

Article

# R-YOLO: A Real-Time Text Detector for Natural Scenes with Arbitrary Rotation

Xiqi Wang <sup>1</sup>, Shunyi Zheng <sup>1,\*</sup>, Ce Zhang <sup>2,3</sup>, Rui Li <sup>1</sup> and Li Gui <sup>1,4,5</sup>

- <sup>1</sup> School of Remote Sensing and Information Engineering, Wuhan University, Wuhan 430079, China; wangxiqi@whu.edu.cn (X.W.); lironui@whu.edu.cn (R.L.); whuguili@whu.edu.cn (L.G.)  
<sup>2</sup> Lancaster Environment Centre, Lancaster University, Lancaster LA1 4YQ, UK; c.zhang9@lancaster.ac.uk  
<sup>3</sup> UK Centre for Ecology & Hydrology, Library Avenue, Bailrigg, Lancaster LA1 4AP, UK  
<sup>4</sup> Collaborative Innovation Center of Geospatial Technology, Wuhan University, Wuhan 430079, China  
<sup>5</sup> School of Electronic Information, Wuhan University, Wuhan 430079, China  
\* Correspondence: syzheng@whu.edu.cn

**Abstract:** Accurate and efficient text detection in natural scenes is a fundamental yet challenging task in computer vision, especially when dealing with arbitrarily-oriented texts. Most contemporary text detection methods are designed to identify horizontal or approximately horizontal text, which cannot satisfy practical detection requirements for various real-world images such as image streams or videos. To address this lacuna, we propose a novel method called Rotational You Only Look Once (R-YOLO), a robust real-time convolutional neural network (CNN) model to detect arbitrarily-oriented texts in natural image scenes. First, a rotated anchor box with angle information is used as the text bounding box over various orientations. Second, features of various scales are extracted from the input image to determine the probability, confidence, and inclined bounding boxes of the text. Finally, Rotational Distance Intersection over Union Non-Maximum Suppression is used to eliminate redundancy and acquire detection results with the highest accuracy. Experiments on benchmark comparison are conducted upon four popular datasets, i.e., ICDAR2015, ICDAR2013, MSRA-TD500, and ICDAR2017-MLT. The results indicate that the proposed R-YOLO method significantly outperforms state-of-the-art methods in terms of detection efficiency while maintaining high accuracy; for example, the proposed R-YOLO method achieves an F-measure of 82.3% at 62.5 fps with 720 p resolution on the ICDAR2015 dataset.

**Keywords:** scene text detection; arbitrarily-oriented text; rotation anchor; convolutional neural network; YOLOv4



**Citation:** Wang, X.; Zheng, S.; Zhang, C.; Li, R.; Gui, L. R-YOLO: A Real-Time Text Detector for Natural Scenes with Arbitrary Rotation. *Sensors* **2021**, *21*, 888. <https://doi.org/10.3390/s21030888>

Academic Editor: Sylvie Le Hegarat-Masclé

Received: 19 December 2020

Accepted: 25 January 2021

Published: 28 January 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Texts in natural scenes, including road traffic signs, billboards, and shopping mall signs, etc. play a crucial role in our daily lives, providing essential information on society and our environment. As a prerequisite for text recognition, real-time text detection is essential in the process of text information extraction and natural language understanding. For applications with high real-time requirements, such as real-time text translation, text interpretation for assisting the visually impaired, intelligent driving and autonomous robot navigation, even a slight delay could have catastrophic consequences. Compared with standard text on documents or the internet, texts in natural scenes are discrepant, having varied sizes, font type, color, language, and orientation. Furthermore, they often have varying illumination intensities, complex backgrounds, and multiple photographing angles, causing challenges in text detection and recognition.

Over the past decade, many state-of-the-art methods have been developed to handle the tasks mentioned above [1–13], wherein horizontal or approximately horizontal text regions are detected with encouraging accuracy. However, text regions in natural scenes are often not horizontal, which limits the practical utility of those methods. In recent

times, certain methods have attempted to address the arbitrarily-oriented text detection problem [14–25]. In general, these methods follow a two-stage strategy based on deep CNN. The detection process is as follows: first, region proposals are generated through a selective search method or a regional proposal network; then, inclined bounding boxes regression is used for inclined proposals. Despite achieving excellent accuracy, the computational efficiency of these methods in real-time applications is far from satisfactory.

To achieve high-precision and real-time detection of arbitrarily-oriented text in complex environments, we propose a one-stage text detector called Rotational You Only Look Once (R-YOLO), based on the YOLOv4 architecture [26]. Particularly, the RDIoU-NMS algorithm is proposed to increase the accuracy and reduce the error suppression of arbitrary-oriented text detection. Compared with the traditional IoU-NMS algorithm, the RDIoU-NMS algorithm not only considers the angle factor of the inclined bounding boxes but also considers the center point distance between the two boxes. We also design a rotating anchor box with angle information to represent text bounding boxes in different directions and improve the loss function of yolov4 to achieve the inclined bounding box regression. In our proposed method, like YOLOv4, Cross Stage Partial Darknet53 (CSPDarknet53) is used as the backbone network. In the CSPDarknet53 Network, the  $1 \times 1$  convolution kernel is widely used to reduce the dimension of the feature map, thereby improving the calculation efficiency. Besides, the R-YOLO method adopts a one-step strategy, which uses a CNN network to directly predict the categories and locations of different targets without relying on the regional proposal. This makes R-YOLO one of the fastest inclined text detection methods. Compare with YOLOv4, a fourth-scale detection branch is added to the architecture of R-YOLO, which can effectively extract shallow features and fuse them with deep features, thereby effectively improve the detection ability of small-scale text.

There are two advantages of R-YOLO. First, as a one-stage method, R-YOLO can detect arbitrarily-oriented text in real-time. Second, our method has four-scale detection branches, which provides generalization capability for small-scale text detection. We choose four popular benchmark datasets (ICDAR2015, ICDAR2013, MSRA-TD500, and ICDAR2017-MLT) to evaluate the performance of the proposed method in terms of classification accuracy and computational efficiency. The major contribution of this paper can be summarized as follows:

1. A novel framework is developed to detect scene texts in arbitrary orientations using a one-stage strategy, where a fully convolutional network (FCN) is employed to generate inclined bounding boxes for text, thereby avoiding the redundant and time-consuming intermediate steps adopted in existing methods. An anchor box with rotation angle information is designed to replace the traditional axis alignment anchor box so that text detection can be adapted to any rotation angle. A new algorithm, RDIoU-NMS, is proposed to substitute the traditional IoU-NMS algorithm.
2. The 4th scale is added into the architecture of YOLOv4 to enhance the performance of detecting small-size natural-scene text.

The remainder of this paper is organized as follows. Section 2 provides an overview of related work. Details of the proposed method are presented in Section 3, followed in Section 4 by a description of numerical experiments conducted to test the performance of the proposed method. Section 5 presents our conclusions and maps out our future work.

All of our code is available publicly at <https://github.com/wxq-888/R-YOLO>.

## 2. Related Work

Scene text detection and recognition have been an active research topic in computer vision over the past few decades. Comprehensive surveys and detailed analyses have been conducted [27–29]. Traditional natural scene text detection methods rely heavily on handcrafted features to distinguish between text and non-text components in natural scene images, including methods employing sliding window (SW) and connected component (CC) techniques [1–4]. SW methods move a multi-scale detection window through all possible locations in an image and then use a pre-trained classifier to identify whether the

detection window contains text. However, the SW process ends up creating a large number of redundant detection windows, which severely limits its efficiency. Among CC-based methods, Maximum Stable Extreme Regions (MSER) [13] and Stroke Width Transform (SWT) [4] are the most representative methods, where connected components are extracted as character candidates to be classified as text or non-text. The MSER method has achieved acceptable performance in ICDAR2013 [30] and ICDAR2015 [31] competitions. However, these traditional methods lag behind deep neural network-based methods in accuracy and adaptability, particularly when encountering challenging scenes such as those with low spatial resolution and geometric distortion.

Recently, with the rapid development of deep learning, natural scene text detection has entered a new era. A wide range of CNN-based text detection methods have been developed and become mainstream with tremendous success. From the perspective of the method used, text detection methods based on deep learning can be divided into three main categories: segmentation-based methods, hybrid methods, and bounding box regression-based methods.

Segmentation-based methods [14–18] strive to address the issue by segmenting the text region from the background and obtaining the boundary box of the text through additional steps. EAST [19] generated a text region map using a U-shape network [32]. It regressed the oriented rectangles or quadrilaterals based on the same feature to create the score map. TextFuseNet [33] considered the text detection task to be an instance segmentation task, where character-, word- and global-level features were extracted and embedded into a multi-path fusion architecture for text detection. The network has the advantage of high detection accuracy, but computational efficiency is low. Hybrid methods [20,21] used segmentation-based methods to predict the score map of the text; thereafter the text bounding box is acquired through regression.

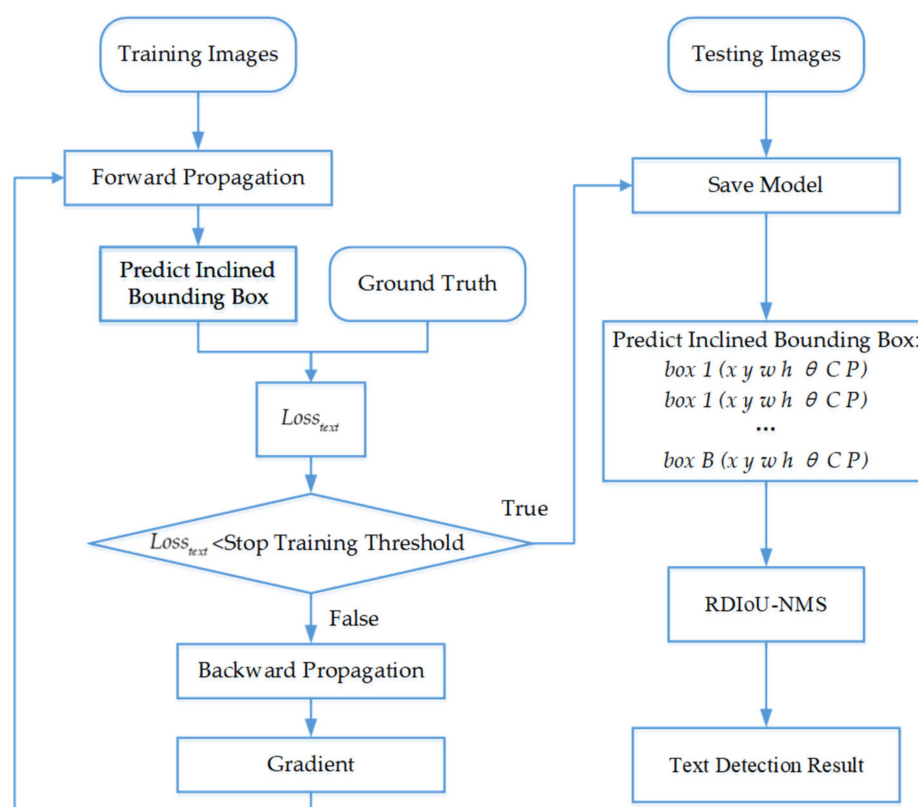
Bounding box regression methods can be categorized into either two-stage methods or one-stage methods. Two-stage methods rely on region proposals. The most representative network is Faster R-CNN [34]. Many state-of-the-art methods such as R2CNN [22] and RRPN [23] are designed on the basis of faster R-CNN. In R2CNN [22], Region-of-Interest (RoI) Pooling with varied pooling sizes was performed several times on the axis-aligned region proposals generated by RPN [34] and the concatenated features were used to classify the proposal, where both axis-aligned box and inclined region box were estimated. R2CNN adopted a single detection scale design and cannot perform multi-scale detection tasks. RRPN [23] method incorporated the rotation factor into the region proposal network and extended the RoI pooling layer into the rotation RoI pooling layer to realize text detection rotation. In another study [35], a two-stage detection scheme based on Scale-based Region Proposal Network (SRPN) was proposed. In the first stage, three tasks of text and non-text classification, text scale estimation, and text boundary determination were performed. The second stage employed a detector to predict the text boundary boxes in text region proposals from the first stage. By contrast, one-stage methods estimate the candidate targets directly, without relying on the region proposal. Typical networks are YOLOv3 [36], YOLOv4 [26], and Single Shot Multibox Detector (SSD) [37]. TextBoxes++ [24] is an end-to-end fast scene text detector with a single deep neural network that is inspired by SSD [37]. TextBoxes++ adopted a “long” convolution kernel to predict the bounding box, where a better receptive field was acquired to cover the long text area. During the test stage, cascaded NMS was used to solve the time-consuming problem of traditional NMS calculations. However, the angle and distance of inclined bounding boxes are not considered in the cascaded NMS method, which is prone to error suppression in dense text areas. He et al. [25] proposed a single-shot text detector that utilized an attention mechanism to enhance the text area in the image and reduced background interference in the convolutional features. RRD [20] used a regression branch and classification branch to perform feature extraction for text detection. The regression branch extracted rotation-sensitive features by actively rotating the convolutional filters, whereas the classification branch extracted rotation-invariant features by pooling the rotation sensitive features.

However, dual-branch feature extraction consumes a large amount of computational resources and has a limited contribution towards increasing accuracy, which cannot be satisfactory in real-time applications.

Compared with two-stage-based methods, a one-stage method regresses the bounding box directly from the convolutional feature maps without relying on the region proposal. Therefore, one-stage-based methods have an advantage in terms of computational efficiency, which is essential for fast detection in real-time applications. This paper presents a novel one-stage method (R-YOLO) for arbitrarily-oriented text detection using a fully convolutional network (FCN) model. The proposed method can not only perform multi-scale detection to obtain detection results but also conduct real-time detection tasks for real-world applications such as image streams or videos.

### 3. Proposed Method

In this section, the novel R-YOLO method is described in detail. R-YOLO is a detection model based on end-to-end deep learning that determines the inclined bounding boxes of the text in a natural scene image and classifies them in a single unified framework. Specifically, we have added a small-scale detection branch, proposed the RDIOU-NMS algorithm, improved the bounding box regression algorithm, and redesigned the loss function of the framework, so that it can realize text detection flexibly in natural scenes. The data processing flow is presented in Figure 1.

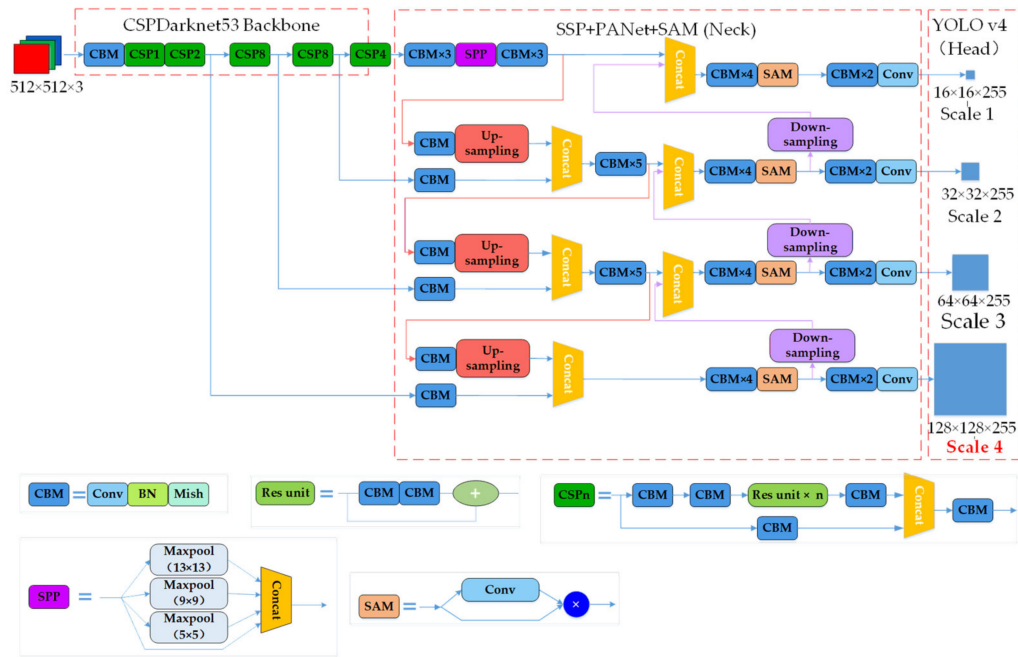


**Figure 1.** Flowchart of natural scene text detection.

#### 3.1. Architecture of R-YOLO

YOLO is a one-stage detection model that transforms target detection into a regression problem. The YOLO family has evolved progressively from YOLOv1 to YOLOv4. Compared with YOLOv3, YOLOv4 uses several effective tricks in target detection to improve the accuracy and efficiency of target detection significantly. Figure 2 clearly describes the neural network structure of the scene text detection algorithm. In the YOLOv4 network model, CSPDarknet53 is used as the backbone. CBM is the basic component of the YOLOv4

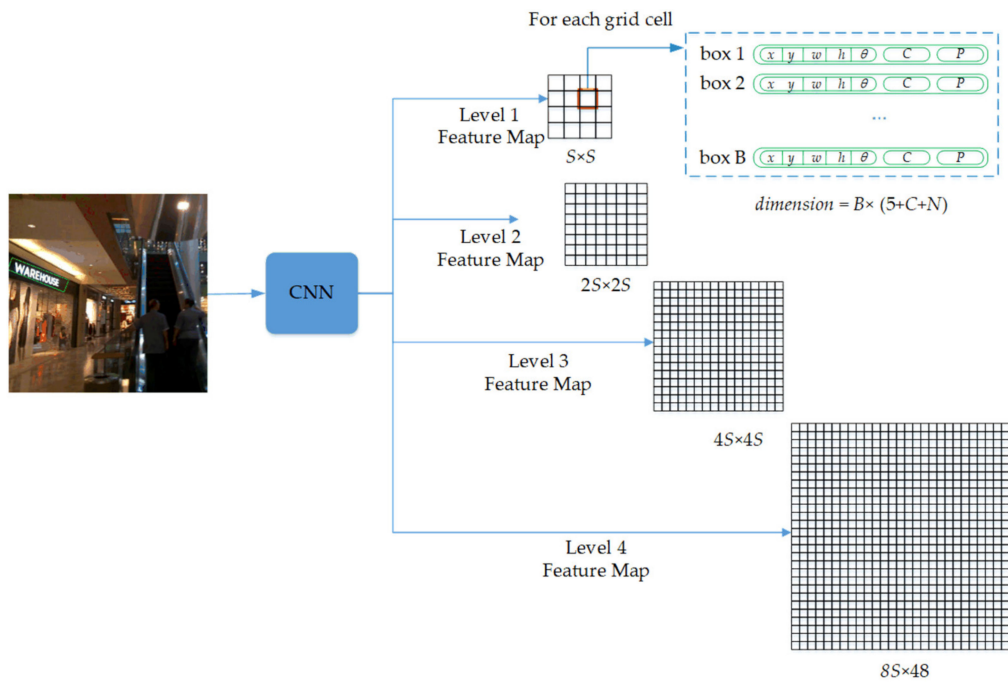
structure, consisting of a convolutional (Conv) layer, a batch normalization (BN) layer, and a Mish activation function. A Res unit exists to construct a deeper network. Center and scale prediction (CSP) consists of three convolutional layers and  $n$  Res unit modules that can enhance CNN's learning ability by dividing low-level features into two parts and then fusing cross-level features. The SPP is a spatial pyramid pooling module, which mainly transforms convolution features of different sizes into pooled features with the same length. It utilizes four scales of  $1 \times 1$ ,  $5 \times 5$ ,  $9 \times 9$ , and  $13 \times 13$  for maximum pooling. As in YOLOv3, three-scale detection heads are used in YOLOv4. In our proposed method, we expand the detection branches to four to deal with the detection of small-scale scene texts.



**Figure 2.** Network architecture of R-YOLO. To achieve fine-grained detection, four branches are used for object detection, with the scale of the feature map of each branch being different. Each branch predicts the confidence, class probability, and the inclined bounding boxes of the text.

R-YOLO inherits the FCN structure of YOLOv4. After the input image passes through the CNN, a feature map of four different sizes is obtained, which is divided into grids of  $S \times S$ ,  $2S \times 2S$ ,  $4S \times 4S$ ,  $8S \times 8S$  non-overlapping cells. For each cell, R-YOLO predicts the  $B$  inclined bounding box as illustrated by Figure 3. Each inclined bounding box contains  $(5 + C + N)$  detection attributes: five values for the position parameters ( $x$ ,  $y$ ,  $w$ ,  $h$ ,  $\theta$ ) of the inclined bounding box and  $C$  values for the confidence of the inclined bounding box. The confidence of text is defined as  $P(\text{Text}) \times \text{RDIOU}_{\text{pred}}^{\text{truth}}$ . If the inclined bounding box contains text, then  $P(\text{Text}) = 1$  and the confidence will be the RDIOU between the predicted inclined bounding box and ground truth. If no text lies in the inclined bounding box, the confidence is set as 0.  $N$  is the number of categories in each inclined bounding box. For natural-scene text detection,  $B = 12$ ,  $C = 1$ , and  $N = 1$ , so the output consists of four tensors of dimensions  $S \times S \times 84$ ,  $2S \times 2S \times 84$ ,  $4S \times 4S \times 84$ , and  $8S \times 8S \times 84$  corresponding to the four feature map levels, respectively.

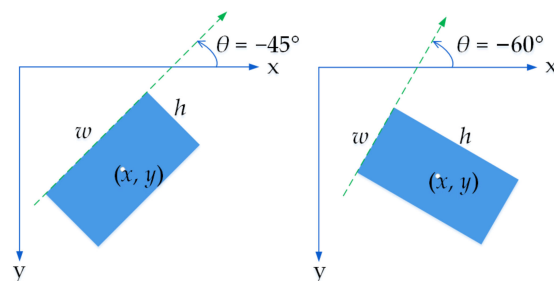




**Figure 3.** Output of R-YOLO consists of four tensors of dimension  $(S, S, B \times (5 + C + N))$ ,  $(2S, 2S, B \times (5 + C + N))$ ,  $(4S, 4S, B \times (5 + C + N))$ , and  $(8S, 8S, B \times (5 + C + N))$  which correspond to the four detection levels (scales). For the ICDAR2015 dataset,  $B = 12$ ,  $C = 1$ , and  $N = 1$ . If an input image size is  $512 \times 512$  pixels, the outputs are four tensors of size  $16 \times 16 \times 84$ ,  $32 \times 32 \times 84$ ,  $64 \times 64 \times 84$ , and  $128 \times 128 \times 84$ .

### 3.2. Inclined Bounding Box Representation

In the training stage, the ground truth of a text region set is represented by  $(x, y, w, h, \theta)$ , where the coordinates  $(x, y)$  are expressed as the coordinates of the ground-truth center point in the image coordinate system, as illustrated in Figure 4. The strategy for determining  $w, h$ , and  $\theta$  is as follows: the edge where the  $x$ -axis rotates counterclockwise and first parallel to the  $x$ -axis is defined as  $w$ . The angle between  $w$  and the  $x$ -axis is  $\theta$  and the range of  $\theta$  is between  $(-90^\circ, 0^\circ]$ . The side perpendicular to  $w$  is denoted as  $h$ , so the value of  $w$  is not necessarily smaller than  $h$ . There are three advantages of using the above representation strategy. First, the uncertainty of the  $\theta$  value due to the periodicity of the angle is eliminated. Second, it is convenient to carry out the regression operation of the inclined bounding box. Third, compared with the traditional 8-point representation of an inclined bounding box  $(x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4)$ , this representation can calculate the new ground-truth value easily after rotating training images.

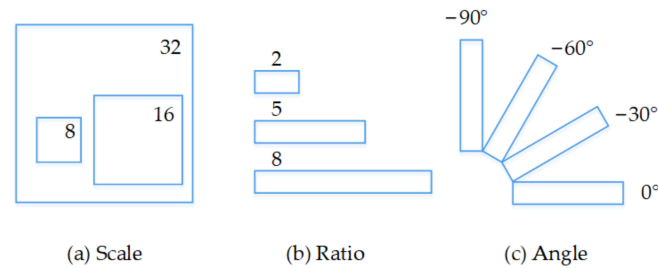


**Figure 4.** Inclined bounding box representation.

### 3.3. Rotation Anchor Box

As the ground-truth box of the text is labeled using a rectangle box with a rotation angle, the traditional anchor horizontal box, represented by scale and aspect ratio parameters only, is not suitable for text detection in natural scenes. Therefore, we design the rotation

anchors (R-anchors) by adjusting several parameters. First, the scales of anchor boxes are designed to be 8, 16, and 32 pixels. Second, as the text regions usually have different scales, we define three aspect ratios of 2:1, 5:1, and 8:1 to cover the text lines with multiple aspect ratios. Furthermore, an orientation parameter is added to control the anchor direction. Four different orientations, namely,  $0^\circ$ ,  $-30^\circ$ ,  $-60^\circ$ , and  $-90^\circ$  are used to ensure that the angle has the optimal initial value during the training process. The anchor strategy is summarized in Figure 5. Following the data representation steps above, an R-anchor is generated with five parameters ( $x, y, w, h, \theta$ ).



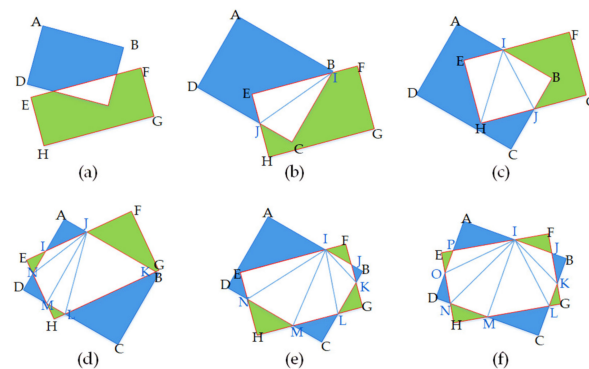
**Figure 5.** R-anchor defined in our framework.

### 3.4. RDIoU-NMS

Predicted inclined bounding boxes can be generated in any direction. The axis-aligned DIoU calculation method adopted in YOLOv4 might result in inaccuracies in the inclined bounding boxes DIoU calculation, which leads to erroneous results in the network learning process. We design a new method, named RDIoU, for the RDIoU calculation of the inclined bounding boxes, which considers not only the angle factor of the rotating bounding boxes but also the center point distance between the two boxes. The intersection of two inclined bounding boxes can create a variety of polygons, as shown in Figure 6. The vertices of the convex polygon can be sorted in the clockwise direction according to the coordinates in the image, and the triangle set can be acquired through triangulation. Taking Figure 6c as an example, the areas of all triangles are calculated and summed together. Finally, inclined RDIoU is derived as:

$$RDIoU = \frac{Area(IBJHE)}{Area(ABCD) + Area(EFGH) - Area(IBJHE)} - \frac{\rho^2(b_1, b_2)}{c^2} \quad (1)$$

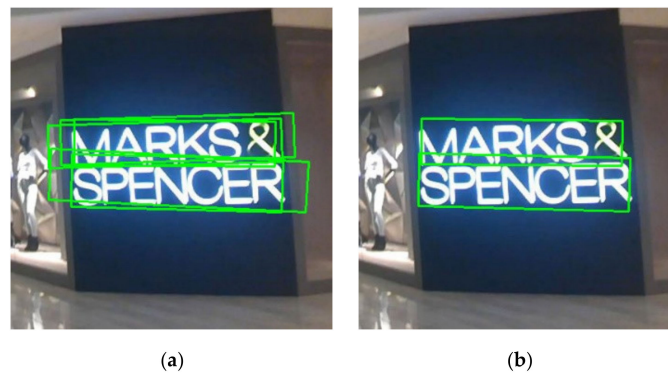
where  $b_1$  and  $b_2$  denote the central points of inclined bounding boxes  $B_1$  and  $B_2$ .  $\rho^2(b_1, b_2)$  is the Euclidean distance and  $c$  is the diagonal length of the smallest enclosing box covering the bounding box  $B_1$  and  $B_2$ .



**Figure 6.** Polygon formed by the intersection of two inclined rectangles: (a) 3 points, (b) 4 points, (c) 5 points, (d) 6 points, (e) 7 points, (f) 8 points. Considering example (c), first, intersection points I, J and inner vertices E, B, and H are sorted clockwise to obtain the convex polygon IBJHE, and then the intersection area  $Area(IBJHE) = Area(\triangle IBJ) + Area(\triangle IJH) + Area(\triangle IHE)$  is calculated.

In the target detection process, a large number of inclined bounding boxes coupled with confidence are generated at the same target position and there is significant overlap between inclined bounding boxes, as illustrated in Figure 7a, where the quantification of overlap degree is expressed by RDIoU. The RDIoU-NMS algorithm is utilized to filter out redundant inclined boxes and maintain the optimal inclined boxes, as shown in Figure 7b. The list of all inclined bounding boxes is  $B$ , the corresponding confidence score list is  $C$ , and the overlap threshold is  $N_t$ . Our goal is to get list  $D$ , which stores the optimal inclined bounding boxes. The steps of the RDIoU-NMS algorithm are as follows:

- Step 1: Sort the confidence scores in list  $C$  from large to small and adjust the order of bounding box storage in list  $B$  to make it consistent with the order of adjusted list  $C$ .
- Step 2: Take the inclined bounding box with the highest confidence as the target for comparison, delete it from list  $B$  and add it into the list  $D$  (initially  $D$  is empty). Calculate the RDIoU between the target inclined bounding boxes and remaining boxes in list  $B$ .
- Step 3: If the RDIoU is larger than the threshold  $N_t$ , delete the bounding box from list  $B$ .
- Step 4: Take the inclined bounding box with the second-highest confidence as the target for comparison and repeat Steps 2 and 3 until there are no more bounding boxes left in list  $B$ .



**Figure 7.** RDIoU-NMS. (a) All detected inclined bounding boxes; (b) Inclined bounding boxes after removing redundancy.

The pseudocode of the algorithm is summarized in Algorithm 1.

---

**Algorithm 1** Calculate RDIoU-NMS

---

**Input:**  $B = \{b_1, b_2, \dots, b_N\}$ ,  $C = \{c_1, c_2, \dots, c_N\}$ ,  $N_t$ , where  $B$  is the list of initial detection rotation boxes,  $C$  contains the corresponding detection confidence, and  $N_t$  is the NMS threshold.

**Output:**  $D, S$ , where  $D$  and  $S$  are the list of final prediction bounding boxes and the corresponding confidence respectively.

```

1. Begin
2.  $D \leftarrow \{\}, S \leftarrow \{\}$ 
3. While  $B \neq \text{empty}$  do
4.    $m \leftarrow \max C$ 
5.    $M \leftarrow b_m, T \leftarrow c_m$ 
6.    $D \leftarrow D \cup M, B \leftarrow B - M$ 
7.    $S \leftarrow S \cup T, C \leftarrow C - T$ 
8.   for  $b_i$  in  $B$  do
9.     if  $\text{RDIoU}(M, b_i) \geq N_t$  then
10.       $B \leftarrow B - b_i, C \leftarrow C - c_i$ 
11.   end
12. end
13. return  $D, S$ 
14. end

```

---



### 3.5. Learning of Text Detection

In YOLOv4, the loss function is defined as the sum of object classification loss, confidence loss, and bounding box regression loss.

$$L_{text} = L_{box} + L_{confidence} + L_{class} \quad (2)$$

Confidence and classification loss are defined as:

$$L_{confidence} = - \sum_{i=0}^{S \times S} \sum_{j=0}^B I_{ij}^{obj} [\hat{C}_i \log(C_i) + (1 - \hat{C}_i) \log(1 - C_i)] \\ - \lambda_{noobj} \sum_{i=0}^{S \times S} \sum_{j=0}^B I_{ij}^{noobj} [\hat{C}_i \log(C_i) + (1 - \hat{C}_i) \log(1 - C_i)] \quad (3)$$

$$L_{class} = - \sum_{i=0}^{S \times S} I_i^{obj} \sum_{c \in \text{classes}} [\hat{p}_i(c) \log(p_i(c)) + (1 - \hat{p}_i(c)) \log(1 - p_i(c))] \quad (4)$$

In Equation (3),  $S \times S$  represents the number of cells in the feature map.  $B$  is the number of predictors in each grid.  $I_{ij}^{obj}$  represents whether there is a target that falls in the  $j$ th bounding box of the  $i$ th grid cell.  $I_{ij}^{noobj}$  indicates whether no target object falls in the  $j$ th bounding box of the  $i$ th grid cell.  $\lambda_{noobj}$  refers to balancing parameters that control the trade-off between these terms.  $\hat{C}_i$  and  $C_i$  denote the true and predicted confidence, respectively. In Equation (4),  $I_i^{obj}$  denotes if the target appears in cell  $i$ .  $\hat{p}_i(c)$  refers to the true probability of the target, while  $p_i(c)$  refers to the predicted value.

In the official code of YOLOv4, two types of bounding box regression loss are implemented: Mean Square Error (MSE) loss and Complete Intersection over Union (CIoU) loss. In our method, we implement inclined boundary box regression based on MSE loss. Given the angle parameter of the inclined bounding box, the calculation complexity of the angle gradient increases during the backpropagation of the CIoU loss function.

The MSE loss is defined as:

$$L_{box} = \lambda_{coord} \sum_{i=0}^{S \times S} \sum_{j=0}^B I_{ij}^{obj} (2 - \hat{w}_i \times \hat{h}_i) [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] \\ + \lambda_{coord} \sum_{i=0}^{S \times S} \sum_{j=0}^B I_{ij}^{noobj} (2 - \hat{w}_i \times \hat{h}_i) [(w_i - \hat{w}_i)^2 + (h_i - \hat{h}_i)^2] \quad (5)$$

Here,  $\lambda_{coord}$  is a balancing parameter with the value set to 1.  $I_{ij}^{noobj}$  indicates whether no target object falls in the  $j$ th bounding box of the  $i$ th grid cell.  $(y_i, x_i, w_i, h_i)$  and  $(\hat{y}_i, \hat{x}_i, \hat{w}_i, \hat{h}_i)$  represent the center coordinate, height, and width of the predicted box and the ground truth, respectively.

The CIoU loss is defined as:

$$L_{box} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v \quad (6)$$

$$\alpha = \frac{v}{(1 - IoU) + v} \quad (7)$$

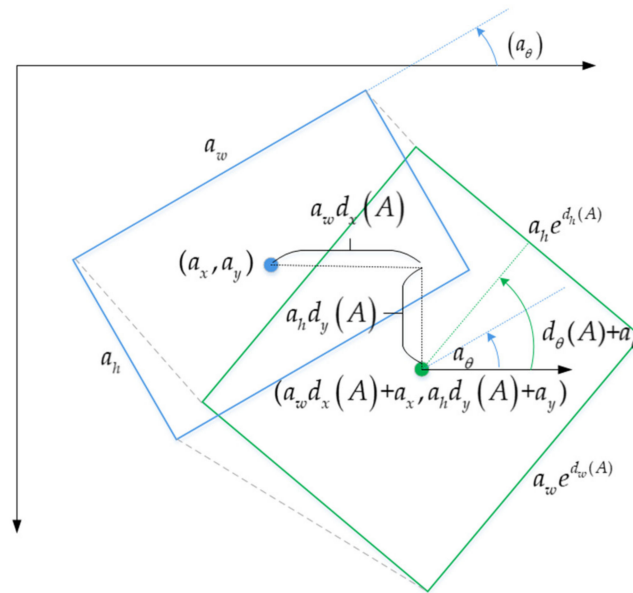
$$v = \frac{4}{\pi^2} * \left( \arctan \frac{w^{gt}}{h^{gt}} - \arctan \frac{w}{h} \right)^2 \quad (8)$$

Here,  $IoU$  is the intersection over union between the predicted box and the ground truth.  $\rho^2(b, b^{gt})$  is the Euclidean distance, and  $c$  is the diagonal length of the smallest enclosing box covering the bounding boxes.  $(w, h)$  and  $(w^{gt}, h^{gt})$  represent the height and width of the predicted box and the ground truth, respectively.

We add an angular loss branch based on the MSE loss function to design the loss function of the inclined bounding box regression. Given a rotation anchor box  $A = (a_x, a_y, a_w, a_h, a_\theta)$  and its corresponding ground-truth box  $G = (g_x, g_y, g_w, g_h, g_\theta)$ , our goal is to learn a mapping  $f$  such that  $f(A) = P$  where  $P = (p_x, p_y, p_w, p_h, p_\theta)$  is the predicted bounding box and  $P \approx G$ . The definition of the mapping relationship between  $A$  and  $P$  is expressed as:

$$\begin{cases} p_x = a_w d_x(A) + a_x \\ p_y = a_h d_y(A) + a_y \\ p_w = a_w \exp(d_w(A)) \\ p_h = a_h \exp(d_h(A)) \\ p_\theta = d_\theta(A) + a_\theta \end{cases} \quad (9)$$

Here,  $d_x(A)$  and  $d_y(A)$  denote the scale-invariant transformation of the two centers between  $A$  and  $P$ .  $d_\theta(A)$  represents angle-invariant transformation and  $d_w(A)$ ,  $d_h(A)$  characterize the exponential scale transformation of width and height respectively. As shown in Figure 8.



**Figure 8.** Illustration of transformation between rotation anchor box and predicted bounding box. The blue rectangle represents the rotation anchor box, and the green rectangle represents the predicted bounding box.

The goal of inclined bounding box regression is to train a set of parameters  $W$  to make  $Y = WX$ . During the training process, the input  $X$  is the feature map of each anchor box instead of  $(a_x, a_y, a_w, a_h, a_\theta)$ , the feature map is represented by  $\phi(A)$ , and  $Y$  is calculated by the ground truth  $G$  and the rotation anchor region  $A$  to obtain the translation and zoom, expressed by  $t_*$  where  $*$  is one of  $(x, y, w, h, \theta)$  as:

$$\begin{cases} t_x = (g_x - a_x) / a_w \\ t_y = (g_y - a_y) / a_h \\ t_w = \log(g_w / a_w) \\ t_h = \log(g_h / a_h) \\ t_\theta = g_\theta - a_\theta \end{cases} \quad (10)$$

Through iterative training,  $W$  makes the  $W\phi(A) \approx t$  with our loss function acquired as:

$$L_{box} = \lambda_{coord} \sum_{i=0}^{s \times s} \sum_{j=0}^B I_{ij}^{obj} (2 - h_i \times w_i) \left( t_*^i - w_*^T \phi(A) \right)^2 \quad (11)$$

Here  $\lambda_{coord}$  refers to balancing parameters with the value set to 1.  $w_i$  and  $h_i$  in  $(2 - h_i \times w_i)$  are the width and height of the ground truth, respectively. The role of  $(2 - h_i \times w_i)$  is responsible for balancing the generated loss value when detecting large and small objects.

#### 4. Experiments

This section evaluates the proposed algorithm on standard benchmarks and compares it with several existing methods. Analysis and discussions regarding our algorithm are also presented in the details.

##### 4.1. Benchmark Datasets

We selected three datasets containing directional text: ICDAR2015 [31], MSRA-TD500 [38], and ICDAR2017-MLT [39] for experiments to evaluate the performance on various directional text. To further demonstrate the versatility of R-YOLO, we also conducted experiments on a popular horizontal text dataset, ICDAR2013 [30]. A brief description of all relevant datasets is given below.

ICDAR2015 [31]: The ICDAR2015 scene text dataset issues from Challenge 4 of the ICDAR2015 Robust Reading Competition. The dataset comprises 1000 training images and 500 testing images, which were captured using Google glasses with relatively low resolutions. The text instance annotations have four vertices, which form an irregular quadrilateral bounding box with orientation information.

MSRA-TD500 [38]: The MSRA-TD500 dataset contains 200 test images and 300 training images, which contain arbitrarily-oriented text in Chinese as well as English. The texts are labeled with inclined boxes made up by the upper left corner of the rectangle, the width and height, and the rotation angle at the sentence level. Some long straight text lines appear in the dataset.

ICDAR2013 [30]: The ICDAR2013 dataset contains 233 test images and 229 training images, which is the key scene text of the ICDAR Robust Reading Competition. The scene text is horizontal and labeled with a horizontal rectangle box, including the upper left vertex and the lower right vertex of the rectangle.

ICDAR2017-MLT [39]: The ICDAR2017-MLT is a large-scale multi-lingual text dataset, which contains 7200 images for training, 1800 images for validating, and 9000 images for testing. The dataset consists of natural scene images containing texts in nine languages with multiple orientations. Some languages are labeled at line-level such as Chinese, Korean, and Japanese, while others are labeled at word-level such as English, French, Arabic, and Bangla. The different text length distributions in different languages make the detection task much more challenging.

We use five strategies to expand the training data set and to improve the robustness of training weights: (1) the image is rotated by 90, 180, and 270 degrees; (2) the image is flipped up and down, left and right; (3) the image is randomly translated jittering; (4) the brightness, contrast, hue, saturation, and noise of an image is adjusted; and (5) the mosaic data enhancement method is adopted, which randomly crops a part of four images and then puts them together into a new image.

The classical evaluation protocols for text detection, word spotting, and end-to-end recognition all rely on precision ( $P$ ), recall ( $R$ ), and F-measure ( $F$ ). Precision represents the ratio of the number of correctly detected text regions to the total number of detected text regions. Recall represents the ratio of the number of correctly detected text regions to the

total number of text regions in the dataset. F-measure is a single measure of quality created by combining recall and precision. These evaluation protocols are expressed as:

$$\begin{cases} P = \frac{TP}{TP+FP} \\ R = \frac{TP}{TP+FN} \\ F = 2 \times \frac{P \times R}{P+R} \end{cases} \quad (12)$$

Here,  $TP$ ,  $FP$ , and  $FN$  are the numbers of hit boxes, incorrectly identified boxes, and missed boxes, respectively.

#### 4.2. Implementation Details

Our scene text detection network is initialized using a pre-trained CSPDarknet53 mode. The number of iterations depends on the size of  $L_{text}$  (the output value of the loss function). When  $L_{text} < 0.5$  for a period of time, the iteration will stop. The weight decay and momentum are set to  $5 \times 10^{-4}$  and 0.9 respectively and the mini-batch size is set to 4. Testing images are resized to  $512 \times 512$ . The threshold  $N_t$  of RDIOU-NMS is set as 0.4. The confidence threshold and RDIOU threshold are set to 0.6 and 0.5, respectively. All the experiments are conducted on a single NVIDIA GeForce RTX 3090 graphic card with 24 GB memory, which adopts a new generation of Ampere architecture design, and its computing performance is faster than Titan X and Titan Xp.

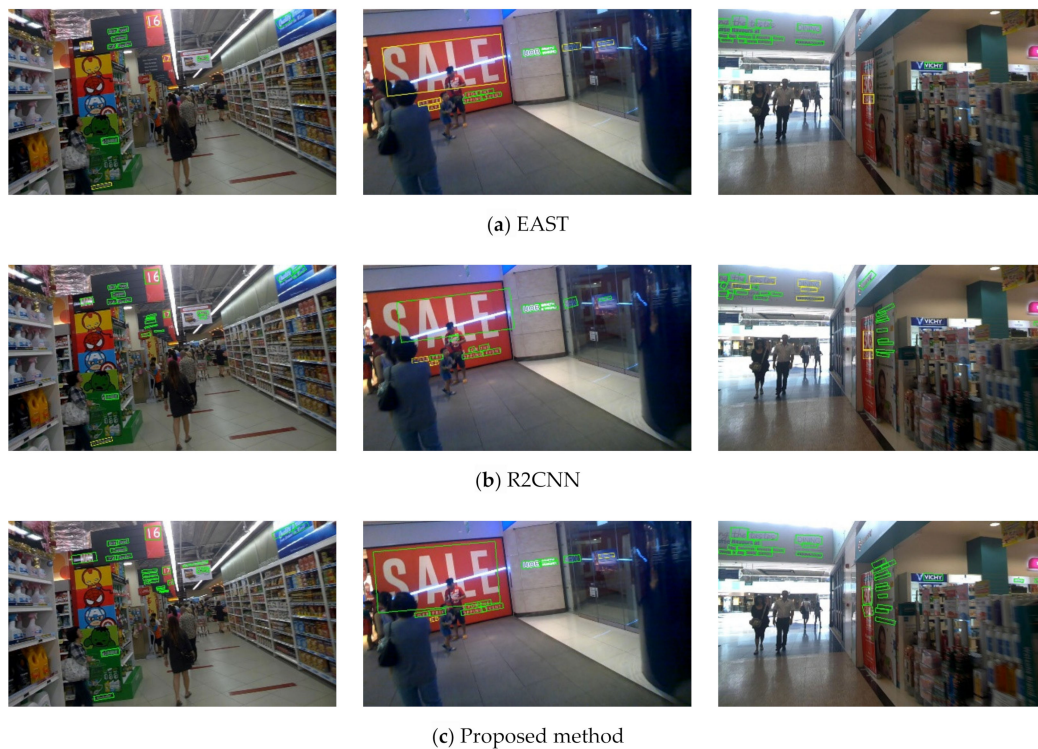
#### 4.3. Evaluation on Oriented Text Benchmark

We evaluate R-YOLO on the ICDAR2015 dataset. The model is fine-tuned for 50 k iterations on the training dataset of ICDAR2015. During the tuning stage, the learning rate starts from  $1.0 \times 10^{-3}$  and is multiplied by 1/10 after  $4.0 \times 10^4$  and  $4.5 \times 10^4$  iterations.

The quantitative results of the proposed method and other state-of-the-art methods are listed in Table 1. Our method achieves an F-measure of 82.3% and a computational speed of 62.5 fps. Compared with SegLink [18], He et al. [25], EAST [19], He et al. [40], DSRN [41], TextBoxes++ [24], and RRD [20], which are one-step methods, our F-measure is higher by 7.3%, 5.3%, 1.6%, 1.3%, 0.9%, 0.6%, and 0.1%, respectively. As regards speed, our proposed method is 3.72 times faster than the fastest method and 56.8 times faster than the slowest method listed in Table 1. This indicates that R-YOLO significantly outperforms other one-step-based methods in terms of detection efficiency and accuracy. Qualitative comparisons of text detection results are given in Figure 9.

**Table 1.** Results on the ICDAR2015 dataset. “R”, “P”, and “F” represent the recall, precision, and F-measure respectively. “OS” refers to the one-stage-based method. The FPS of the detection method running on Titan X or Titan Xp in the table is from the original paper.

Method	OS	R [%]	P [%]	F [%]	Device	FPS
DMPNet [42]		68.2	73.2	70.6	-	-
FTPN [43]		68.2	78.0	72.8	-	-
RRPN [23]		77.0	84.0	80.0	Titan X	4.70
R2CNN [22]		79.6	85.6	82.5	Tesla K80	2.20
SRPN+VGG <sub>Det</sub> [35]		79.7	92.0	85.4	Titan Xp	16.5
SRPN+SRPN <sub>Det</sub> [35]		74.8	85.2	79.6	Titan Xp	35.1
TextFuseNet [33]		89.7	94.7	92.1	Tesla V100	4.10
SegLink [18]	✓	76.8	73.1	75.0	-	-
He et al. [25]	✓	73.0	80.0	77.0	-	-
EAST [19]	✓	78.3	83.3	80.7	Titan X	16.8
He et al. [40]	✓	80.0	82.0	81.0	Titan X	1.10
DSRN [41]	✓	79.6	83.2	81.4	Titan X	8.80
TextBoxes++ [24]	✓	76.7	87.2	81.7	Titan Xp	11.6
RRD [20]	✓	79.0	85.6	82.2	Titan Xp	6.50
R-YOLO	✓	78.2	87.0	82.3	RTX 3090	62.5



**Figure 9.** Qualitative comparisons of text detection results on some ICDAR2015 incidental text images. Green bounding boxes: correct detections; yellow bounding boxes: missed ground truths.

We also beat the FTPN [43], RRPN [23], and SRPN+SRPN<sub>Det</sub> [35] methods based on the two-step strategy; our method's F-measure is higher by 9.5%, 2.3%, and 2.7%, respectively. Compared with TextFuseNet [33], which is an instance segmentation-based method, the detection accuracy of the proposed method is 9.8% lower. TextFuseNet [33] utilizes three branches to obtain three levels of features and adopts multi-path fusion architecture to obtain fused features for text detection, which consumes a large amount of computational resources, resulting in the detection speed is only 1/15 of ours. Compared with R2CNN [22] and SRPN+VGG<sub>Det</sub> [35], we lose 0.2% and 3.1% accuracy. From the test results, it is obvious that there is still a gap between our approach and some two-stage-based methods in terms of detection accuracy. However, our proposed method has significant advantages in terms of detection speed. Overall, the R-YOLO method achieves comparable performance with most two-stage methods, while maintaining real-time detection speed.

#### 4.4. Evaluation on Long Text Benchmark

To further test the ability of our proposed method to detect long texts, we perform fine-tuning experiments on the MSRA-TD500 dataset and stop after about 45 k iterations. During the tuning stage, training images are resized to  $512 \times 512$ . The learning rate starts from  $1.0 \times 10^{-3}$  and is multiplied by 1/10 after  $3.5 \times 10^4$  and  $4.0 \times 10^4$  iterations.

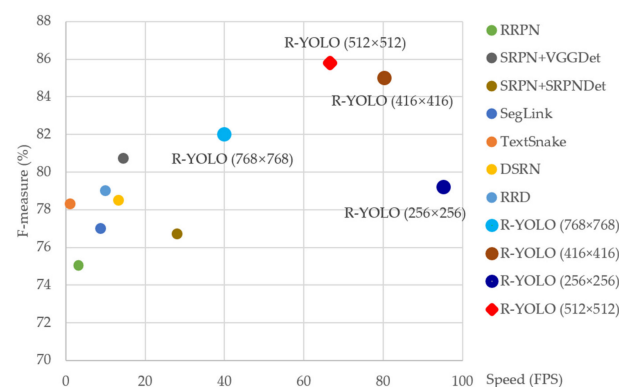
As summarized in Table 2, testing images of four different sizes are evaluated. R-YOLO ( $256 \times 256$ ) achieves an F-measure of 79.2%, while the detection speed is 95.2 fps. R-YOLO ( $512 \times 512$ ) achieves a precision, recall, and F-measure of 90.2%, 81.9%, and 85.8% respectively, while the detection speed is 66.6 fps. From the test results, we find that the detection speed is related to the resolution of the test image. As the resolution of the test image decreases, the detection speed increases. The detection accuracy is related to the size of the training image. When the size of the test image is closer to the training image, the detection accuracy is higher. Therefore, we can resample the test image to the scale of the training image to increase the detection accuracy. Compared with SRPN+VGG<sub>Det</sub> [35], which is the state-of-the-art method in terms of detection accuracy, the F-measure of R-YOLO ( $512 \times 512$ ) is 5.1% higher. It is also 4.5 times faster. The results



show that our proposed method achieves a performance that is comparable to that of state-of-the-art methods, which means it can also process multi-oriented long texts satisfactorily. Figure 10 shows comparisons of several recent scene text detection methods. Some qualitative results are visualized in Figure 11.

**Table 2.** Quantitative results of different methods are evaluated on the MSRA-TD500 dataset. “R”, “P”, and “F” represent the recall, precision, and F-measure respectively. “OS” refers to the one-stage-based method. R-YOLO (512 × 512) indicates that testing images are resized to 512 × 512.

Method	OS	R [%]	P [%]	F [%]	FPS
RRPN [23]		69.0	82.0	75.0	3.3
SRPN+SRPN <sub>Det</sub> [35]		70.8	83.6	76.7	28.1
SRPN+VGG <sub>Det</sub> [35]		77.0	84.9	80.7	14.6
He et al. [40]	✓	70.0	77.0	74.0	-
EAST [19]	✓	67.4	87.3	76.1	-
SegLink [18]	✓	70.0	86.0	77.0	8.9
TextSnake [16]		73.9	83.2	78.3	1.1
DSRN [41]	✓	71.2	87.6	78.5	13.3
RRD [20]	✓	73.0	87.0	79.0	10.0
R-YOLO (768 × 768)	✓	76.5	88.3	82.0	40.0
R-YOLO (416 × 416)	✓	79.9	90.9	85.0	80.3
R-YOLO (256 × 256)	✓	71.6	88.6	79.2	95.2
R-YOLO (512 × 512)	✓	81.9	90.2	85.8	66.6



**Figure 10.** Comparisons of several recent scene text detection methods on the MSRA-TD500 dataset in terms of accuracy and speed. R-YOLO (512 × 512) (in red) achieves the ideal tradeoff between effectiveness and efficiency. Detailed results are listed in Table 2.



**Figure 11.** Results on MSRA-TD500.

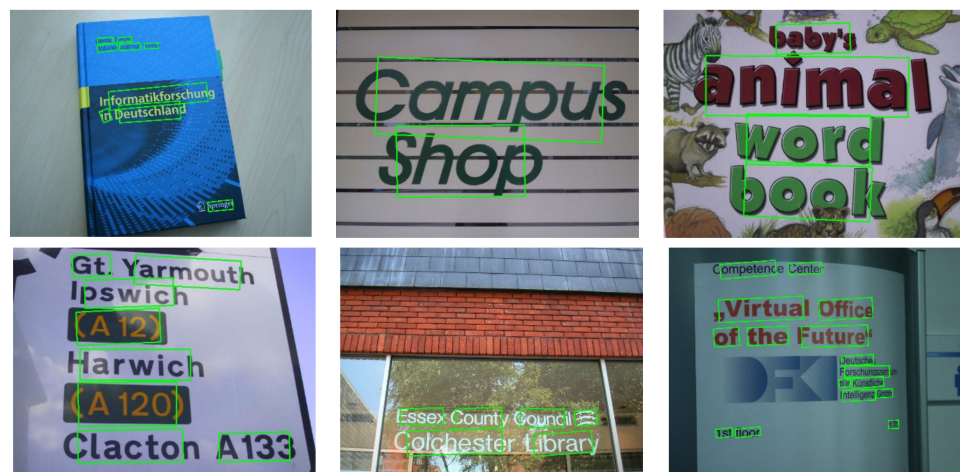
#### 4.5. Evaluation on Horizontal Text Benchmark

We also conducted experiments on ICDAR2013 [30] to test the general adaptability of our method. This dataset contains 233 focused scene text images where the text in the images is horizontal. During the tuning stage, the model is fine-tuned for 25 k iterations. The learning rate starts from  $1.0 \times 10^{-3}$  and is multiplied by 1/10 after  $1.5 \times 10^4$  and  $2.0 \times 10^4$  iterations.

Table 3 compares the results of YOLOv4 and the proposed method. The recall rate is improved from 71.5% to 82.9%, and the F-measure is improved from 80.1% to 86.4%, while the speed reduces by 0.2 fps only. R-YOLO achieves at least 1.3% improvement over other methods except for SRPN+VGG<sub>Det</sub> [35] and TextFuseNet [33] on this dataset. However, TextFuseNet [33] performs text detection by fusing three levels of features and only processes four images per second. This is not viable for real-time detection. Some detection results obtained on the benchmarks are illustrated in Figure 12, which show that our method can suitably handle horizontal text detection in natural images.

**Table 3.** Detection results on ICDAR2013. “R”, “P”, and “F” represent the recall, precision, and F-measure respectively. “OS” refers to the one-stage-based method.

Method	OS	R [%]	P [%]	F [%]	FPS
Faster R-CNN [34]		71.0	75.0	73.0	-
RRPN [23]		72.0	90.0	80.0	-
SRPN+VGG <sub>Det</sub> [35]		84.2	92.5	88.2	20.9
SRPN+SRPN <sub>Det</sub> [35]		83.3	86.4	84.8	30.5
TextFuseNet [33]		92.3	96.5	94.3	4.00
SSD [37]	✓	60.0	80.0	68.0	-
TextBoxes++ [24]	✓	74.0	86.0	80.0	-
YOLOv4 [26]	✓	71.5	91.0	80.1	47.2
R-YOLO	✓	82.9	90.1	86.4	47.0



**Figure 12.** Example results of R-YOLO on ICDAR2013.

#### 4.6. Evaluation on Multi-Lingual Text Benchmark

As shown in Table 4, we conduct an experiment to test the effectiveness of the fourth detection branch we added. Compared with the R-YOLO-3 method, R-YOLO-4 achieves better performance with the four-scale detection branch. The recall rate is improved from 69.5% to 71.7%, the F-measure increases from 72.9% to 74.3%, and the speed is reduced by 3.6 fps. These experimental results indicate that the fourth detection branch can effectively enhance the detection accuracy. Shallow features have higher resolution and contain more location and detailed information, which is an effective way to solve the problem of small-text detection. However, as the depth of the network increases, it is easy to lose shallow

features. The fourth detection branch can effectively extract shallow features and fuse them with deep features. The fused features have rich detailed information as well as semantic information of deep features, which can effectively improve the detection ability of small-scale text.

**Table 4.** Detection results on ICDAR2017-MLT. “R”, “P”, and “F” represent the recall, precision, and F-measure respectively. OS stands for one-stage-based method. R-YOLO-3 stands for R-YOLO with three detection branches. R-YOLO-4 stands for R-YOLO with four detection branches. R-YOLO-RIoU refers to using the traditional Riou-NMS algorithm to remove redundant inclined bounding boxes.

Method	RN	OS	R [%]	P [%]	F [%]	FPS
FOTS [44]			81.8	62.3	70.8	23.9
Lyu et al. [45]			74.3	70.6	72.4	-
LOMO [21]			67.2	80.2	73.1	-
CRAFT [46]			68.2	80.6	73.9	8.60
GNNets [47]			70.1	79.6	74.5	-
DB-ResNet-18 [48]			63.8	81.9	71.7	41.0
DB-ResNet-50 [48]			67.9	83.1	74.7	19.0
R-YOLO-RIoU		✓	66.3	78.0	71.7	67.5
R-YOLO-3	✓	✓	69.5	76.7	72.9	71.2
R-YOLO-4	✓	✓	71.7	77.1	74.3	67.6

In order to verify the effectiveness of the RDioU-NMS algorithm, we conducted a comparative experiment between the RDioU-NMS algorithm and the Riou-NMS algorithm under the same conditions. Compared with the R-YOLO-RIoU method, R-YOLO-4 reduces the losses of F-measure and Recall by 2.6% and 5.4%. This demonstrates that the RDioU-NMS algorithm can effectively reduce the missed detection rate.

Compare with the previous methods, R-YOLO-4 achieves state-of-the-art results in terms of speed, surpassing the second-fastest DB-ResNet-50 [48] method at 26.6 fps. For the accuracy, R-YOLO-4 surpasses the FOTS [44], DB-ResNet-18 [48], Lyu et al. [45], LOMO [21], and CRAFT [46] methods by 3.5%, 2.6%, 1.9%, 1.2%, and 0.4%, respectively. This indicates that our method is competitive in multi-language text detection. Figure 13 demonstrates some detection results of R-YOLO-4 on ICDAR2017-MLT.



**Figure 13.** Some visualization results on the ICDAR2017-MLT benchmarks.



#### 4.7. Analysis and Discussion

R-YOLO can achieve higher speeds than *state-of-the-art* methods because our network has two advantages. First, the proposed method adopts CSPDarknet53 as the backbone network. Compared with the existing detection methods listed in Table 1 using VGGNet or ResNet as the backbone, a large number of  $1 \times 1$  convolution kernels are exploited in the CSPDarknet53 network to reduce the dimensions of the feature maps, which reduces the number of parameters and the size of the model considerably. Second, compared with the methods based on the two-stage strategy, our proposed method based on the one-stage strategy regresses the bounding box directly from the convolutional feature maps without relying on the region proposal, thus saving time required to calculate the region proposal.

Some qualitative comparisons are illustrated in Figure 9. From the detection results in the figure, we can observe that EAST, as well as R2CNN, missed a part of the text area and our method has achieved satisfactory detection performance. There are several reasons for this: first, EAST relies on an accurate segmentation score map as the score of the bounding boxes. However, the text region segmentation is challenging in complex environments. If the score map is not accurate enough, it is difficult to achieve accurate results. Our proposed method does not suffer from such limitations. It relies on anchor boxes and regresses the bounding boxes directly from the convolutional feature maps, where rich information is reserved as compared to the score map. Second, compared with the traditional inclined NMS algorithm used by R2CNN, a distance penalty is adopted in the proposed RDIOU-NMS algorithm to address the problem of false suppression caused by overlapping bounding boxes of different texts. In particular, in the dense text area, the effect is more satisfactory. Third, a variety of data enhancement approaches are widely adopted in our training set to improve the robustness of training weights. Fourth, in our network, effective tricks are utilized to improve the ability of network feature extraction such as SSP, PANet, and SAM, and their effectiveness has been verified in YOLOv4. The combined effect of the above-mentioned reasons makes R-YOLO more robust than competing methods in detecting arbitrarily-oriented text.

#### 4.8. Limitations of the Proposed Algorithm

The proposed method outperforms the existing methods significantly in terms of detection efficiency while maintaining high accuracy, but has limitations in small-size natural scene text detection, although the addition of detection branches is helpful to improve the detection accuracy. This is a common limitation for YOLO-based object detectors. Another limitation is that the method is not good at detecting curved text.

### 5. Conclusions

In this paper, a series of improvements based on YOLOv4 are proposed to enable text detection in natural scenes where the text could be arbitrarily-oriented and of varied scales. To improve the performance of detecting small-size natural scene texts, we have added a detection branch. In order to select the optimal slanted bounding box, we proposed RDIOU-NMS, which not only considers the angle factor of the inclined bounding box but also the center-point distance between the two boxes. In addition, the representations of anchor box, bounding box regression algorithm, and loss function are improved to adapt to the detection of arbitrarily rotated text. Experimental comparisons and model analyses were conducted on the ICDAR2015, MSRA-TD500, ICDAR2013, and ICDAR2017-MLT datasets. On the ICDAR2015 dataset, our method achieved an F-measure of 82.3 at 62.5 fps with 720 p resolution. The results show that our proposed method can achieve an advanced level of text detection with very high computational efficiency. However, it still has room for improvement in terms of detection accuracy. First, the network backbone may be improved by using an advanced attention mechanism. Second, the improved loss function of inclined bounding boxes based on Pixels-IOU loss could be considered. Our future research will focus on these areas.

**Author Contributions:** Conceptualization, methodology and investigation, X.W., S.Z., and R.L.; software and validation, X.W.; writing—original draft preparation, X.W., and S.Z.; writing—review and editing, C.Z., R.L. and L.G.; supervision, S.Z.; funding acquisition, S.Z. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the National Natural Science Foundation of China (Project Number: 41671452 and 41701532) and the China Postdoctoral Science Foundation-funded project (2017M612510).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Neumann, L.; Matas, J. Scene text localization and recognition with oriented stroke detection. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Sydney, Australia, 3–6 December 2013; pp. 97–104.
2. Pan, Y.; Hou, X.; Liu, C. A hybrid approach to detect and localize texts in natural scene images. *IEEE Trans. Image Process.* **2011**, *20*, 800–813. [\[PubMed\]](#)
3. Yin, X.; Yin, X.; Huang, K.; Hao, H. Robust text detection in natural scene images. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *36*, 970–983. [\[PubMed\]](#)
4. Epshtein, B.; Ofek, E.; Wexler, Y. Detecting text in natural scenes with stroke width transform. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), San Francisco, CA, USA, 13–18 June 2010; pp. 2963–2970.
5. Huang, W.; Qiao, Y.; Tang, X. Robust scene text detection with convolution neural network induced MSER trees. In Proceedings of the European Conference on Computer Vision (ECCV), Zurich, Switzerland, 6–12 September 2014; pp. 497–511.
6. Huang, W.; Lin, Z.; Yang, J.; Wang, J. Text localization in natural images using stroke feature transform and text covariance descriptors. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Portland, OR, USA, 23–28 June 2013; pp. 97–104.
7. Zhong, Z.; Jin, L.; Huang, S.; Feng, Z. DeepText: A new approach for text proposal generation and text detection in natural images. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, USA, 5–9 March 2017; pp. 1208–1212.
8. Liao, M.; Shi, B.; Bai, X.; Wang, X.; Liu, W. TextBoxes: A fast text detector with a single deep neural network. *arXiv* **2016**, arXiv:1611.06779.
9. Tian, Z.; Huang, W.; He, T.; He, P.; Qiao, Y. Detecting text in natural image with connectionist text proposal network. *arXiv* **2016**, arXiv:1609.03605v1.
10. Tian, S.; Pan, Y.; Huang, C.; Lu, S.; Yu, K.; Tan, C. Text flow: A unified text detection system in natural scene images. *arXiv* **2016**, arXiv:1604.06877v1.
11. Zhang, Z.; Shen, W.; Yao, C.; Bai, X. Symmetry-based text line detection in natural scenes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 2558–2567.
12. Gupta, A.; Vedaldi, A.; Zisserman, A. Synthetic data for text localisation in natural images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 2315–2324.
13. Sun, L.; Huo, Q.; Jia, W. A robust approach for text detection from natural scene images. *Pattern Recognit.* **2015**, *48*, 2906–2920. [\[CrossRef\]](#)
14. Deng, D.; Liu, H.; Li, X.; Cai, D. Pixellink: Detecting scene text via instance segmentation. In Proceedings of the 32nd AAAI Conference Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; pp. 6773–6780.
15. Zhang, Z.; Zhang, C.; Shen, W.; Yao, C.; Liu, W.; Bai, X. Multi-oriented text detection with fully convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 4159–4167.
16. Long, S.; Ruan, J.; Zhang, W.; He, X.; Wu, W.; Yao, C. Textsnake: A flexible representation for detecting text of arbitrary shapes. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 20–36.
17. Yang, Q.; Cheng, M.; Zhou, W.; Chen, Y.; Qiu, M.; Lin, W. Inceptext: A new inception-text module with deformable PSROI pooling for multi-oriented scene text detection. *arXiv* **2018**, arXiv:1805.01167v2.
18. Shi, B.; Bai, X.; Belongie, S. Detecting oriented text in natural images by linking segments. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2550–2558.
19. Zhou, X.; Yao, C.; Wen, H.; Wang, Y.; Zhou, S.; He, W.; Liang, J. EAST: An efficient and accurate scene text detector. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 5551–5560.
20. Liao, M.; Zhu, Z.; Shi, B.; Xia, G.; Bai, X. Rotation-sensitive regression for oriented scene text detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 5909–5918.



21. Zhang, C.; Liang, B.; Huang, Z.; En, M.; Han, J.; Ding, E.; Ding, X. Look more than once: An accurate detector for text of arbitrary shapes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 10552–10561.
22. Jiang, Y.; Zhu, X.; Wang, X.; Yang, S.; Li, W.; Wang, H.; Fu, P.; Luo, Z. R2CNN: Rotational region CNN for arbitrarily-oriented scene text detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 3610–3615.
23. Ma, J.; Shao, W.; Ye, H.; Wang, L.; Wang, H.; Zheng, Y.; Xue, X. Arbitrary-oriented scene text detection via rotation proposals. *IEEE Trans. Multimed.* **2018**, *20*, 3111–3122. [\[CrossRef\]](#)
24. Liao, M.; Shi, B.; Bai, X. TextBoxes++: A single-shot oriented scene text detector. *IEEE Trans. Image Process.* **2018**, *27*, 3676–3690. [\[CrossRef\]](#) [\[PubMed\]](#)
25. He, P.; Huang, W.; He, T.; Zhu, Q.; Qiao, Y.; Li, X. Single shot text detector with regional attention. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 3047–3055.
26. Bochkovskiy, A.; Wang, C. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934v1.
27. Lin, H.; Yang, P.; Zhang, F. Review of scene text detection and recognition. *Arch. Comput. Methods Eng.* **2019**, *27*, 433–454. [\[CrossRef\]](#)
28. Brisinello, M.; Grbić, R.; Vranješ, M.; Vranješ, D. Review on text detection methods on scene images. In Proceedings of the 2019 International Symposium ELMAR, Zadar, Croatia, 23–25 September 2019; pp. 51–56.
29. Raisi, Z.; Nael, M.A.; Fieguth, P.; Wardell, S.; Zelek, J. Text detection and recognition in the wild: A review. *arXiv* **2020**, arXiv:2006.04305v2.
30. Karatzas, D.; Shafait, F.; Uchida, S.; Iwamuram, M.; Bigorda, L.; Mestre, S.; Mas, J.; Mota, D.F. ICDAR 2013 robust reading competition. In Proceedings of the 12th International Conference on Document Analysis and Recognition (ICDAR), Washington, DC, USA, 25–28 August 2013; pp. 1484–1493.
31. Karatzas, D.; Gomez-Bigorda, L.; Nicolaou, A.; Ghosh, S.; Bagdanov, A.; Iwamura, M.; Matas, J.; Neumann, L.; Chandrasekhar, V.R.; Lu, S.; et al. ICDAR2015 competition on robust reading. In Proceedings of the 12th International Conference on Document Analysis and Recognition (ICDAR), Oradea, Romania, 11–12 June 2015; pp. 1156–1160.
32. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the 18th International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI), Munich, Germany, 5–9 October 2015; pp. 234–241.
33. Ye, J.; Chen, Z.; Liu, J.; Du, B. TextFuseNet: Scene Text Detection with Richer Fused Features. In Proceedings of the 29th International Joint Conference on Artificial Intelligence (IJCAI-20), Yokohama, Japan, 7–15 January 2021.
34. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In Proceedings of the Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, Montreal, QC, Canada, 7–12 December 2015; pp. 91–99.
35. He, W.; Zhang, X.; Yin, F.; Luo, Z.; Ogier, J.; Liu, C. Realtime multi-scale scene text detection with scale-based region proposal network. *Pattern Recognit.* **2020**, *98*, 107026. [\[CrossRef\]](#)
36. Redmon, J.; Farhadi, A. YOLOv3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
37. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single shot multibox detector. In Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 8–16 October 2016; pp. 21–37.
38. Yao, C.; Bai, X.; Liu, W.; Ma, Y.; Tu, Z. Detecting texts of arbitrary orientations in natural images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, 16–21 June 2012; pp. 1083–1090.
39. Nayeef, N.; Yin, F.; Bizid, I.; Choi, H.; Feng, Y.; Karatzas, D.; Luo, Z.; Pal, U.; Rigaud, C.; Chazalon, J. ICDAR2017 robust reading challenge on multi-lingual scene text detection and script identification-rrc-mlt. In Proceedings of the 12th International Conference on Document Analysis and Recognition (ICDAR), Kyoto, Japan, 13–15 November 2017; pp. 1454–1459.
40. He, W.; Zhang, X.; Yin, F.; Liu, C. Deep direct regression for multi-oriented scene text detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 745–753.
41. Wang, Y.; Xie, H.; Fu, Z.; Zhang, Y. DSRN: A deep scale relationship network for scene text detection. In Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI), Macao, China, 10–16 August 2019; pp. 947–953.
42. Liu, Y.; Jin, L. Deep matching prior network: Toward tighter multi-oriented text detection. *arXiv* **2018**, arXiv:1703.01425v1.
43. Liu, F.; Chen, C.; Gu, D.; Zheng, J. FTPN: Scene text detection with feature pyramid based text proposal network. *IEEE Access* **2019**, *7*, 44219–44228. [\[CrossRef\]](#)
44. Liu, X.; Liang, D.; Yan, S.; Chen, D.; Qiao, Y.; Yan, J. FOTS: Fast oriented text spotting with a unified network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, Utah, USA, 18–22 June 2018; pp. 5676–5685.
45. Lyu, P.; Yao, C.; Wu, W.; Yan, S.; Bai, X. Multi-oriented Scene Text Detection via Corner Localization and Region Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 7553–7563.
46. Baek, Y.; Lee, B.; Han, D.; Yun, S.; Lee, H. Character region awareness for text detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 9365–9374.

- 
47. Xu, Y.; Duan, J.; Kuang, Z.; Yue, X.; Sun, H.; Guan, Y.; Zhang, W. Geometry Normalization Networks for Accurate Scene Text Detection. *arXiv* **2019**, arXiv:1909.00794v1.
  48. Liao, M.; Wan, Z.; Yao, C.; Chen, K.; Bai, X. Xiang Bai Real-time Scene Text Detection with Differentiable Binarization Dec. *arXiv* **2019**, arXiv:1911.08947v2.