# Transformer-based Text Detection in the Wild

Zobeir Raisi[1], Mohamed A. Naiel[1], Georges Younes[1], Steven Wardell[2] and John S. Zelek[1]
[1]University of Waterloo, Waterloo, ON, Canada, N2L 3G1
[2]ATS Automation Tooling Systems Inc., Cambridge, ON, Canada, N3H 4R7
{zraisi, mohamed.naiel, gyounes, jzelek}@uwaterloo.ca, swardell@atsautomation.com

## Abstract

*A major limitation to most state-of-the-art visual localization methods is their ineptitude to make use of ubiquitous signs and directions that are typically intuitive to humans. Localization methods can greatly benefit from a system capable of reasoning about a variety of cues beyond low-level features, such as street signs, store names, building directories, room numbers, etc.*

*In this work, we tackle the problem of text detection in the wild, an essential step towards achieving text-based localization and mapping. While current state-of-the-art text detection methods employ ad-hoc solutions with complex multi-stage components to solve the problem, we propose a Transformer-based architecture inherently capable of dealing with multi-oriented texts in images. A central contribution to our work is the introduction of a loss function tailored to the rotated text detection problem that leverages a rotated version of a generalized intersection over union score to properly capture the rotated text regions.*

*We evaluate our proposed model qualitatively and quantitatively on several challenging datasets namely, ICDAR15, ICDAR17, and MSRA-TD500, and show that it outperforms current state-of-the-art methods in text detection in the wild.*

## 1. Introduction

Visual localization has played an essential role in recent advancements of several technologies such as augmented reality, self-driving cars and autonomous robotic navigation. However, most localization methods rely on low-level information (corners, edges, *etc.*) that does not necessarily correlate to topologically meaningful map representations [1]. Humans on the other hand, are capable of navigating an unexplored environment by simply following directions from signs and texts that low level features cannot capture. This is where text detection in the wild can play an important role as it enables image-based localization methods to reason about the ubiquitous navigation labels surrounding them to navigate through unexplored environments.

However, texts can have several fonts, different colors, can appear on various surfaces, in different locations in the image, and with a wide range of orientations and scales [2, 3]. For example, they can appear anywhere from building names, store fronts, street signs, to shopping mall signs, *etc.* Therefore, reliable and consistent text detection is of utmost importance.

At its core, text detection is the process of localizing a word or a sentence in a given image. To that end, several recent scene text detection methods [4–10] have utilized deep convolutional neural networks (DCNNs) as feature extractors [11–14], and solved for text detection by casting it as an object detection problem. Despite achieving promising results on various challenging datasets [15–17], their performance is still lacking in several key challenging scenarios, including and not limited to in-plane-rotations, multi-oriented and multi-resolution text, complex fonts, special characters, perspective distortion, occlusions, shadows, illumination artifacts, and image blurriness [3, 18]. We attribute these shortcomings to the ad-hoc multi-layered approaches most of these methods have deployed in an attempt to model the wide range of variation texts can have in the wild.

As such, we propose to account for these variations within our model, by leveraging the power of Transformer [19], which is a recent deep learning architecture that learns how to encode and decode data by looking not only backward but also forward to extract relevant information from a whole sequence. This new approach allows models to solve for complex tasks, such as machine translation [19], speech recognition [20], and recently, object detection [21, 22] and scene text recognition [23, 24].

To the best of our knowledge, this is the first time a Transformer is introduced for scene text detection. Unlike the baseline Transformer-based method in [21, 22] that only generates rectangular bounding boxes for detected objects, and therefore, it is not designed for handling arbitrary shape detection; we propose a new architecture that is able to detect multi-oriented text. Our contributions are as follows:

1. We improve the detection performance by using the Transformer [21] architecture, and by leveraging a differentiable loss function that accepts text instances' arbitrary shapes.

2. We propose using a rotated text representation that can better represent multi-oriented text regions.

3. We validate the performance of the proposed method by conducting several quantitative and qualitative experiments on challenging scenarios, and show that the proposed method outperforms the state-of-the-art on three public benchmark datasets, namely, ICDAR15 [16], ICDAR17 [17], and MSRA-TD500 [15].

## 2. Related Work

Text detection methods can be broadly categorized into two main groups:

1. First, *segmentation-based* methods [7, 8, 25–27] mainly use Mask-RCNN [14] as a backbone to produce a segmentation mask. They also consist of additional segmentation heads alongside the detection bounding box. Although these methods [7, 8, 25–27] offer high-precision detection when text is horizontal, they usually require multiple post-processing steps to infer the produced segmentation mask and predict precisely oriented bounding boxes [10]. Furthermore, their complicated architectures usually require high inference time due to the refinement of region proposal and label generation for arbitrary oriented text prediction.

2. On the other hand, *region-based* methods [4, 5, 9, 10, 18, 28–30] often predict candidate bounding box directly for the target region of interest. Unlike segmentation-based methods, region-based methods are more straight-forward and efficient for predicting the target region. However, applying the standard object detection frameworks directly for detecting arbitrarily-oriented text may cause redundant background noise, and unnecessary overlap [9]. Thus, for more accurate detection, many methods adopted rotated bounding boxes approach to better represent oriented text as in [4–6, 9, 10].

Particularly, EAST [4] presented a fast text detector that makes dense predictions which are then processed using locality-aware non-maximum suppression (NMS) to detect multi-oriented texts in an image. Later, TextBoxes++ [5] improved the rectangular detection architecture by using a long convolution kernel, increasing the number of region proposals, and replacing the rectangle bounding boxes of text with rotated boxes in order to detect arbitrarily-oriented text. In [9], Deng *et al*. introduced a mechanism called STELA for learning anchors and making the two-stage framework of Faster-RCNN into a one-stage detector to make the final oriented text detection more efficient. Recently, Wang *et al*. proposed RYOLO [10] that incorporated angle information of rotated boxes and feature maps of different scales to extend the standard YOLO framework for detecting rotated text. Although some of the mentioned methods [9, 10] achieved state-of-the-art performance on several benchmark datasets, they require a complicated architecture with multiple stages of post-processing like NMS and rotating anchor design.

## 3. Methodology

Our main goal is to address the challenges of multi-oriented scene text detection by proposing a modified Transformer-based architecture [21]. Transformers [19] are attention-based deep-learning architectures that can scan through each element of a sequence using a self-attention module, and provide an update by aggregating information from the whole sequence. When compared to previous deep-learning approaches, Transformers can better capture the global dependencies among the input and output sequences with the help of an attention mechanism [31]. During training, the encoder's multi-head self-attention layer learns how to separate individual words in the scene image by performing the global computations, whereas the decoder learns how to attend to different characters in words by using different learnable vectors (also referred to as object queries). This is a very important feature since when properly trained, the last layer of the decoder is capable of directly predicting the targets' location without the need for multiple post-processing steps, as mentioned in Section 2, which are typically required by other architectures [4, 5, 7–10, 32].

### 3.1. Architecture

The overall architecture of the proposed text detection scheme is shown in Figure 1. During the *encoding phase*, the $i^{th}$ training color image $\mathcal{I}_i \in \mathbb{R}^{H_0 \times W_0 \times 3}$ is first processed to extract its features. While there are several ways of extracting features from an image such as RCNN [33], YOLO [13] *etc.*, we choose a ResNet [34] as CNN backbone because of its parameter efficiency and its ability in handling the vanishing gradient problem. The CNN produces a corresponding lower resolution feature map $F_i \in \mathbb{R}^{H \times W \times c}$, where $c$ indicates the number of channels, $H = H_0/\eta$, and $W = W_0/\eta$ with $\eta$ being the downsampling factor. In order to reduce the computational cost of the encoding stage, the number of channels within the feature map $F_i$, are reduced using a $1 \times 1$ convolutional layer, resulting in $F_i' \in \mathbb{R}^{H \times W \times d}$, where $d < c$.

As in [19], we also make use of 2D positional encoding maps $P \in \mathbb{R}^{H \times W \times d}$, which are added to $F_i'$ such that

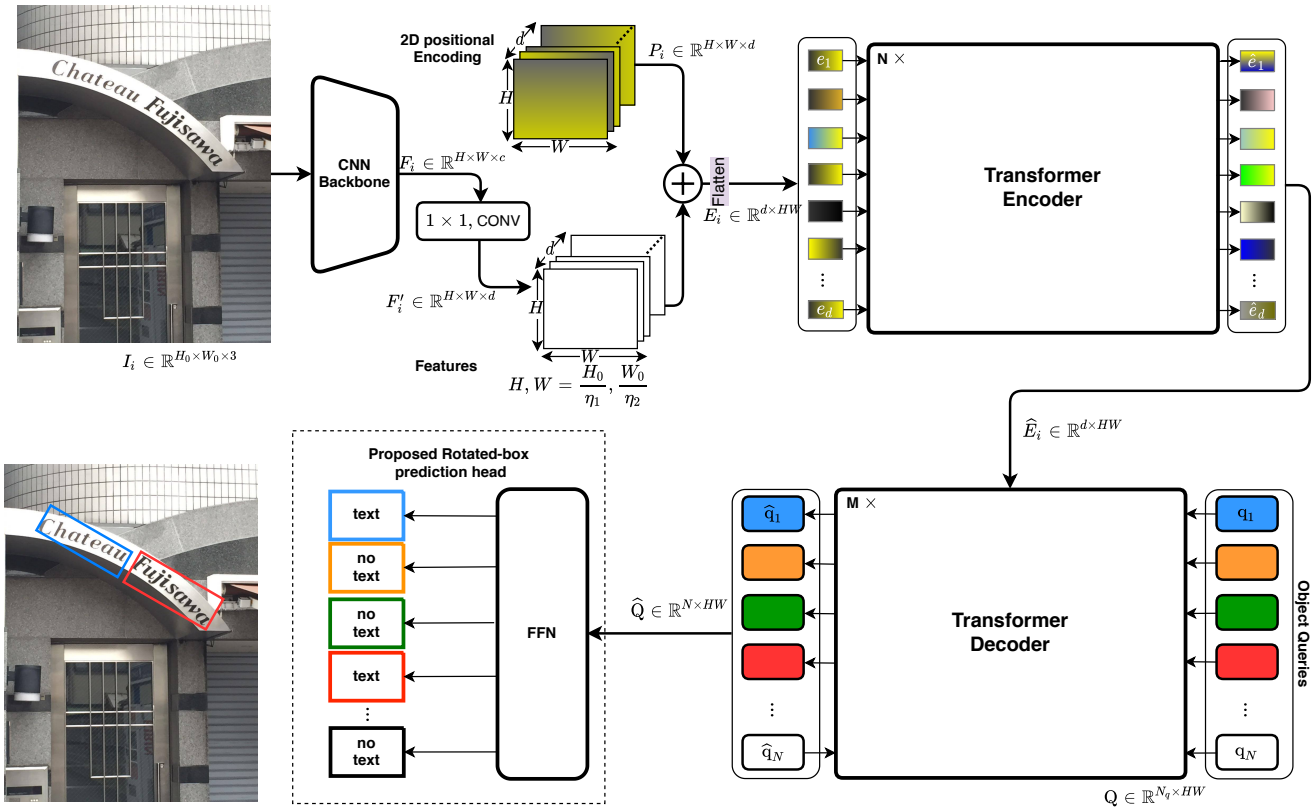



Figure 1. Block diagram of the proposed text-detection scheme using Transformer. Unlike the framework in [21], the proposed framework aims to represent text regions utilizing quadrilateral-based predictions instead of the classical rectangular-based predictions used in [21].

$F_i'' = F_i' + P$. The positional encoded map $F_i''$, allows the multi-head self-attention layer to better capture the 2D spatial information. Since, the encoder in the Transformer only accepts a set of vectors as input, the $d$ channels of $F_i''$ are vectorized and stacked to form one feature matrix $E_i$ of the form:

$$E_i = \begin{bmatrix} e_{i,1} \\ e_{i,2} \\ \vdots \\ e_{i,d} \end{bmatrix} \in \mathbb{R}^{d \times HW}, \tag{1}$$

with the vector $e_{i,j} = \text{Mat2Vec}(F_i''(:,:,j)) \in 1 \times HW$, and Mat2Vec is a matrix to vector converter.

The standard encoder of Transformer with $N = 6$ layers [19] is then used to generate the $i^{th}$ encoded feature matrix $\hat{E}_i \in \mathbb{R}^{d \times HW}$. This encoder also includes a multi-head self-attention and FFN layers. The multi-head self-attention mechanism in Transformer's encoder allows the model to handle the scale differences in text instances [22].

In the *decoding phase*, as in [21] the encoded feature matrix $\hat{E}_i$, along with a fixed set of learnable embeddings, called object queries $Q \in \mathbb{R}^{N_q \times HW}$, are passed through a Transformer decoder of $M = 6$ layers, where $N_q$ denotes the maximum number of text instance queries that can appear in each input image, and $Q = [q_1^\top, \ldots, q_{N_q}^\top]^\top$ such that the $k^{th}$ vector $q_k$ is of size $1 \times HW$. The decoded set of feature vectors $\hat{Q} \in \mathbb{R}^{N_q \times HW}$ is then fed into the FFN layers, which consists of a three layer perceptron with a ReLU activation function plus a $d$-dimensional hidden layer, and a linear projection layer to predict the bounding box and class label for each query. Finally, a bipartite matching [35] is used at the end to predict the loss between the predicted and ground-truth text instances.

### 3.2. Rotated Scene Text Representation

Rectangular bounding boxes [36] (shown in Figure 2-a), of the form $b' = [x, y, w, h]^\top$, are considered the simplest representation of a localized horizontal text region, where $(x, y)$ are the center point coordinates, and $w$ and $h$ are the box's width and height, respectively. Unfortunately, this representation falls short when dealing with irregular text regions [3] as (a) it limits the ability of a given detector to distinguish between overlapped or nearby text regions, and (b) it includes many irrelevant background areas that can affect the detector's loss function during training, and can generate noisy regions that might hinder subsequent analysis, i.e., text recognition.

To address these limitations, several works [4–6, 9, 10, 18, 28–30, 37] have used a rotated bounding box representation as shown in Figure 2-b. In this work, we also adopt a


(a) Rectangle bounding box $(x, y, w, h)$ (b) Rotated Rectangle bounding box $(x, y, w, h, \theta)$

Figure 2. Illustrations of different techniques for representing bounding boxes for scene text detection.

rotated rectangular-bounding boxes representation that embeds the box orientation angle, $\theta$, within the box description as:

$$b = [x, y, w, h, \cos(\theta), \sin(\theta)]^\top \quad (2)$$

where $\theta \in [-90°, 90°)$.

### 3.3. Loss Function

To allow the Transformer architecture to predict the orientation of a text region, we propose a loss function tailored to the task at hand.

Unlike [21], which uses a generic Generalized Intersection over Union (GIoU) with $\ell_1$-regression [38] (shown in Figure 3-a), we propose a rotated-box-based GIoU loss (shown in Figure 3-b), along with a Smooth-ln regression based loss to properly handle rotated texts as follows.

Let $\hat{b}_i$ and $b_j$ denote the $i^{th}$ predicted and $j^{th}$ ground truth bounding boxes, respectively, then we define our loss function as:

$$\mathcal{L}^r_{\text{box}}(\hat{b}_i, b_j) = \lambda_1 \mathcal{L}^r_{reg}(\hat{b}_i, b_j) + \lambda_2 \mathcal{L}^r_{\text{GIoU}}(\hat{b}_i, b_j) \quad (3)$$

where $\lambda_1$ and $\lambda_2 \in \mathbb{R}$ are hyper-parameters, and $\mathcal{L}^r_{reg}(\cdot)$ and $\mathcal{L}^r_{\text{GIoU}}(\cdot)$ are the rotated box based loss functions that will be introduced in (4) and (5), respectively.

**Smooth-ln based Regression Loss:** It is used in computing $\mathcal{L}^r_{reg}(.)$ from (3) as it was found to be more efficient in arbitrary scene text detection than the Smooth-$\ell_1$ loss [39], and is also capable of resisting more outliers, and adjusting the regressive steps [40]. As such, our adopted regression loss is defined as:

$$\mathcal{L}^r_{reg}(\hat{b}_i, b_j) = (|\Delta b_{ij}| + 1)\ln(|\Delta b_{ij}| + 1) - |\Delta b_{ij}| \quad (4)$$

where $\Delta b_{ij} = \hat{b}_i - b_j$ and $|\cdot|$ denotes the absolute operator.

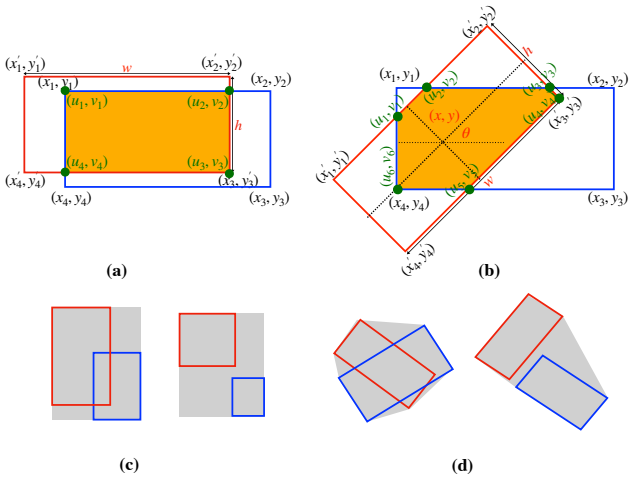



Figure 3. Examples of the intersection (highlighted in *orange*) and convex hull (highlighted in *grey*) computation for horizontal boxes (a) and (c), and for rotated boxes (b) and (d). Note that computing the area of intersection between two rotated bounding boxes can be more complex than the horizontal case.

**Rotated Box based GIoU Loss**: As it was shown in [21], the GIoU loss has a significant impact on the detection performance. In our model, the GIoU loss between the $i^{th}$ predicted and $j^{th}$ ground truth boxes, $\hat{b}_i$ and $b_j$ respectively, is computed as:

$$\mathcal{L}^r_{\text{giou}}(\hat{b}_i, b_j) = 1 - \text{GIoU}(\hat{b}_i, b_j). \quad (5)$$

However, unlike [38] that uses a rectangular bounding box representation for the GIoU loss computation, we use a rotated bounding box representation that better fits text regions. The GIoU for two arbitrarily rotated boxes $\hat{b}_i, b_j \subseteq \mathbb{S} \in \mathbb{R}^n$ is defined as:

$$\text{GIoU}(\hat{b}_i, b_j) = \text{IoU}(\hat{b}_i, b_j) - \frac{\mathbf{Area}(C \backslash (\hat{b}_i \cup b_j))}{\mathbf{Area}(C)} \quad (6)$$

$$\text{with} \quad \text{IoU}(\hat{b}_i, b_j) = \frac{\mathbf{Area}(\hat{b}_i \cap b_j)}{\mathbf{Area}(\hat{b}_i \cup b_j)}, \quad (7)$$

and $C$ denotes the smallest convex hull area that encloses both boxes $\hat{b}_i$ and $b_j$, and $\mathbf{Area}(\cdot)$ is the area of a set. As illustrated in orange in Figure 3-b, the overlapping region of two rotated boxes constructs a polygon ($p$). In the next section, we will describe how we compute the different terms of Equations (5), (6) and (7).

### 3.4. Implementation Details

**Computing the term $\mathbf{Area}(\hat{b}_i \cup b_j)$ in (6) and (7):** In order to calculate the area of an arbitrarily rotated box, $b$, we first obtain the corners of the box using its centered representation, i.e., $b = [x, y, w, h, \cos(\theta), \sin(\theta)]^\top$, as follows

[41]:

$$
\begin{aligned}
x_1 &= x + \frac{-wc_0 + hs_0}{2\gamma}, \quad & y_1 &= y + \frac{-ws_0 - hc_0}{2\gamma}, \\
x_2 &= x + \frac{wc_0 + hs_0}{2\gamma}, \quad & y_2 &= y + \frac{ws_0 - hc_0}{2\gamma}, \\
x_3 &= x + \frac{wc_0 - hs_0}{2\gamma}, \quad & y_3 &= y + \frac{ws_0 + hc_0}{2\gamma}, \\
x_4 &= x + \frac{-wc_0 - hs_0}{2\gamma}, \quad & y_4 &= y + \frac{-ws_0 + hc_0}{2\gamma}.
\end{aligned} \tag{8}
$$

where $\{(x_i, y_i), i = 1, ..., 4\}$ are the coordinates of the box corners in counterclockwise direction (Figure 3), $\gamma = \sqrt{c_0^2 + s_0^2}$, and $c_0$ and $s_0$ are $\cos(\theta)$ and $\sin(\theta)$, respectively. Now, the area of a box can be computed as follows:

$$
\begin{aligned}
\mathbf{Area}(b) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \times \\
\sqrt{(x_2 - x_3)^2 + (y_2 - y_3)^2}
\end{aligned} \tag{9}
$$

By using (9) and (11), the area of the union for two arbitrarily-rotated bounding boxes, i.e., the $i^{th}$ predicted and $j^{th}$ ground truth bounding boxes, can be computed by substituting in the following expression:

$$
\mathbf{Area}(\hat{b}_i \cup b_j) = \mathbf{Area}(\hat{b}_i) + \mathbf{Area}(b_j) - \mathbf{Area}(\hat{b}_i \cap b_j) \tag{10}
$$

**Computing the term $\mathbf{Area}(\hat{b}_i \cap b_j)$ in (7):** We first determine the corners of two rotated boxes $(\hat{b}_i, b_j)$ using (8), and start with one rotated box $(\hat{b}_i)$ as the candidate intersection polygon. Then, we apply the method of sequential cutting [41] for calculating the intersection between an edge in the first candidate box $\hat{b}_i$, i.e., the first line equation $\alpha_i u + \beta_i v + \tau_i = 0$, with any edge in the second box under comparison $b_j$, i.e., the second line equation $\alpha_j u + \beta_j v + \tau_j = 0$, by solving to obtain the coordinates of the lines intersection $(u, v)$, where $\alpha_i, \beta_i, \tau_i$ and $\alpha_j, \beta_j, \tau_j$ are the coefficients of the lines equations that can be obtained independently using the lines corners in (8). We repeat the above process until no more edges remain and we come up with the candidate intersection polygon.

Finally, by using the vertices of resulted intersection polygon $p = \hat{b}_i \cap b_j$, its area can be calculated as follow [42]:

$$
\mathbf{Area}(p) = \left| \frac{\sum_{k=1}^{n} u_k v_{\mathbf{mod}(k+1,n)} - v_k u_{\mathbf{mod}(k+1,n)}}{2} \right| \tag{11}
$$

where $|\cdot|$ denotes the absolute operator, and $\mathbf{mod}(a, b)$ represents the modulo operator that obtains the remainder of dividing $a$ by $b$, and $(u_k, v_k)$ are the coordinates of the $k^{th}$ vertex in the intersection polygon $p$.

Using (10) and (11), the IoU in (7) for two arbitrarily-rotated bounding boxes can now be obtained.

**Computing the variable $C$ in (6):** For computing the convex hull of boxes, the areas highlighted by grey in Figure 3-c and Figure 3-d, we implemented the Andrew's monotone chain algorithm [43]. In this algorithm, after calculating the corner points of two rotated boxes using (8), we sort first the 8 points of two rotated boxes. Next, we go through the points and add each point to the hull. Always after adding a point to the hull, we make sure that the last line of two points in the hull does not make a counter-clockwise turn. We then repeatedly remove the second last two point from the hull, and concatenate the lower and upper hulls that gives the convex hull polygon [44]. At the end, we calculate the area of the obtained polygon using (11).

## 4. Experimental Results

As in [21], we use ResNet-50 as a backbone feature extractor. The whole network with 6 encoders and 6 decoders is trained with a batch size of 2 on four NVIDIA V100 16GB GPUs with AdamW [45] optimizer. Different from [21], we use 300 object queries instead of 100 and replace the original prediction head with our proposed rotated bounding box prediction. We first train the proposed network for $\sim$ 50 epochs on a combination of 10k images of VISD [46] and 10k images of Unreal-Text [47] synthetic datasets and then fine-tune for $\sim$ 200 epochs on each of the real datasets [15–17]. We apply a standard data augmentation for the training images, which involves randomly resizing between 480 and 1033, horizontal flipping, and normalizing.

### 4.1. Datasets

We evaluate our method on three public benchmark datasets that contain images of different locations like street views, traffic signs, shopping mall billboards, *etc.* These datasets also include multi-oriented text instances, which are described as follows:

**ICDAR15:** This dataset [16] contains 1000 images for training and 500 images for testing. The annotations of this dataset are at the word-level represented using quadrilateral boxes at the word level. This dataset is more challenging in orientation, illumination variation, and complex background of text instances than ICDAR13 [48]. Most of the images in this dataset are from indoor environments. The annotations of this dataset are represented using quadrilateral boxes, where the ground-truth annotations are in the four corner vertices format that each annotation box can be expressed as $g = [x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4]^\top$. For fine-tuning of our proposed network on this dataset, we convert the bounding boxes of this dataset from quadrilateral boxes into rotated boxes format by using the mapping function $\Phi$ as:

$$
g \xmapsto{\Phi} b \tag{12}
$$

Table 1. Quantitative comparison among some of the recent text detection methods on ICDAR15 [16], ICDAR17 [17] and MSRA-TD500 [15] datasets using precision (P), recall (R) and F-measure, where bold and underline denote best and second best performances respectively, and "–" refer to Non Available data.

| Method | ICDAR15 | | | ICDAR17 | | | MSRA-TD500 | | |
|---|---|---|---|---|---|---|---|---|---|
| | P | R | F-measure | P | R | F-measure | P | R | F-measure |
| ROTDC [18] | – | – | – | – | – | – | 87.00% | 63.00% | 74.00% |
| RRPN [6] | 84.00% | 77.00% | 80.00% | – | – | – | 82.00% | 69.00% | 75.00% |
| D2MO [28] | 82.00% | **80.00%** | 81.00% | – | – | – | 77.00% | 70.00% | 74.00% |
| EAST [4] | 83.30% | 78.30% | 80.70% | – | – | – | 87.30% | 67.40% | 76.10% |
| TextBoxess++ [5] | 82.20% | 76.40% | 79.20% | – | – | – | – | – | – |
| RRD [29] | 85.60% | 79.00% | 82.20% | – | – | – | 87.00% | 73.00% | 79.00% |
| FOTS [37] | 85.60% | <u>79.80%</u> | 82.80% | 80.90% | 57.50% | 67.20% | – | – | – |
| MOSTD [30] | 87.20% | 76.70% | 81.70% | <u>83.80%</u> | 55.60% | 66.80% | – | – | – |
| PSE-Net [27] | 81.50% | 79.70% | 80.60% | 73.77% | **68.21%** | 70.88% | – | – | – |
| STELA [9] | <u>88.70%</u> | 78.60% | <u>83.33%</u> | 78.70% | 65.50% | <u>71.50%</u> | – | – | – |
| R-YOLO [10] | 87.00% | 78.20% | 82.30% | 78.00% | <u>66.30%</u> | 67.50% | <u>90.20%</u> | <u>81.90%</u> | <u>85.80%</u> |
| *Proposed Method* | **89.83%** | 78.28% | **83.65%** | **84.75%** | 63.23% | **72.42%** | **90.92%** | **83.84%** | **87.23%** |

where $b$ is the annotation in rotated bounding box format (2), and $\Phi$ is the `cv2.minAreaRect`[1] function in OpenCV [49], followed by a conversion that maps the rotation angle $\theta \in [-90°, 90°)$ to match the box representation definition in (2).

**ICDAR17:** It is a large-scale word-level multi-lingual text dataset [17] comprised of 18000 natural scene images, sorted into 7200 for training, 1800 for validation and 9000 for testing. Similar to ICDAR15, This dataset also uses quadrilateral annotations [16], which we convert to our proposed rotated boxes format with the same procedure described in the preceeding paragraph. It is noteworthy to mention that ICDAR17 is more challenging than ICDAR15 due to the varying text instances sizes, and the abundance of tiny text instances.

**MSRA-TD500**: This dataset [15] has been explicitly designed for arbitrarily oriented text detection, which has rotated bounding box representation in the text line level. This dataset contains 200 test and 300 training images of Chinese and English languages. This dataset's images vary from indoor (office and mall) and outdoor (street) scenes. The bounding boxes in this dataset are annotated in $('x, 'y, w, h, \theta)$ format, where $('x, 'y)$ are the coordinates of the top left corner, $w$ and $h$ are the width and height of the box, and $\theta$ represents the rotation angle. This format is mapped to the standard rotated box format in (2) by obtaining the center of the box, and the terms $\cos(\theta)$ and $\sin(\theta)$.

**SVT:** The images of the street view text (SVT) dataset [50] are collected using Google Street View camera. The images are mainly taken from outdoor locations, and it has a large number of text instances with low resolution, and some images are blurry. We only use this dataset for *qualitative results* (Section 4.4) due to its annotations are in rectangular bounding boxes format that does not offer a fair, objective measure when used to assess rotated bounding boxes representation based methods.

[1] https://bit.ly/3dCauBm

### 4.2. Evaluation Metrics

For quantitative evaluation, we use the ICDAR15 IoU Metric [16], which is obtained for the $i^{th}$ ground-truth and $j^{th}$ detection bounding box as shown in (7), where a threshold of IoU $\geq 0.5$ is used for counting a correct detection and therefore calculating the precision ($P$) and recall ($R$). As in [4, 7, 8, 25], we also use the F-measure that is a function in the precision and recall, and it is defined as follow:

$$\text{F-measure} = 2\frac{P \times R}{P + R} \tag{13}$$

### 4.3. Quantitative Results

In this section, a quantitative comparison for the proposed and existing state-of-the-arts methods in [4–6, 9, 10, 18, 27–30, 37] on three challenging datasets, namely, ICDAR15 [16], ICDAR17 [17] and MSRA-TD500 [15] datasets, is presented.

**Detection Accuracy:** As depicted in Table 1, the proposed method offers an F-measure of 83.65% on the ICDAR15 dataset, which outperforms all the methods in comparison, including one-stage [4, 5, 9, 10, 27] and two-stage [6, 30] text detectors. By considering ICDAR17, which is a larger and more challenging dataset than ICDAR15, our proposed method also offers the highest performance in terms of precision and F-measure, than that offered by FOTS [37], MOSTD [30], STELA [9] and R-YOLO [10]. This higher accuracy confirms the advantage of using a Transformer in focusing on the regions of interest.

For the MSRA-TD500 dataset, which requires predicting line-level instead of word-level text detection, as can

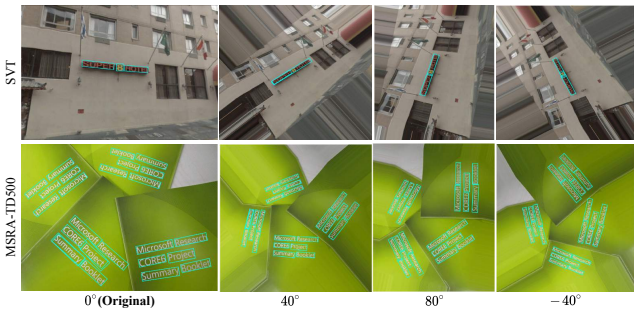



Figure 4. Sample qualitative results showing text detections when rotating the original image from 0° with different orientation angles. The bounding boxes of detected regions are shown with a *cyan* color.

be seen from Table 1 also the proposed method provides the best detection performance compared to the considered state-of-the-art methods [4, 6, 10, 18, 28, 29].

We argue that this high performance of the proposed method is mainly attributed to the attention mechanism that allows the Transformer architecture to relate among different parts of characters of a word or text-line in a given text image to make the final prediction. In addition, utilizing the GIoU loss with the rotated box representation offers the entire architecture a precise detection capability.

**Loss Function Ablation:** We validate the performance gain caused by our proposed loss function by comparing its detection accuracy on ICDAR15 and ICDAR17 datasets against a baseline model. The baseline model [21] uses a rectangular box based prediction head which consists of an $\ell_1$ bounding box regression loss [39], and a rectangular GIoU based loss [38]. On the other hand, the proposed method uses an rotated box based prediction head, consisting of a Smooth-ln loss (4) and a rotated GIoU based loss (5) as presented in Section 3.3. The results in Table 2 show that the proposed rotated box based method outperforms the baseline by a large margin; not to mention that using non-rotated rectangular boxes for text detection exhibit poor results on the multi-oriented datasets.

**Computational Speed:** Using a single NVIDIA RTX 2070 (8GB GPU), our proposed model clocks at an average of 10 FPS inference speed. This speed is higher than some of the segmentation-based [7, 25, 27] and two-stage detectors [6], that require multiple stages of post-processing and regional proposal [3]. Nevertheless, some one-stage detectors *e.g.* STELA [9] and R-YOLO [10] are capable of performing inference at higher speeds when compared to Transformer-based architectures [3, 31] at the cost of a slightly reduced accuracy.

### 4.4. Qualitative Results

**Robustness to Rotated Text:** We also experimented with rotated images at four angles $(-40°, 0°, 40°, 80°)$ and evaluating the proposed method's robustness to different text orientations. Figure 4 illustrates some qualitative samples from this experiment. As it can be seen, the proposed method can detect text instances of various orientations accurately.

Table 2. Effect of using prediction head with a loss function that is based on a rectangular (baseline method [21]) or rotated (proposed method) box representation, where the ICDAR15 [16] and ICDAR17 [17] datasets are used, and P, R and F denote precision, recall and F-measure.

| Method | ICDAR15 | | | ICDAR17 | | |
|---|---|---|---|---|---|---|
| | P | R | F | P | R | F |
| Baseline | 69.77% | 69.23% | 69.50% | 67.46% | **66.00%** | 66.72% |
| *Proposed* | **89.83%** | **78.28%** | **83.65%** | **84.75%** | 63.23% | **72.42%** |

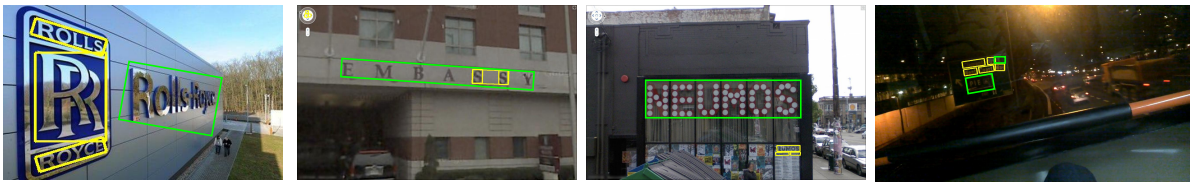
**Challenging Conditions:** Figure 5 illustrates the proposed method's detection results for several challenging cases from ICDAR15, ICDAR17, MSRA-TD500 and SVT datasets. As it can be seen, the proposed method performs well on the first three datasets that include challenging fonts, illumination variation, in-plane rotation, and low contrast text instances. To show the generalization capability of our proposed method, we also experimented with using our ICDAR17 fine-tuned model on a different dataset, namely, the SVT dataset. It can be seen from Figure 5 that the proposed model is able to handle low resolution and rotated texts without requiring any extra fine-tuning, thereby confirming the Transformer's attention modules capability to reason about feature maps in different scales. While our proposed method is designed for detecting multi-orient text, it can be seen from Figure 5 that it is also capable of detecting curved text instances. For example, from this figure, the proposed method detected the curved line-text in the second image of MSRA-TD500 with one bounding box, and it also detected the three curved words in the first image of SVT with three separate boxes.

Figure 6 shows some failure cases of the proposed method. For instance, Figure 6-a characterizes failure cases caused by large perspective distortions, and similar text font color to the background, leading to some missed detections. Also Figure 6-b shows the effect of large separation between the word's characters on the detection; causing our model to only detect a subset of the whole word.

For complex fonts as shown in Figure 6-c, the proposed method also fails to detect the text. We attribute this missing detection to the scarcity of such fonts in the training data. Despite the severe illumination changes and small text instances shown in Figure 6-d, our proposed model was able to detect most instances and only missed a few. The missed detections are mainly caused by the Transformer's reduced performance when detecting text of low-resolution [21, 22].

These challenging examples indicate that there is still a room to improve the proposed scheme's performance by tackling the challenges of complex fonts, illumination variations, low-resolution text and geometric distortions.


Figure 5. Sample qualitative results of the proposed method on some challenging examples from ICDAR15, ICDAR17, MSRA-TD500 and SVT datasets. PO: Partial Occlusion, DF: Difficult Fonts, IV: Illumination Variation, IB: Image Blurriness, LR: Low Resolution, PD: Perspective Distortion, OT: Oriented Text, and CT: Curved Text.



(a) HPD (b) CS (c) DF (d) LR, IV

Figure 6. Qualitative results of failed cases. Yellow and green bounding boxes show the correct detection and missed ground truths, respectively. HPD: High Perspective distortion, CS: Character space, DF: Difficult font, LR: Low Resolution, IV: Illumination Variation.

## 5. Conclusion

We have presented a Transformer-based architecture for multi-oriented text detection in the wild. Extensive experiments on three challenging datasets have solidified the viability of our approach as it outperforms state-of-the-art methods, including recent rotated-bounding-box-based text detectors, in terms of precision and F-measure, while maintaining a favorable recall. Achieving these results would not have been possible without the proposed rotated bounding box representation and its associated loss function, tailored to the multi-oriented text detection problem.

## Acknowledgment

We would like to thank the Ontario Centres of Excellence (OCE), the Natural Sciences and Engineering Research Council of Canada (NSERC), and ATS Automation Tooling Systems Inc., Cambridge, ON, Canada for supporting this research work.

## References

[1] N. Piasco, D. Sidibé, C. Demonceaux, and V. Gouet-Brunet, "A survey on visual-based localization: On the benefit of het-

erogeneous data," *Pattern Recognition*, vol. 74, pp. 90–109, 2018. 1

[2] H. Lin, P. Yang, and F. Zhang, "Review of scene text detection and recognition," *Archives of Computational Methods in Eng.*, pp. 1–22, 2019. 1

[3] Z. Raisi, M. A. Naiel, P. Fieguth, S. Wardell, and J. Zelek, "Text detection and recognition in the wild: A review," *arXiv preprint arXiv:2006.04305*, 2020. 1, 3, 7

[4] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, and J. Liang, "EAST: an efficient and accurate scene text detector," in *Proc. IEEE Conf. on Comp. Vision and Pattern Recognit.*, 2017, pp. 5551–5560. 1, 2, 3, 6, 7

[5] M. Liao, B. Shi, and X. Bai, "Textboxes++: A single-shot oriented scene text detector," *IEEE Trans. on Image process.*, vol. 27, no. 8, pp. 3676–3690, 2018. 2, 6

[6] J. Ma, W. Shao, H. Ye, L. Wang, H. Wang, Y. Zheng, and X. Xue, "Arbitrary-oriented scene text detection via rotation proposals," *IEEE Trans. on Multimedia*, vol. 20, no. 11, pp. 3111–3122, 2018. 2, 3, 6, 7

[7] Y. Baek, B. Lee, D. Han, S. Yun, and H. Lee, "Character region awareness for text detection," in *Proc. IEEE Conf. on Comp. Vision and Pattern Recognit.*, 2019. 2, 6, 7

[8] J. Liu, X. Liu, J. Sheng, D. Liang, X. Li, and Q. Liu, "Pyramid mask text detector," *CoRR*, vol. abs/1903.11800, 2019. 2, 6

[9] L. Deng, Y. Gong, X. Lu, Y. Lin, Z. Ma, and M. Xie, "STELA: A real-time scene text detector with learned anchor," *IEEE Access*, vol. 7, pp. 153 400–153 407, 2019. 2, 3, 6, 7

[10] X. Wang, S. Zheng, C. Zhang, R. Li, and L. Gui, "R-YOLO: A real-time text detector for natural scenes with arbitrary rotation," *Sensors*, vol. 21, no. 3, p. 888, 2021. 1, 2, 3, 6, 7

[11] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. in Neural Info. Process. Sys.*, 2015, pp. 91–99. 1

[12] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Eur. Conf. on Comp. Vision*. Springer, 2016, pp. 21–37.

[13] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. on Comp. Vision and Pattern Recognit.*, 2016, pp. 779–788. 2

[14] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. on Comp. Vision*, 2017, pp. 2961–2969. 1, 2

[15] C. Yao, X. Bai, W. Liu, Y. Ma, and Z. Tu, "Detecting texts of arbitrary orientations in natural images," in *Proc. IEEE Conf. on Comp. Vision and Pattern Recognit.*, 2012, pp. 1083–1090. 1, 2, 5, 6

[16] D. Karatzas, L. Gomez-Bigorda, A. Nicolaou, S. Ghosh, A. Bagdanov, M. Iwamura, J. Matas, L. Neumann, V. R. Chandrasekhar, S. Lu *et al.*, "ICDAR 2015 competition on robust reading," in *Proc. Int. Conf. on Document Anal. and Recognition (ICDAR)*, 2015, pp. 1156–1160. 2, 5, 6, 7

[17] M. Iwamura, N. Morimoto, K. Tainaka, D. Bazazian, L. Gomez, and D. Karatzas, "ICDAR2017 robust reading challenge on omnidirectional video," in *Proc. IAPR Int. Conf. on Document Anal. and Recognition (ICDAR)*, vol. 1, 2017, pp. 1448–1453. 1, 2, 5, 6, 7

[18] R. Endo, Y. Kawai, H. Sumiyoshi, and M. Sano, "Scene-text-detection method robust against orientation and discontiguous components of characters," in *Proc. IEEE Conf. on Comput. Vision and Pattern Recognit. Workshops*, 2017, pp. 1–9. 1, 2, 3, 6, 7

[19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008. 1, 2, 3

[20] W. Chan, C. Saharia, G. Hinton, M. Norouzi, and N. Jaitly, "Imputer: Sequence modelling via imputation and dynamic programming," *arXiv preprint arXiv:2002.08926*, 2020. 1

[21] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," *arXiv preprint arXiv:2005.12872*, 2020. 1, 2, 3, 4, 5, 7

[22] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable detr: Deformable transformers for end-to-end object detection," *arXiv preprint arXiv:2010.04159*, 2020. 1, 3, 7

[23] J. Lee, S. Park, J. Baek, S. Joon Oh, S. Kim, and H. Lee, "On recognizing texts of arbitrary shapes with 2D self-attention," in *IEEE CVPR*, 2020, pp. 546–547. 1

[24] Z. Raisi, M. A. Naiel, P. Fieguth, S. Wardell, and J. Zelek, "2D positional embedding-based transformer for scene text recognition," *Journal of Computational Vision and Imaging Systems*, vol. 6, no. 1, pp. 1–5, 2021. 1

[25] D. Deng, H. Liu, X. Li, and D. Cai, "Pixellink: Detecting scene text via instance segmentation," in *Proc. AAAI Conf. on Artif. Intell.*, 2018. 2, 6, 7

[26] W. Wang, E. Xie, X. Song, Y. Zang, W. Wang, T. Lu, G. Yu, and C. Shen, "Efficient and accurate arbitrary-shaped text detection with pixel aggregation network," in *Proc. IEEE Int. Conf. on Comp. Vision*, 2019, pp. 8440–8449.

[27] W. Wang, E. Xie, X. Li, W. Hou, T. Lu, G. Yu, and S. Shao, "Shape robust text detection with progressive scale expansion network," in *Proc. IEEE/CVF Conf. on Comp. Vision and Pattern Recognit.*, 2019, pp. 9336–9345. 2, 6, 7

[28] W. He, X.-Y. Zhang, F. Yin, and C.-L. Liu, "Deep direct regression for multi-oriented scene text detection," in *Proc. IEEE Int. Conf. on Comp. Vision*, 2017, pp. 745–753. 2, 3, 6, 7

[29] M. Liao, Z. Zhu, B. Shi, G.-s. Xia, and X. Bai, "Rotation-sensitive regression for oriented scene text detection," in *Proc. IEEE Conf. on Comp. Vision and Pattern Recognit.*, 2018, pp. 5909–5918. 6, 7

[30] P. Lyu, C. Yao, W. Wu, S. Yan, and X. Bai, "Multi-oriented scene text detection via corner localization and region segmentation," in *Proc. IEEE Conf. on Comp. Vision and Pattern Recognit.*, 2018, pp. 7553–7563. 2, 3, 6

[31] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, "Transformers in vision: A survey," *arXiv preprint arXiv:2101.01169*, 2021. 2, 7

[32] Y. Liu, S. Zhang, L. Jin, L. Xie, Y. Wu, and Z. Wang, "Omnidirectional scene text detection with sequential-free box discretization," *arXiv preprint arXiv:1906.02371*, 2019. 2

[33] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. on Comput. Vision and Pattern Recognit.*, 2014, pp. 580–587. 2

[34] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proc. IEEE Conf. on Comp. Vision and Pattern Recognit. (CVPR)*, pp. 770–778, 2015. 2

[35] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955. 3

[36] M. Liao, B. Shi, X. Bai, X. Wang, and W. Liu, "Textboxes: A fast text detector with a single deep neural network," in *Proc. AAAI Conf. on Artif. Intell.*, 2017. 3

[37] X. Liu, D. Liang, S. Yan, D. Chen, Y. Qiao, and J. Yan, "FOTS: Fast oriented text spotting with a unified network," in *Proc. IEEE Conf. on Comp. Vision and Pattern Recognit.*, 2018, pp. 5676–5685. 3, 6

[38] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union," June 2019. 4, 7

[39] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. on Comput. Vision*, 2015, pp. 1440–1448. 4, 7

[40] Y. Liu and L. Jin, "Deep matching prior network: Toward tighter multi-oriented text detection," in *Proc. IEEE Conf. on Comp. Vision and Pattern Recognit.*, 2017, pp. 1962–1969. 4

[41] M. Van Kreveld, O. Schwarzkopf, M. de Berg, and M. Overmars, *Computational geometry algorithms and applications*. Springer, 2000. 5

[42] W. H. Beyer, *Standard mathematical tables and formulae*. CRC press, 1991. 5

[43] A. M. Andrew, "Another efficient algorithm for convex hulls in two dimensions," *Information Processing Letters*, vol. 9, no. 5, pp. 216–219, 1979. 5

[44] A. Laaksonen, "Competitive programmer's handbook," *Preprint*, 2017. 5

[45] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017. 5

[46] F. Zhan, S. Lu, and C. Xue, "Verisimilar image synthesis for accurate detection and recognition of texts in scenes," in *Proc. of the European Conf. on Comput. Vision (ECCV)*, 2018, pp. 249–266. 5

[47] S. Long and C. Yao, "Unrealtext: Synthesizing realistic scene text images from the unreal world," *arXiv preprint arXiv:2003.10608*, 2020. 5

[48] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L. G. i Bigorda, S. R. Mestre, J. Mas, D. F. Mota, J. A. Almazan, and L. P. De Las Heras, "ICDAR 2013 robust reading competition," in *Proc. Int. Conf. on Document Anal. and Recognition*, 2013, pp. 1484–1493. 5

[49] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000. 6

[50] K. Wang, B. Babenko, and S. Belongie, "End-to-end scene text recognition," in *Proc. Int. Conf. on Comp. Vision*, 2011, pp. 1457–1464. 6