RRPN training after different iterations are visualized. The short white line on feature map represents the R-anchor with the highest response to the text instance. The orientation of short line is orientation of this R-anchor, while short line length indicates the level of response. We can observe that brighter field of feature map focuses on the text region and other region becomes darker after 150,000 iterations. Moreover, the orientations of the regions become closer to the orientation of text instance as the iteration increases.

3.6. Accurate Proposal Refinement

Skew IoU Computation The rotation proposals can be generated in any orientations, so IoU computation on 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 Skew IoU computation with thought of triangulation [25] and Figure 6 shows the geometric principles.

Skew Non-Maximum Suppression (Skew-NMS) Traditional NMS only takes the IoU factor into consideration (*e.g.* IoU threshold is 0.7), but it is insufficient for arbitrary-oriented proposals. For instance, anchor with ratio 1:8 and

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

Algorithm 1 IoU computation

Input: Rectangles $R_1, R_2, ..., R_N$
IoU$[1, N][1, N] \leftarrow 0$
for each pair of R_i, R_j $(i < j)$ **do**
    Point set $PSet \leftarrow \emptyset$
    Add intersection points of R_i and R_j to $PSet$
    Add the vertices of R_i inside R_j into $PSet$
    Add the vertices of R_j inside R_i into $PSet$
    Sort $PSet$ to anti-clockwise order
    Compute intersection I of $PSet$ by triangulation
    IoU$(i, j) \leftarrow$ (Area(R_i) + Area(R_j) - I)/I
end for
return IoU

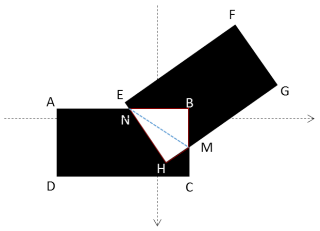
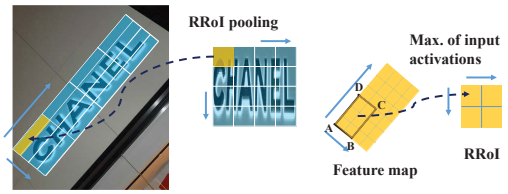


Figure 6: Intersection area computation of two rectangles in arbitrary orientation. First add intersection points N,M and inside vertices H,B into $PSet$, sort $PSet$ to get convex polygon $NHMB$, and then calculate the intersection area $S_{NHMB} = S_{\Delta NHM} + S_{\Delta NMB}$



(a) (b)

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

angle difference of $\frac{\pi}{12}$ are with the IoU of 0.31, which are less than 0.7; however, it may be annotated as positive samples. Therefore, the Skew-NMS consists of 2 phases: (i) keep the max IoU for proposals with IoU larger than 0.7; (ii) if all proposals with an IoU in range $[0.3, 0.7]$, keep the proposal with minimum angle difference with ground truth (the angle difference shall be less than $\frac{\pi}{12}$).

3.7. RRoI Pooling Layer

As presented in Fast-RCNN [8], RoI pooling layer extracts a fixed-length feature vector from the feature map for each proposal. Each feature vector is feed into fully-connected layers that finally branch in to the sibling cls and reg layers and the outputs are predicted localization and class of an object in an input image. As the feature map of image only needs to be computed once per image rather than computed 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 hyper-parameters that are independent of any RoI.

Algorithm 2 RRoI pooling

RRoI_width $\leftarrow W_r$
RRoI_height $\leftarrow H_r$
RRoI_map[1,RRoI_height][1,RRoI_width] $\leftarrow 0$
for $i \in$ [1,RRoI_height],$j \in$ [1,RRoI_width] **do**
    $A, B, C, D \leftarrow$ affineTransformation(subRegion(i, j))
    RRoI_map$[i][j] \leftarrow$ max(featureMap(A, B, C, D))
end for
return RRoI_map

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 RRPN. We first set the RRoI layer hyper-parameters 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 Figure 7(a)); each subregions have the same orientation as the proposal. Figure 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 affine transformation and grouped to range the border of subregion. Then max pooling is performed in every subregion and max-pooled values are saved in the matrix of each RRoI; the pseudo-code for RRoI pooling is shown in Algorithm 2. Comparing with RoI pooling, RRoI Pooling can pool any regions with various angle, aspect ratio, or scale, into a fix size feature map. Finally, the proposals are transferred into RRoIs and sent to the classifiers to give the result of text or background.

4. Experiments

We perform the rotation-based framework on three popular text detection benchmarks: MSRA-TD500 [38], ICDAR2015 [18] and ICDAR2013 [19]; we follow the evaluation protocol of these benchmarks. MSRA-TD500 dataset contains 300 training images and 200 testing images. Annotations of the images consist of both position and orientation of each text instance, and the benchmark can be used to evaluate text detection performance over the multi-oriented text instance. As the dataset of MSRA-TD500 is relatively smaller, its experiments are designed for exploiting alter-

a.	b.	c.	d.	e.	P	R	F	ΔF
Faster-RCNN [27]					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%
√	√	√	√	√	82.1%	67.7%	74.2%	18.3%

Table 1: Evaluation on MSRA-TD500 with different strategies and settings. Experiments of Faster-RCNN are based on the original source code. P, R and F are abbreviation of Precision, Recall, and F-measure respectively. ΔF is the improvement of F-measure over baseline. The strategies include: a. context of text region; b. training dataset enlargement; c. border padding; d. scale jittering; e. post processing.

native settings. ICDAR 2015 is released for the text localization of incidental scene text challenge (Task 4.1) in ICDAR 2015 Robust Reading Competition, which has 1500 images in total. Different from previous ICDAR Robust Reading Competition, the text instance annotations are with 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 ICDAR 2013 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 see the adaptability of our approach on specific orientation.

Implementation Details. Our network is initialized by pre-training of a model for ImageNet classification [30]. The weights of the network are updated by using a learning rate of 10^{-3} for the first 200,000 iterations and 10^{-4} for next 100,000 iterations, weight decay of 5×10^{-4} and a momentum of 0.9.

Due to our different R-anchor strategy, the total number of proposals for each image is nearly 6 times of that in previous approaches such as Faster-RCNN. In order to make the efficient detection, we filter R-anchors of breaking the border of image. Therefore, the speed of our system is similar with previous works in both training and testing stage.

4.1. MSRA-TD500

We train the baseline system with 300 images from MSRA-TD500 training set; the input image patch is resized with long side of 1000 pixels. The evaluation result is precision of 57.4%, recall of 54.5%, and F-measure of 55.9%, which are much better performance than original Faster-RCNN with P, R and F of 38.7%, 30.4% and 34.0%. We make the comparison between rotation and horizontal

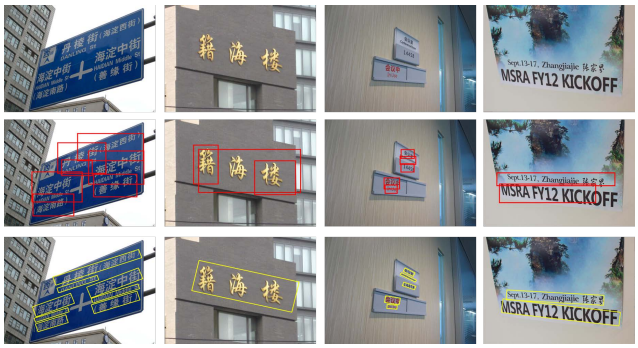


Figure 8: Comparison of rotation and horizontal region proposal. First row: original images; second row: text detection based on horizontal region proposal; third row: text detection based on rotation region proposal.

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%

Table 2: Exploit the text region context by enlargement of the text bounding box with different factor of original size.

Small instances Hard situations

Figure 9: Examples of the failed detection on MSRA-TD500. The red box is the detection result and the green box is groundtruth.

region proposal and some detection results are illustrated in Figure 8. The rotation based approach is able to have accurate detection with less background area, which indicates the effectiveness of incorporation the rotation strategy.

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

Context of Text Region. Incorporating the contextual

information is proved to be useful for the general object detection task (*e.g.*, [26]) and we wonder whether it can promote a text detection system. We keep the center of the rotated bounding box and orientation, and enlarge both width and height by a factor of $1.X$ on the data preprocess step. During testing phase, we divide the enlargement for every proposal. As shown in Table 2, all the experiments are with obvious increase in F-measure. The reason may probably be that as the bounding box become larger, more context information of text instance is grasped and the information of orientation can be better captured, so that the orientation of the proposals can be more precisely predicted.

Training Dataset Enlargement. we adopt HUST-TR400 (contains 400 images with text instance annotated with the same parameters as MSRA-TD500) [37] as an additional dataset and form the training set of 700 images from both datasets. There is a significant improvement of all the measurements and the F-measure is 60.8%, showing that the network is better trained and more robust when handling noisy inputs.

Border Padding. With our filtering strategy, most of the breaking bound R-anchors are eliminated. However, as the bounding box is rotated with 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 of each side to reserve more positive proposals. Experiment shows that adding border padding to the image improves the detection results. Besides, combining border padding with enlargement of text region and training dataset makes a further improvement with F-measure of 63.3%.

Scale Jittering. There are still a number of small text regions in both training dataset, and we add the scale jittering for the robustness of our system. The image patch is re-scaled with long side of a random size less than 1300 pixels and then sent into the network. The F-measure based on scale jittering is 69.3%, which has 6% improvement over the experiment without jittering.

Post Processing. The annotation of MSRA-TD500 prefers to label the region of a whole text line. Thus, the length of text line is not with a fixed range, sometimes to be very long. However, the ratios for the R-anchor are fixed and may not be large enough to cover all the length, which leads to several short bounding box results for a single text region. To handle this extremely long text line issue, a post-processing step is incorporated by combine multiple short detection boxes in the neighborhood area to one proposal. With the post-processing, the performance further boost to F-measure of 74.2%.

4.2. ICDAR Benchmarks

ICDAR 2015. Following the strategies and parameters on MSRA-TD500 evaluation, we train the baseline system with the training dataset of ICDAR2015 containing 1000 images with 10886 text instances. The evaluation result is precision 45.42%, recall 72.56%, and F-measure 55.87%.

Approach	RRPN		[39]	[32]	[44]
#Image	2077	1229	1529	3000	1529
Train 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%

Table 3: Training set and results for ICDAR2015. I*XX* indicate ICDAR20*XX* training set; M500 indicates MSRA-TD500, and CA indicates data Collected by Authors of [32].

Figure 10: Text detection in ICDAR2015 with the model trained on ICDAR2015 training set (including all unreadable text instances). The green areas are the positive detections with IoU > 0.5, red areas represent false detection.

As MSRA-TD500 tends to provide text line ground truth and ICDAR is word level annotations, the precision of our approach is lower than other methods that reach the same magnitude of F-measure. We analysis the text detection results and find three reasons. First, even we resize the images using the scale jittering strategy, some of the incidental text regions are still too small. Second, there exist some small unreadable text instances (labeled with '###') in ICDAR2015 training set (some are shown in Figure 10), which may lead to some false detection results of text-like instance. Finally, our training dataset is much smaller than previous approaches such as [39, 32, 44].

To handle the small text region issue, we make a larger scale by jittering the image patch with long side in a random size less than 1700 pixels before sending into the network. We also check the impact of small unreadable text instances by randomly removing these instances from training set. Figure 11 displays the curves of the measurements. The recall rate remains the same level around 72%-73% unless we remove all the unreadable instances, while precision significantly increases with the proportion. Therefore, we randomly remove 80% unreadable text instances on 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 [19], ICDAR2003 [21] and SVT [35]. As listed in Table 3, the training images for different approaches are in the same order of magnitude and ours achieve better performance.

ICDAR 2013. In order to examine the adaptability of our approach, we also conduct experiments on horizontal-based ICDAR2013 benchmark. We reuse the model trained

MSRA-TD500					ICDAR2015				ICDAR2013			
Approach	P	R	F	Time	Approach	P	R	F	Approach	P	R	F
Yao *et al*. [37]	64	62	61	7.2s	CTPN [32]	74	52	61	Faster-RCNN [27]	75	71	73
Yin *et al*. [42]	71	61	65	0.8s	Yao *et al*. [39]	72	59	65	Gupta *et al*. [9]	92	76	83
Kang *et al*. [17]	71	62	66	-	SCUT_DMP_v2	77	77	77	Yao *et al*. [39]	89	80	84
Yin *et al*. [41]	81	63	71	1.4s	MSRA-v1	85	74	79	DeepText [45]	85	81	85
Zhang *et al*. [44]	83	67%	74	2.1s	SRC-B	84	80	82	CTPN [32]	93	83	88
Yao *et al*. [39]	77	75	76	0.62s	NLPR-CASIA	85	83	84	NLPR-CASIA	95	89	91
RRPN	82	68	74	0.3s	RRPN	82	73	77	RRPN	90	72	80
RRPN*	82	69	75	0.3s	RRPN*	84	77	80	RRPN*	95	88	91

Table 4: Comparison with state-of-the-arts on three benchmarks. Faster-RCNN results in ICDAR2013 are reported in [45].

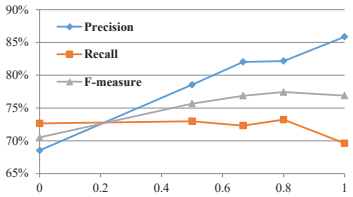


Figure 11: The effect of unreadable text instance proportion using cross-validation on training set. Horizontal axis represents the removing proportion of unreadable text instances and vertical axis represents the percentage of F-measure.

(a) MSRA-TD500

(b) ICDAR2015

(c) ICDAR2013

Figure 12: Text detection results on the benchmarks.

for ICDAR2015 and the 5-tuple rotation proposals are fit into horizontal-aligned rectangles. The result is Precision 90.22%, Recall 71.89%, and F-measure 80.02% under ICDAR 2013 evaluation protocol. As shown in Table 4, there are 7% improvement comparing with Faster-RCNN, which confirms the robustness of our detection framework with rotation factor.

4.3. Comparisons with State-of-the-art

The experimental results of our method compared with state-of-the-art approaches are given in Table 4. As the RRPN models are trained separately for MSRA-TD500 and ICDAR, we also train an unified model (RRPN*) trained on all of the training sets for the generalization issue. For MSRA-TD500 dataset, the performance of our RRPN reaches the same magnitude of start-of-the-art approaches, such as [39] and [44]. As our system make the end-to-end text detection, it is more efficient than others with only 0.3s processing time per testing image. For ICDAR benchmarks, the substantial performance gains over the published works confirm the effectiveness of using rotation region proposal and rotation RoI for text detection task. The recent work DeepText [45] is also detection based approach, but is based on the Inception structure. We believe that our rotation-based framework is also complementary to Inception since they focus on different levels of information. Some detection results on the benchmarks are illustrated in Figure 12.

5. Conclusion

In this paper, we have introduced a rotation based detection framework for arbitrary-oriented text detection. Inclined rectangle proposals are generated with the text region orientation angle information from higher convolutional layer of network, which is able to detect text in multi-orientation. A novel RRoI pooling layer is also designed and adapted to the rotated RoIs. Experimental comparisons with state-of-the-arts on MSRA-TD500, ICDAR2013 and ICDAR2015 show the effectiveness and efficiency of proposed RRPN and RRoI on text detection task.

References

[1] A. Bissacco, M. Cummins, Y. Netzer, and H. Neven. Photoocr: Reading text in uncontrolled conditions. In *ICCV*, pages 785–792, 2013. 1

[2] D. Chen and J. Luettin. A survey of text detection and recognition in images and videos. Technical report, 2000. 2

[3] X. Chen and A. L. Yuille. Detecting and reading text in natural scenes. In *CVPR*, pages 366–373, 2004. 1

[4] X. Chen and A. L. Yuille. Detecting and reading text in natural scenes. In *CVPR*, volume 2, pages II–366, 2004. 2

[5] H. Cho, M. Sung, and B. Jun. Canny text detector: Fast and robust scene text localization algorithm. In *CVPR*, 2016. 2

[6] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009. 2

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

[8] R. Girshick. Fast r-cnn. In *ICCV*, 2015. 2, 4, 5

[9] A. Gupta, A. Vedaldi, and A. Zisserman. Synthetic data for text localisation in natural images. In *CVPR*, pages 2315–2324, 2016. 8

[10] T. He, W. Huang, Y. Qiao, and J. Yao. Accurate text localization in natural image with cascaded convolutional text network. *arXiv preprint arXiv:1603.09423*, 2016. 1

[11] T. He, W. Huang, Y. Qiao, and J. Yao. Text-attentional convolutional neural network for scene text detection. *IEEE Trans. IP*, 25(6):2529–2541, 2016. 2

[12] W. Huang, Y. Qiao, and X. Tang. Robust scene text detection with convolution neural network induced mser trees. In *ECCV*, pages 497–511, 2014. 1

[13] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman. Reading text in the wild with convolutional neural networks. *IJCV*, 116(1):1–20, 2014. 2

[14] M. Jaderberg, K. Simonyan, A. Zisserman, and k. kavukcuoglu. Spatial transformer networks. In *NIPS*, pages 2017–2025, 2015. 2

[15] M. Jaderberg, A. Vedaldi, and A. Zisserman. Deep features for text spotting. In *ECCV*, pages 512–528, 2014. 1, 2

[16] K. Jung, K. I. Kim, and A. K. Jain. Text information extraction in images and video: a survey. *Pattern Recognition*, 37(5), 2004. 2

[17] L. Kang, Y. Li, and D. Doermann. Orientation robust text line detection in natural images. In *CVPR*, 2014. 8

[18] D. Karatzas, L. Gomez-Bigorda, A. Nicolaou, et al. Icdar 2015 competition on robust reading. In *ICDAR*, pages 1156–1160, 2015. 2, 5

[19] D. Karatzas, F. Shafait, S. Uchida, et al. Icdar 2013 robust reading competition. In *ICDAR*, 2013. 2, 5, 7

[20] 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. PAMI*, 25(12):1631–1639, 2003. 2

[21] S. M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, and R. Young. ICDAR 2003 robust reading competitions. In *ICDAR*, 2003. 7

[22] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image & Vision Computing*, 22(10):761–767, 2004. 1, 2

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

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

[25] D. A. Plaisted and J. Hong. A heuristic triangulation algorithm. *Journal of Algorithms*, 8(3):405–437, 1987. 4

[26] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016. 7

[27] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Trans. PAMI*, 2016. 2, 3, 6, 8

[28] A. Risnumawan, P. Shivakumara, C. S. Chan, and C. L. Tan. A robust arbitrary text detection system for natural scene images. *Expert Systems with Applications*, 41(18):8027–8048, 2014. 2

[29] A. Shahab, F. Shafait, and A. Dengel. Icdar 2011 robust reading competition challenge 2: Reading text in scene images. In *ICDAR*, pages 1491–1496, 2011. 2

[30] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 2, 3, 6

[31] S. Tian, Y. Pan, C. Huang, S. Lu, K. Yu, and C. L. Tan. Text flow: A unified text detection system in natural scene images. In *ICCV*, 2015. 1

[32] Z. Tian, W. Huang, T. He, P. He, and Y. Qiao. Detecting text in natural image with connectionist text proposal network. In *ECCV*, pages 56–72, 2016. 2, 7, 8

[33] S. Uchida. Text localization and recognition in images and video. In *Handbook of Document Image Processing and Recognition*, pages 843–883. 2014. 2

[34] J. R. R. Uijlings, K. E. A. V. D. Sande, T. Gevers, and A. W. M. Smeulders. Selective search for object recognition. *IJCV*, 104(2):154–171, 2013. 2

[35] K. Wang and S. Belongie. Word spotting in the wild. In *ECCV*, pages 591–604, 2010. 1, 2, 7

[36] T. Wang, D. J. Wu, A. Coates, and A. Y. Ng. End-to-end text recognition with convolutional neural networks. In *ICPR*, pages 3304–3308, 2012. 2

[37] C. Yao, X. Bai, and W. Liu. A unified framework for multioriented text detection and recognition. *IEEE Trans. IP*, 23(11):4737–49, 2014. 7, 8

[38] C. Yao, X. Bai, W. Liu, Y. Ma, and Z. Tu. Detecting texts of arbitrary orientations in natural images. In *CVPR*, pages 1083–1090, 2012. 2, 5

[39] C. Yao, X. Bai, N. Sang, X. Zhou, S. Zhou, and Z. Cao. Scene text detection via holistic, multi-channel prediction. *arXiv preprint arXiv:1606.09002*, 2016. 1, 2, 7, 8

[40] Q. Ye and D. Doermann. Text detection and recognition in imagery: A survey. *IEEE Trans. PAMI*, 37(7):1480–1500, 2015. 2

[41] X. C. Yin, W. Y. Pei, J. Zhang, and H. W. Hao. Multi-orientation scene text detection with adaptive clustering. *IEEE Trans. PAMI*, 37(9), 2015. 8

[42] X. C. Yin, X. Yin, K. Huang, and H. W. Hao. Robust text detection in natural scene images. *IEEE Trans. PAMI*, 36(5):970–83, 2014. 8

[43] S. Zhang, M. Lin, T. Chen, L. Jin, and L. Lin. Character proposal network for robust text extraction. In *ICASSP*, pages 2633–2637, 2016. 2

[44] Z. Zhang, C. Zhang, W. Shen, C. Yao, W. Liu, and X. Bai. Multi-oriented text detection with fully convolutional networks. In *CVPR*, 2016. 1, 7, 8

[45] Z. Zhong, L. Jin, S. Zhang, and Z. Feng. Deeptext: A unified framework for text proposal generation and text detection in natural images. *arXiv preprint arXiv:1605.07314*, 2016. 2, 8

[46] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In *ECCV*, pages 391–405, 2014. 2