Few Could Be Better Than All:
Feature Sampling and Grouping for Scene Text Detection

Jingqun Tang[1], Wenqing Zhang[2], Hongye Liu[1], MingKun Yang[2],
Bo Jiang[1], Guanglong Hu[1], Xiang Bai[2*]
[1]NetEase, [2]Huazhong University of Science and Technology
{jingquntang, liuhongye1998, bjiang002, guanglong.hu}@163.com,
{wenqingzhang, yangmingkun, xbai}@hust.edu.cn

Abstract

Recently, transformer-based methods have achieved promising progresses in object detection, as they can eliminate the post-processes like NMS and enrich the deep representations. However, these methods cannot well cope with scene text due to its extreme variance of scales and aspect ratios. In this paper, we present a simple yet effective transformer-based architecture for scene text detection. Different from previous approaches that learn robust deep representations of scene text in a holistic manner, our method performs scene text detection based on a few representative features, which avoids the disturbance by background and reduces the computational cost. Specifically, we first select a few representative features at all scales that are highly relevant to foreground text. Then, we adopt a transformer for modeling the relationship of the sampled features, which effectively divides them into reasonable groups. As each feature group corresponds to a text instance, its bounding box can be easily obtained without any post-processing operation. Using the basic feature pyramid network for feature extraction, our method consistently achieves state-of-the-art results on several popular datasets for scene text detection.

1. Introduction

Scene text detection has been an active research field for a long time, because of its wide range of practical applications, such as scene understanding, automatic driving, and photo translation. As a key prior component of scene text reading, scene text detection aims to precisely locate text in scene images. Despite the noticeable improvement achieved by existing methods [10, 40, 41, 49], it is still a challenging task due to the variety of scene text, *e.g.* different scales, complicated illumination, perspective distortion,

Figure 1. The illustration of feature sampling and grouping. (a) The confidence score map for text regions indicates the pixel importance for text detection. (b) The text features at red points containing geometric and context information of foreground text are selected by scores. (c) The sampled features from the same text instance are implicitly grouped at the feature level by a transformer. (d) The bounding boxes can be easily obtained from the grouped features.

multi-orientations, and complex shapes. Moreover, most scene text detection methods depend on complicated processing to generate or refine the predicted results, such as anchor generation, non-maximum suppression (NMS) [31], binarization [17], or contour extraction [36].

Inspired by the advantages of the transformer [39] in natural language processing (NLP), lots of works [2, 5, 23, 28, 35, 53] introduce it into vision tasks to extract global-range features and model long-distance dependencies in images, while showing promising performance. Especially in object

---
*Corresponding Author

detection, DETR-based methods [2, 28, 53] successfully use transformers to remove the complicated hand-designed processes (*e.g.* NMS and anchor generation) from the former object detection frameworks [8, 20, 33].

Although transformers bring advantages in global-range feature modeling to DETR-based frameworks [2], they may suffer from handling the small objects and the high computational complexity. For instance, a recent DETR-based scene text detector [32] cannot achieve the satisfactory detection accuracy on the ICDAR2015 dataset [12] and ICDAR2017-MLT dataset [30], since the text instances in these two datasets have much larger variance of scales and aspect ratios. It is often insufficient for transformers to capture small text on the feature map at small scales, while the time cost of a DETR-based method with multi-scale feature maps is unpredictable. Essentially, unexpected background noise in higher-resolution feature maps would significantly increase the computational cost and disturb the transformer modeling. Though, some recent works [28, 53] improve the efficiency of transformer-based object detectors by optimizing the attention operations, they fail to achieve the competitive results in scene text detection (refer to the results reported in Tab. 5).

In this paper, we propose a simple yet effective transformer-based architecture for scene text detection. We argue that feature learning with the relationship of all pixels is not necessary, as foreground text instances only occupy a few small and narrow regions in scene images. Intuitively, we firstly sample and collect the features that are highly relevant to scene text as illustrated in Fig. 1(a)(b). Then, we adopt a transformer for modeling the relationship of the sampled features so that they can be properly grouped. As shown in Fig. 1(c)(d), benefiting from the powerful attention mechanism of the transformer, each feature group will correspond to a text instance, which is quite convenient for predicting its bounding box.

Different from the previous scene text detection methods [1, 15, 17, 40, 49, 51] that usually learn the deep representations of scene text images in a holistic manner with CNNs, our detection method based on only a few representative features has three prominent advantages: 1) it can significantly eliminate the redundant background information, which is beneficial for improving the effectiveness and efficiency of the detection process; 2) Using a transformer to group the sampled features, we can obtain more accurate grouping results and bounding boxes without any post-processing operation; 3) As the feature sampling and grouping are implemented in an end-to-end fashion, the two stages can jointly improve the final detection performance. To verify the effectiveness of the proposed feature sampling-and-grouping scheme, we conduct extensive experiments on several popular datasets [3, 11, 12, 30, 47, 48] for scene text detection, consistently achieving the state-of-

of-art results. In addition, the comparison with the recent transformer-based detectors [2, 28, 32, 53] also proves the effectiveness of our method.

2. Related Work

Lots of works on scene text detection have been proposed before, which can be roughly divided into two categories: bottom-up methods and top-down methods.

Bottom-up methods firstly detect/segment the basic components or pixels of scene text, which are then formed into bounding boxes with some heuristic operations. In an early method, CTPN [38] develops a vertical anchor mechanism to predict sequential proposals, and naturally connects them into bounding boxes by a recurrent neural network. To better detect long and dense text, SegLink [34, 37] detects components and links of each text instance, and combines them together to generate the final detection results. In addition, the fundamental components can be defined as characters with affinity boxes (*e.g.* CRAFT [1]) or center points with radius (*e.g.* TextSnake [24]). These methods are more flexible in detecting text with various shapes, as long as the components can be detected and grouped into final results. However, it suffers from missing components and background noise, and the final detection results are susceptible to the grouping post-process. Our proposed method, which is also a bottom-up method, can predict the bounding boxes by sampling and grouping at the feature level while not relying on any post-processing.

Top-down methods directly predict bounding boxes of scene text at the word or line level. Inspired by the popular object detectors [20, 33], some methods [15, 16, 27] adjust default anchors into quadrilaterals or rotated bounding boxes to fit the multi-orientations and various aspect ratios of scene text. EAST [51] directly regresses the coordinates of multi-oriented bounding boxes on the entire feature map. To directly detect curved text in the wild, recent methods [21, 54] adopt Bezier curves or Fourier signatures for locating scene text, and apply extra processes (*e.g.* Bezier-Align, Inverse Fourier Transformation, and NMS) to generate the final detection results. These top-down methods are usually more straightforward than the bottom-up ones, but they still need some hand-designed processes, such as anchor generation, NMS, and binarization.

Inspired by the power of transformers in natural language processing, the pioneer work, DETR [2], presents a novel transformer-based architecture for object detection. It discards several hand-designed processes employed in [8, 20, 33], while achieving promising performance. Although a recent method [32] has tried to apply the DETR-based architecture to scene text detection, it can not achieve a satisfying detection performance on ICDAR2015 [12] and ICDAR2017-MLT [30]. Since scene text is more challenging than common objects for its extreme variance of scales
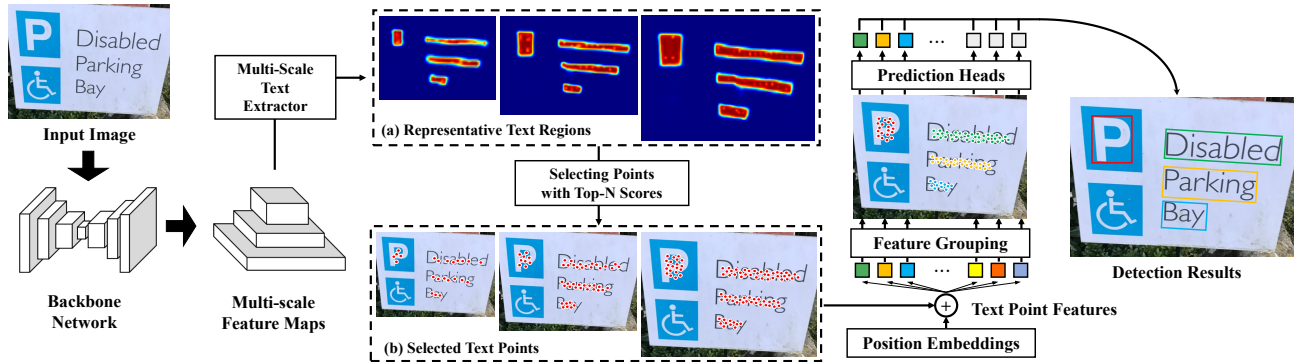
Figure 2. The overview of our proposed transformer-based architecture. It consists of a backbone network, a multi-scale feature sampling network, and a feature grouping network. Specifically, multi-scale feature maps are first produced from the backbone network. Next, a multi-scale text extractor is used to predict the confidence scores of the representative text regions at the pixel level. Then, we select text point features with top-N scores and concatenate them with position embeddings. After that, we adopt a transformer to model the relationship between the sampled features and implicitly group them into fine representations by the attention mechanism. Finally, the detection results are obtained from the prediction heads.

and aspect ratios, transformers cannot obtain sufficient information from a single scale feature map. The multi-scale scheme can somewhat cope with this problem, but it incurs a huge computational overhead for transformers. Different from those DETR-based methods [28, 53] focusing on improving the attention units, we propose to eliminate redundant background information directly and select a few important features [52] from multi-scale feature maps. Thus, both computational overhead and the quality of sampled features can be taken into account, which facilitates transformers being better employed for text detection.

3. Methodology

In this section, we first introduce the overall architecture of the proposed scene text detection method. Then, we elaborate on the proposed feature sampling-and-grouping scheme and further analyze the advantages of feature sampling in transformer modeling. Finally, we describe the details about the training of our proposed method.

3.1. Network Architecture

As shown in Fig. 2, our proposed transformer-based architecture is composed of a backbone network, a feature sampling network, and a feature grouping network.

The backbone is the basic feature pyramid network (FPN) [18] equipped with ResNet-50 [9]. The produced feature maps $\mathbf{F}$ in three different scales (*i.e.* 1/4, 1/8, 1/16) are used for feature sampling.

In our feature sampling network, the three feature maps are first down-sampled to smaller scales (*i.e.* 1/8, 1/16, 1/32) by a Coord-Convolution layer [19] and a constrained deformable pooling layer. Then, several convolution layers

are employed to generate confidence score maps to distinguish representative text regions. After that, we only select the features with top-N_k scores in each scale layer k, and gather them into a sequence form with a shape $(\sum_k N_k, C)$, where C is the channel number.

In our feature grouping network, the sampled features are first concatenated with position embeddings. Then, we adopt transformer encoder layers to model their relationships, and implicitly aggregate the features from the same text instance. Finally, scores and coordinates of bounding boxes (or polygons) are obtained via a text/non-text classification head and a text detection head, respectively.

3.2. Feature Sampling

Despite the novel structure and promising performance in object detection, transformer-based methods [2, 32] can not perform well on scene text detection due to the extreme variance of scales and aspect ratios. Following previous text detectors [14, 17, 18, 40], we use multi-scale features from the FPN to boost the detection performance. Nevertheless, such a scheme incurs unbearable computational cost and much longer convergence time for transformers. We observe that foreground text instances only occupy small and narrow regions, and useful information for localizing text is relatively sparse. Hence, we propose a feature sampling network to decrease redundant background noise involved by multi-scale features, reducing the computational complexity and facilitating feature learning for transformers.

Multi-Scale Text Extractor To sample representative features from foreground text, we apply a simple multi-scale text extractor to predict the confidence scores for text regions at the pixel level. Following CoordConv [19], we

first concatenate each feature map with two extra channels of normalized coordinates to introduce location information. Let $\mathbf{F}$ denote the feature maps from the FPN in different scales (*i.e.* 1/4, 1/8, 1/16), and

$$\mathbf{F} = \{f_k \in \mathbb{R}^{H_k \times W_k \times C} | k = 0, 1, 2\}. \quad (1)$$

Then the position information is injected via

$$\widehat{f_k} = Conv(f_k \oplus C_k), \quad (2)$$

where $\oplus$ stands for the concatenation operation, and $C_k \in \mathbb{R}^{H_k \times W_k \times 2}$ denotes the normalized coordinates.

Inspired by deformable ROI pooling [4], we specifically design a constrained one to down-sample the multi-scale feature maps. Since the text area is relatively concentrated, the predicted offsets in deformable pooling with further distance will introduce irrelevant information into the pooled features. Thus, we add a learnable scaling parameter to constrain the predicted offsets, and pool $\widehat{f_k}$ to $\tilde{f}_k$ with smaller scales (*i.e.* 1/8, 1/16, 1/32).

Finally, we construct a simple scoring net $\mathbb{S}$ composed of convolution layers and a Sigmoid function to generate the confidence score maps for representative text regions at all scales. To better distinguish the importance of pixels at different positions in each text instance, different scores over positions are used for supervision. To generate the score maps, we adjust the Gaussian heatmap generation in general object detection [6, 13] for text instances in the word level. Specifically, a two-dimensional Gaussian distribution is implemented to generate the ground truth $\mathbf{S^t} = \{S_k^t | k = 0, 1, 2\}$ for $\mathbb{S}$, ensuring that the central part of each text instance has the highest importance score, and the scores gradually decrease from the center to contours.

Feature Sampling To reduce the redundant background noise, we design a strategy for selecting representative features that are highly relevant to foreground text. These features, containing rich geometric and context information of foreground text, would be sufficient for text localization.

Let $\mathbf{S}$ denote the predicted score maps, and

$$\mathbf{S} = \{S_k \in \mathbb{R}^{H'_k \times W'_k} | S_k = \mathbb{S}(\tilde{f}_k), k = 0, 1, 2\}. \quad (3)$$

Then, we sort scores in S_k, and select features with top-N_k scores in $\tilde{f}_k$ of each scale, respectively. The selected features are gathered into $\bar{\mathbf{F}} \in \mathbb{R}^{N \times C}$ for the incoming transformer modeling:

$$\bar{\mathbf{F}} = [\bar{f}_n \in \mathbb{R}^C | n = 0, 1, ..., N], \quad (4)$$

where $N = \sum_{k=0}^{2} N_k$, and N_k is the number of selected features in different scales.

Thus, the number of enormous features at all scales can be significantly reduced. The primary selected features are

probably from foreground text regions, which would contain sufficient geometric and context information for text detection.

3.3. Feature Grouping

Through feature selection, only a few representative features that are highly relevant to foreground text are concatenated for the incoming transformer modeling. To reserve the position information of the sampled features, we add the position embeddings into $\bar{\mathbf{F}}$. Then, we adopt a transformer structure to implicitly aggregate features from the same text instance by attention mechanism. The basic form is a stacked network with four transformer encoder layers, which are composed of self-attention modules, feed-forward layers, and layer normalization. Following [39], we construct our self-attention module as

$$Attn(\widehat{\mathbf{F}}) = softmax(\frac{Q(\widehat{\mathbf{F}})K(\widehat{\mathbf{F}})^T}{\sqrt{C'}})V(\widehat{\mathbf{F}}), \quad (5)$$

where $\widehat{\mathbf{F}} \in \mathbb{R}^{N \times C'}$ denotes the sampled features with position embeddings, and C' is the channel number. Q, K and V denote the different linear layers.

For previous methods [2, 28], the core issue of applying the attention operation on a feature map $\boldsymbol{x} \in \mathbb{R}^{H \times W \times C'}$ is the computational complexity on all spatial locations. In the original DETR [2] encoder, the complexity of attention operation is $O((HW)^2 C')$, which is quadratic with the spatial size. However, in our method, it is only related to the number N of selected features $\widehat{\mathbf{F}}$, and the complexity becomes $O(N^2 C')$. In our implementation, the selected number $N^2 \ll (HW)^2$, and thus the complexity of our transformer could be significantly reduced.

Finally, the output text features are fed into two prediction heads for classification and text detection. The text detection head is composed of fully-connected layers and a Sigmoid function. It can regress the coordinates of rotated bounding boxes in the form of $\mathcal{B}(x, y, h, w, \theta)$ or 8 control points of Bezier-Curve [21] for arbitrary-shaped text. x, y, h, w, and θ are the coordinates of the center point, height, width, and angle, respectively.

3.4. Optimization

The proposed model is trained in an end-to-end manner, and the objective function consists of three parts as follows:

$$\mathcal{L} = \lambda_c \widehat{\mathcal{L}}_{class} + \lambda_d \widehat{\mathcal{L}}_{det} + \lambda_f \mathcal{L}_{fs}, \quad (6)$$

where $\widehat{\mathcal{L}}_{class}$ is the loss for classification, $\widehat{\mathcal{L}}_{det}$ is the loss for text detection, and $\mathcal{L}_{fs}$ is the loss for feature selection. λ_c, λ_d, and λ_f are scaling factors. Following DETR [2], we adopt Hungarian algorithm for pair-wise matching before calculating losses for $\widehat{\mathcal{L}}_{class}$ and $\widehat{\mathcal{L}}_{det}$.

Figure 3. The qualitative results of our proposed method in different cases, including multi-oriented text, long text, multi-lingual text, low-resolution text, curved text, dense text. For curved text detection, the Bezier curves' control points are drawn in red.

Loss for classification We adopt the Cross Entropy loss for text/non-text classification after pair-wise matching by Hungarian algorithm. It can be formulated as

$$\widehat{\mathcal{L}}_{class} = \frac{1}{N} \sum_x -[\widehat{g}_x \cdot \log(\widehat{p}_x) + (1 - \widehat{g}_x) \cdot \log(1 - \widehat{p}_x)], \quad (7)$$

where N is the total number of selected features, $\widehat{g}_x$ represents the label of sample x and $\widehat{p}_x$ represents the predicted probability. The elements with $\widehat{}$ denote the probabilities or labels of the matched samples after pair-wise matching.

Loss for text detection For multi-oriented text detection, we adapt the Gaussian Wasserstein Distance (GWD) loss [45] into a scale-invariant form to better balance the loss weights of text with different scales. Due to the extreme variance of scales, the loss of small text has a negligible influence on the gradient back-propagation compared with the loss of large text. Hence, we adjust the GWD loss as follows:

$$\widehat{\mathcal{L}}_{det} = \frac{1}{N_r} \sum_x (1 - \frac{1}{\tau + f(d^2(\frac{\widehat{u}_x}{|\widehat{t}_x|}, \frac{\widehat{t}_x}{|\widehat{t}_x|}))}), \quad (8)$$

where $\widehat{u}_x$ denotes the predicted rotated bounding box, $\widehat{t}_x$ denotes the target one, and $|*|$ denotes its area. N_r is the number of bounding boxes after pair-wise matching. The elements with $\widehat{}$ denote the matched bounding boxes or the target ones after pair-wise matching. $f(\cdot)$ represents a non-linear function, and τ is a hyper-parameter to modulate the loss. d^2 will be explained in the Appendix. According to the GWD loss [45], we set $f(d^2) = \log(d^2 + 1)$ and $\tau = 3$. By normalizing $\widehat{u}_x$ and $\widehat{t}_x$ with the area of $\widehat{t}_x$, we can decrease the negative effect of the scale imbalance.

For arbitrary-shaped text detection, we adopt the losses for Bezier-Curve in ABC-Net [21]. Thus, the prediction

head for text detection is changed to two heads for predicting both bounding boxes and the control points of Bezier curves, respectively. In the bounding box prediction head, the center point coordinates, box width and box height are predicted for each bounding box $\bar{\mathcal{B}}(x, y, h, w)$. In the Bezier curve prediction head, it predicts the coordinates of 8 control points for each text instance.

Loss for feature selection We apply a smooth L1 loss for optimizing the importance score maps in our feature selection as follows:

$$\mathcal{L}_{fs} = \frac{1}{N_f} \sum_k L1_{smooth}\{S_k, S_k^t\}, k = 0, 1, 2, \quad (9)$$

where N_f is the total size of all score maps. S_k and S_k^t are the predicted score map and the target map, respectively.

4. Experiments

In this section, we first introduce the datasets and implementation details in our experiments. Then, we present the evaluation results on public benchmarks and an ablation study on feature sampling. Finally, we compare our proposed method with some popular transformer-based detection methods.

4.1. Datasets

SynthText [7] is a large synthetic dataset including 800k images. It is only used to pre-train our models.

ICDAR 2015 (IC15) [12] contains 1000 training images and 500 testing images in English, most of which are severely distorted or blurred. All images are annotated with quadrilateral boxes at the word level.

MLT-2017 (MLT17) [30] is proposed for multi-lingual scene text detection. It contains 7200 training images, 1800

Method	ICDAR 2015			MSRA-TD500			Total-Text			CTW1500		
	P	R	F	P	R	F	P	R	F	P	R	F
TextSnake [24]	84.9	80.4	82.6	83.2	73.9	78.3	82.7	74.5	78.4	67.9	85.3	75.6
TextField [44]	84.3	83.9	84.1	87.4	75.9	81.3	81.2	79.9	80.6	83.0	79.8	81.4
PSE-Net [40]	86.9	84.5	85.7	-	-	-	84.0	78.0	80.9	84.8	79.7	82.2
CRAFT [1]	89.8	84.3	86.9	88.2	78.2	82.9	87.6	79.9	83.6	86.0	81.1	83.5
PAN [41]	84.0	81.9	82.9	84.4	83.8	84.1	89.3	81.0	85.0	86.4	81.2	83.7
DB [17]	91.8	83.2	87.3	91.5	79.2	84.9	87.1	82.5	84.7	86.9	80.2	83.4
ContourNet [42]	87.6	86.1	86.9	-	-	-	86.9	83.9	85.4	84.1	83.7	83.9
DRRG [49]	88.5	84.7	86.6	88.1	82.3	85.1	86.5	84.9	85.7	85.9	83.0	84.5
MOST [10]	89.1	87.3	88.2	90.4	82.7	86.4	-	-	-	-	-	-
Raisi *et al.* [32]	89.8	78.3	83.7	90.9	83.8	87.2	-	-	-	-	-	-
TextBPN [50]	-	-	-	86.6	84.5	85.6	90.7	85.2	87.9	86.5	83.6	85.0
Ours (RBox)	90.9	87.3	**89.1**	91.6	84.8	**88.1**	-	-	-	-	-	-
Ours (Bezier)	91.1	86.7	88.8	91.4	84.7	87.9	90.7	85.7	**88.1**	88.1	82.4	**85.2**

Table 1. Detection results on ICDAR2015, MSRA-TD500, Total-Text, and CTW1500. "P", "R", and "F" represent Precision, Recall, and F- measure, respectively.

validation images, and 9000 testing images. All images are annotated with quadrilateral boxes at the word level.

MSRA-TD500 [47] is a multi-lingual text dataset in Chinese and English. It includes 300 training images and 200 testing images with multi-oriented long text. Following previous works [10, 17, 24], we include HUST-TR400 [46] as the extra training data in the fine-tuning stage.

MTWI [11] is a large-scale dataset for Chinese and English web text reading. It contains some challenging cases, such as complex layout, small text, and watermarks. There are 10000 training images and 10000 images for testing, and all text instances are annotated at the line level.

Total-Text [3] is a dataset that contains text of various shapes, including horizontal, multi-oriented, and curved. It contains 1255 training images and 300 testing images, and the text instances are labeled at the word level.

CTW1500 [48] is a curved text dataset, which consists of 1000 training images and 500 testing images. The text instances are annotated at the text-line level.

4.2. Implementation Details

Our model for oriented text detection is denoted as **Ours (RBox)**, and that for arbitrary-shaped text detection is denoted as **Ours (Bezier)**. **Ours (RBox)** is first pre-trained on SynthText for 150 epochs, and then fine-tuned on each corresponding real-world dataset for another 100 epochs. **Ours (Bezier)** follows the experiment settings of ABC-Net [21], and adds its Bezier Curve Synthetic Dataset for pretraining. We optimize our models by AdamW [25] with a weight decay of $1e^{-4}$ and a momentum of 0.9. The initial learning rate for pre-training and fine-tuning is $1e^{-3}$ and $5e^{-4}$, respectively. Both of them will decay to $1e^{-4}$ after the 40th

epoch. More details can be referred to Appendix.

4.3. Evaluation on Benchmarks

To compare with previous scene text detectors, we evaluate our proposed method on several popular benchmarks for scene text detection. We adopt the best model configuration in the #5 of Tab. 4 for evaluating on all benchmarks. As shown in Fig. 3, we provide some qualitative results in different cases, including multi-oriented text, long text, multilingual text, small text, low-resolution text, and curved text.

Multi-oriented text detection We evaluate our method for multi-oriented text on the IC15 dataset and the MSRA-TD500 dataset, which contain lots of small, low-resolution, and long text instances. As shown in Tab. 1, our model outperforms previous state-of-the-art method by 0.9% on both IC15 and MSRA-TD500. Compared with the former DETR-based method [32], our proposed model shows a much better detection performance (89.1% *vs.* 83.7%) on small and blurry text of IC15. Compared with previous CNN-based methods on MSRA-TD500, our method outperforms them by at least 1.7% in terms of f-measure, owing to the advantages of transformers in extracting global-range features and long-distance dependencies. With the help of Swin-transformer [23], our model could further improve the detection performance as shown in Tab. 7.

Curved text detection To prove our method's effectiveness on curved text, we evaluate it on two popular curved text benchmarks, *i.e.* the Total-Text dataset and the CTW1500 dataset. As shown in Tab. 1, our method obtains 0.2% improvement in terms of f-measure compared with the state-of-the-art method TextBPN [50]. With the help of Bezier-Curve [21], our method could generate polygons for curved

Method	P	R	F	FPS
Corner [26]	83.8	55.6	66.8	-
CRAFT [1]	80.6	68.2	73.9	8.6
PSE-Net [40]	73.8	68.2	70.7	-
DB [17]	83.1	67.9	74.7	**19.0**
DRRG [49]	75.0	61.0	67.3	-
Xiao *et al.* [43]	84.2	72.8	78.1	-
MOST [10]	82.0	72.0	76.7	10.1
Raisi *et al.* [32]	84.8	63.2	72.4	-
Ours (RBox)	87.3	73.2	**79.6**	13.1

Table 2. Detection results on the MLT-2017 test dataset.

Method	P	R	F	FPS
SegLink * [34]	70.0	65.4	67.6	-
TextBoxes++ * [15]	66.8	56.3	61.1	-
Seglink++ [37]	74.7	69.7	72.1	-
BDN † [22]	77.3	70.0	73.4	2.7
PAN † [41]	78.9	68.9	73.5	16.9
MOST [10]	78.8	71.1	74.7	**23.5**
Ours (RBox)	78.4	72.3	**75.2**	21.5

Table 3. Detection results on the MTWI dataset. * and †indicate that the results are reported by SegLink++ [37] and MOST [10], respectively.

ID	Sampled Features			IC15			MLT17 val		
	L0	L1	L2	P	R	F	P	R	F
#1	64	-	-	75.2	60.4	67.0	79.9	53.2	63.9
#2	64	128	-	86.5	78.3	82.2	82.7	65.9	73.4
#3	16	32	64	82.4	73.7	77.8	78.9	61.1	68.9
#4	32	64	128	88.1	84.0	86.0	84.1	72.8	78.0
#5	64	128	256	90.9	87.3	**89.1**	86.8	73.4	**79.5**
#6	128	256	512	90.2	87.9	89.0	85.9	73.8	79.4
#7	Adaptive Sampling			90.7	87.2	88.9	-	-	-

Table 4. The experiments of feature sampling number on the IC15 test dataset and the MLT17 validation dataset. "L0", "L1" and "L2" denote the feature maps in different scales (*i.e.* 1/32, 1/16, 1/8).

text, which can not be precisely detected by the former DETR-based method [32]. Moreover, our method with Bezier-Curve could also achieve state-of-the-performance performance on the IC15 and MSRA-TD500 datasets.

Multi-lingual text detection To demonstrate the robustness of our model for different languages, we evaluate it on two large-scale scene text datasets (*i.e.* the MLT17 test dataset and the MTWI dataset). As shown in Tab. 2, compared with the state-of-the-art model [43], our model obtains 3.1%, 0.4%, and 1.5% improvements in terms of precision, recall, and f-measure, respectively. We also evaluate our model on the MTWI dataset, which contains multi-lingual text from web images. Our method achieves the best performance 75.2% in terms of f-measure with a competitive inference speed (21.5 FPS).

4.4. Experiments on Feature Sampling

To demonstrate the effectiveness of our proposed feature sampling scheme, we conduct several experiments with different sampling configurations on the IC15 dataset and the MLT17 validation dataset. As shown in #1, #2, and #5 of Tab. 4, our method can significantly improve the performance with the help of higher-resolution feature maps. For IC15, sampling features at all scales outperforms the

other two configurations by 22.1% and 6.9%, respectively. Consistently, it achieves 15.6% and 6.1% performance gain compared with others on MLT17. In addition, we conduct four configurations to explore the effects of sampling numbers from #3 to #6 in Tab. 4. We observe that the performance can increase with more sampling features, but stagnates in the last. The models with fewer sampled features can not perform well, because these features do not contain enough geometric and context information of all text instances. From #5 and #6, we find the performance slightly decreases as the sampling number increases, which may introduce more redundant features and incur negative effects.

To further evaluate the impact of sampling points, we we adopt an adaptive sampling scheme for training in #7. For every training image, we sort all features from the foreground text area by the predicted scores, and sample a fixed percentage (25%) of them with top scores. In this way, the sampling number is adaptive to the foreground feature number, and the performance of adaptive sampling is close to #5 and #6. Hence, our method is not sensitive when the sampling number is larger than that of #5. Moreover, we try to use all the features in different scales for the transformer modeling, but encounter the issue of "Out Of Memory" during training. Assuming the size of input images is 1024×1024, the sizes of $L0$, $L1$, and $L2$ would be 32×32, 64×64, and 128×128, respectively. The whole features mixed with background are difficult to model, and lead to a huge computational cost which is nearly 1400 times more than that of #5. Thus, our feature sampling is effective to decrease complexity for multi-scale feature maps and preserve the important information for scene text detection.

4.5. Comparisons with Transformer-Based Detection Methods

In this part, we compare our model with some popular transformer-based methods (*i.e.* DETR [2], Deformable DETR [53], and Conditional DETR [29]) in object detection. We use their official codes and follow our training settings for fair comparisons. Noticeably, we adjust their codes
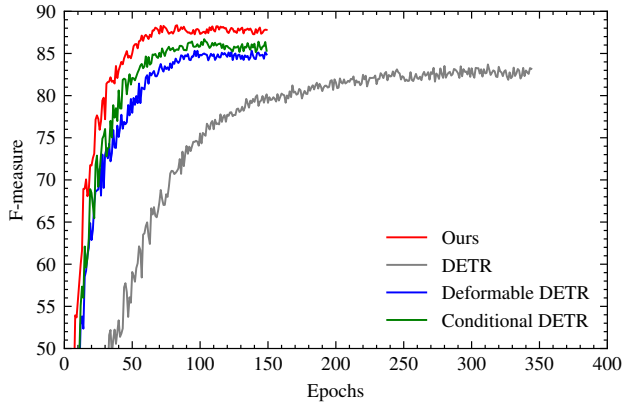
Figure 4. The convergence curves for DETR, Deformable DETR, Conditional DETR and Ours (RBox) on SynthText. The training and validation set is split from SynthText with a ratio 8:2. We train the previous methods by adjusting their official codes for multi-oriented text detection and follow the same settings as ours.

Methods	IC15			MLT17 val		
	P	R	F	P	R	F
DETR* [2]	87.9	75.4	81.2	84.6	63.4	72.5
Deformable DETR* [53]	88.3	84.7	86.5	86.5	69.3	77.0
Conditional DETR* [29]	87.5	81.8	84.6	85.9	67.8	75.8
Raisi *et al.* [32]	89.8	78.3	83.7	-	-	-
Ours (RBox)	90.9	87.3	**89.1**	86.8	73.4	**79.5**

Table 5. Comparisons with transformer-based methods on the IC15 test dataset and the MLT17 validation dataset. * indicates the methods are trained by adjusting their official codes for multi-oriented text detection.

Method	FLOPs	Params	FPS
DETR [2]	38.9G	41.3M	9.7
Deformable DETR [53]	36.8G	39.8M	7.6
Conditional DETR [29]	42.2G	43.2M	9.1
Ours (RBox)	35.9G	38.3M	12.9

Table 6. Comparisons with transformer-based methods on FLOPs, the number of parameters, and the inference speed.

Transformer Layer	IC15			MLT17 val		
	P	R	F	P	R	F
Basic Layer	90.8	87.3	89.1	86.8	73.4	79.5
Swin Transformer Layer	90.9	88.1	**89.5**	87.2	73.4	**79.7**

Table 7. The experiment on the transformer layers in our feature grouping network. Details can be referred to the Appendix.

for multi-oriented text detection by adding angle regression and using our loss function.

Since pre-training on SynthText is a necessary step in previous methods [10, 17, 42, 49], we first compare the convergence speed on SynthText. We train most models excluding DETR with the same training settings as ours, but train DETR for 350 epochs for its low convergence speed. As illustrated in Fig. 4, the convergence speed of our method is much faster than DETR, because ours can significantly reduce redundant information and ease transformer modeling. Compared with the other two methods [29, 53] focusing on increasing the efficiency of the attention units, our feature sampling-and-grouping scheme has a simpler pipeline but demonstrates a competitive convergence speed with a better detection performance. After fine-tuning, our proposed model obtains the best detection performance in terms of f-measure on IC15 and MLT17 as shown in Tab. 5.

In addition, we compare the FLOPs, the number of model parameters, and the inference speed with the previous transformer-based methods. For a fair comparison, we resize both sides of input images to 640 for all models to calculate the FLOPs, and use the same images from the IC15 test dataset to measure the inference speed by FPS. The number of object queries is set to 100 for previous methods, and we adopt the #5 configuration in Tab. 4 for ours. As shown in Tab. 6, our proposed transformer-based architecture has a lower computational cost in terms of FLOPs and a faster inference speed.

4.6. Limitation

For our feature sampling-and-grouping scheme, it is hard to deal with the "text overlapping" cases, which mean two

text instances overlap each other. Although our feature grouping network can model the relationship of the sampled features, the features of the overlapping text are quite complex and tangled. Thus, our proposed method sometimes fails in these cases, which are shown in the Appendix.

5. Conclusion

In this paper, we present a simple yet effective transformer-based architecture for scene text detection. Different from previous methods in scene text detection, our method leverages only a few representative features containing sufficient geometric and context information of foreground text. It is able to effectively reduce the redundant background noise and overcome the complexity limitation of the self-attention module. With the power of transformers, we can obtain more accurate bounding boxes without any post-processing. Through extensive experiments, we demonstrate the effectiveness of our proposed method by consistently achieving state-of-the-art results on both multi-oriented text datasets and arbitrary-shaped text datasets.

References

[1] Youngmin Baek, Bado Lee, Dongyoon Han, Sangdoo Yun, and Hwalsuk Lee. Character region awareness for text detection. In *CVPR*, 2019. 2, 6, 7

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 1, 2, 3, 4, 7, 8

[3] Chee Kheng Ch'ng and Chee Seng Chan. Total-text: A comprehensive dataset for scene text detection and recognition. In *ICDAR*, 2017. 2, 6

[4] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *ICCV*, 2017. 4

[5] Xiyang Dai, Yinpeng Chen, Jianwei Yang, Pengchuan Zhang, Lu Yuan, and Lei Zhang. Dynamic detr: End-to-end object detection with dynamic attention. In *ICCV*, 2021. 1

[6] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection. In *ICCV*, 2019. 4

[7] Ankush Gupta, Andrea Vedaldi, and Andrew Zisserman. Synthetic data for text localisation in natural images. In *CVPR*, 2016. 5

[8] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017. 2

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 3

[10] Minghang He, Minghui Liao, Zhibo Yang, Humen Zhong, Jun Tang, Wenqing Cheng, Cong Yao, Yongpan Wang, and Xiang Bai. Most: A multi-oriented scene text detector with localization refinement. In *CVPR*, 2021. 1, 6, 7, 8

[11] Mengchao He, Yuliang Liu, Zhibo Yang, Sheng Zhang, Canjie Luo, Feiyu Gao, Qi Zheng, Yongpan Wang, Xin Zhang, and Lianwen Jin. Icpr2018 contest on robust reading for multi-type web images. In *ICPR*, 2018. 2, 6

[12] Dimosthenis Karatzas, Lluis Gomez-Bigorda, Anguelos Nicolaou, Suman Ghosh, Andrew Bagdanov, Masakazu Iwamura, Jiri Matas, Lukas Neumann, Vijay Ramaseshan Chandrasekhar, Shijian Lu, et al. Icdar 2015 competition on robust reading. In *ICDAR*, 2015. 2, 5

[13] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *ECCV*, 2018. 4

[14] Minghui Liao, Guan Pang, Jing Huang, Tal Hassner, and Xiang Bai. Mask textspotter v3: Segmentation proposal network for robust scene text spotting. In *ECCV*, 2020. 3

[15] Minghui Liao, Baoguang Shi, and Xiang Bai. Textboxes++: A single-shot oriented scene text detector. *TIP*, 27(8):3676–3690, 2018. 2, 7

[16] Minghui Liao, Baoguang Shi, Xiang Bai, Xinggang Wang, and Wenyu Liu. Textboxes: A fast text detector with a single deep neural network. In *AAAI*, 2017. 2

[17] Minghui Liao, Zhaoyi Wan, Cong Yao, Kai Chen, and Xiang Bai. Real-time scene text detection with differentiable binarization. In *AAAI*, 2020. 1, 2, 3, 6, 7, 8

[18] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 3

[19] Rosanne Liu, Joel Lehman, Piero Molino, Felipe Petroski Such, Eric Frank, Alex Sergeev, and Jason Yosinski. An intriguing failing of convolutional neural networks and the coordconv solution. *NeurIPS*, 2018. 3

[20] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *ECCV*, 2016. 2

[21] Yuliang Liu, Hao Chen, Chunhua Shen, Tong He, Lianwen Jin, and Liangwei Wang. Abcnet: Real-time scene text spotting with adaptive bezier-curve network. In *CVPR*, 2020. 2, 4, 5, 6

[22] Yuliang Liu, Sheng Zhang, Lianwen Jin, Lele Xie, Yaqiang Wu, and Zhepeng Wang. Omnidirectional scene text detection with sequential-free box discretization. In *IJCAI*, 2019. 7

[23] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021. 1, 6

[24] Shangbang Long, Jiaqiang Ruan, Wenjie Zhang, Xin He, Wenhao Wu, and Cong Yao. Textsnake: A flexible representation for detecting text of arbitrary shapes. In *ECCV*, 2018. 2, 6

[25] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 6

[26] Pengyuan Lyu, Cong Yao, Wenhao Wu, Shuicheng Yan, and Xiang Bai. Multi-oriented scene text detection via corner localization and region segmentation. In *CVPR*, 2018. 7

[27] Jianqi Ma, Weiyuan Shao, Hao Ye, Li Wang, Hong Wang, Yingbin Zheng, and Xiangyang Xue. Arbitrary-oriented scene text detection via rotation proposals. *TMM*, 20(11):3111–3122, 2018. 2

[28] Depu Meng, Xiaokang Chen, Zejia Fan, Gang Zeng, Houqiang Li, Yuhui Yuan, Lei Sun, and Jingdong Wang. Conditional detr for fast training convergence. In *ICCV*, 2021. 1, 2, 3, 4

[29] Depu Meng, Xiaokang Chen, Zejia Fan, Gang Zeng, Houqiang Li, Yuhui Yuan, Lei Sun, and Jingdong Wang. Conditional detr for fast training convergence. In *ICCV*, 2021. 7, 8

[30] Nibal Nayef, Fei Yin, Imen Bizid, Hyunsoo Choi, Yuan Feng, Dimosthenis Karatzas, Zhenbo Luo, Umapada Pal, Christophe Rigaud, Joseph Chazalon, et al. Icdar2017 robust reading challenge on multi-lingual scene text detection and script identification-rrc-mlt. In *ICDAR*, 2017. 2, 5

[31] Alexander Neubeck and Luc Van Gool. Efficient non-maximum suppression. In *ICPR*, 2006. 1

[32] Zobeir Raisi, Mohamed A Naiel, Georges Younes, Steven Wardell, and John S Zelek. Transformer-based text detection in the wild. In *CVPR Workshop*, 2021. 2, 3, 6, 7, 8

[33] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: towards real-time object detection with region proposal networks. *PAMI*, 39(6):1137–1149, 2016. 2

[34] Baoguang Shi, Xiang Bai, and Serge Belongie. Detecting oriented text in natural images by linking segments. In *CVPR*, 2017. 2, 7

[35] Zhiqing Sun, Shengcao Cao, Yiming Yang, and Kris M Kitani. Rethinking transformer-based set prediction for object detection. In *ICCV*, 2021. 1

[36] Satoshi Suzuki et al. Topological structural analysis of digitized binary images by border following. *Computer vision, graphics, and image processing*, 30(1):32–46, 1985. 1

[37] Jun Tang, Zhibo Yang, Yongpan Wang, Qi Zheng, Yongchao Xu, and Xiang Bai. Seglink++: Detecting dense and arbitrary-shaped scene text by instance-aware component grouping. *PR*, 96:106954, 2019. 2, 7

[38] Zhi Tian, Weilin Huang, Tong He, Pan He, and Yu Qiao. Detecting text in natural image with connectionist text proposal network. In *ECCV*, 2016. 2

[39] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, 2017. 1, 4

[40] Wenhai Wang, Enze Xie, Xiang Li, Wenbo Hou, Tong Lu, Gang Yu, and Shuai Shao. Shape robust text detection with progressive scale expansion network. In *CVPR*, 2019. 1, 2, 3, 6, 7

[41] Wenhai Wang, Enze Xie, Xiaoge Song, Yuhang Zang, Wenjia Wang, Tong Lu, Gang Yu, and Chunhua Shen. Efficient and accurate arbitrary-shaped text detection with pixel aggregation network. In *ICCV*, 2019. 1, 6, 7

[42] Yuxin Wang, Hongtao Xie, Zheng-Jun Zha, Mengting Xing, Zilong Fu, and Yongdong Zhang. Contournet: Taking a further step toward accurate arbitrary-shaped scene text detection. In *CVPR*, 2020. 6, 8

[43] Shanyu Xiao, Liangrui Peng, Ruijie Yan, Keyu An, Gang Yao, and Jaesik Min. Sequential deformation for accurate scene text detection. In *ECCV*, 2020. 7

[44] Yongchao Xu, Yukang Wang, Wei Zhou, Yongpan Wang, Zhibo Yang, and Xiang Bai. Textfield: Learning a deep direction field for irregular scene text detection. *TIP*, 28(11):5566–5579, 2019. 6

[45] Xue Yang, Junchi Yan, Qi Ming, Wentao Wang, Xiaopeng Zhang, and Qi Tian. Rethinking rotated object detection with gaussian wasserstein distance loss. In *ICML*, 2021. 5

[46] Cong Yao, Xiang Bai, and Wenyu Liu. A unified framework for multioriented text detection and recognition. *TIP*, 23(11):4737–4749, 2014. 6

[47] Cong Yao, Xiang Bai, Wenyu Liu, Yi Ma, and Zhuowen Tu. Detecting texts of arbitrary orientations in natural images. In *CVPR*, 2012. 2, 6

[48] Liu Yuliang, Jin Lianwen, Zhang Shuaitao, and Zhang Sheng. Detecting curve text in the wild: New dataset and new solution. *arXiv preprint arXiv:1712.02170*, 2017. 2, 6

[49] Shi-Xue Zhang, Xiaobin Zhu, Jie-Bo Hou, Chang Liu, Chun Yang, Hongfa Wang, and Xu-Cheng Yin. Deep relational reasoning graph network for arbitrary shape text detection. In *CVPR*, 2020. 1, 2, 6, 7, 8

[50] Shi-Xue Zhang, Xiaobin Zhu, Chun Yang, Hongfa Wang, and Xu-Cheng Yin. Adaptive boundary proposal network for arbitrary shape text detection. In *ICCV*, 2021. 6

[51] Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, and Jiajun Liang. East: an efficient and accurate scene text detector. In *CVPR*, 2017. 2

[52] Zhi-Hua Zhou, Jianxin Wu, and Wei Tang. Ensembling neural networks: many could be better than all. *Artificial intelligence*, 137(1-2):239–263, 2002. 3

[53] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *ICLR*, 2021. 1, 2, 3, 7, 8

[54] Yiqin Zhu, Jianyong Chen, Lingyu Liang, Zhanghui Kuang, Lianwen Jin, and Wayne Zhang. Fourier contour embedding for arbitrary-shaped text detection. In *CVPR*, 2021. 2