Text Spotting Transformers

Xiang Zhang[1] Yongwen Su[2] Subarna Tripathi[3] Zhuowen Tu[1]
[1]UC San Diego [2]Shanghai Jiao Tong University [3]Intel Labs
{xiz102, ztu}@ucsd.edu, heyue2001@gmail.com, subarna.tripathi@intel.com

Abstract

In this paper, we present TExt Spotting TRansformers (TESTR), a generic end-to-end text spotting framework using Transformers for text detection and recognition in the wild. TESTR builds upon a single encoder and dual decoders for the joint text-box control point regression and character recognition. Other than most existing literature, our method is free from Region-of-Interest operations and heuristics-driven post-processing procedures; TESTR is particularly effective when dealing with curved text-boxes where special cares are needed for the adaptation of the traditional bounding-box representations. We show our canonical representation of control points suitable for text instances in both Bezier curve and polygon annotations. In addition, we design a bounding-box guided polygon detection (box-to-polygon) process. Experiments on curved and arbitrarily shaped datasets demonstrate state-of-the-art performances of the proposed TESTR algorithm.

1. Introduction

Text detection and recognition in natural scenes, called text spotting, is an active area of research in computer vision [11, 15, 23, 24, 29, 33, 38]. Text spotting is of great importance in real-world applications such as mapping, autonomous driving, and image retrieval. The text spotting problem typically consists of two sub-tasks: 1) text detection that localizes text boxes in a natural image, and 2) text recognition that reads the characters from the detected text. Despite its practical significance and a steady progress observed recently, text spotting remains a challenging problem that requires further improvement. The main difficulty in text spotting is contributed by multiple factors including large variations in font, size, style, color, shape, occlusion, distortion, and layout for natural scene images.

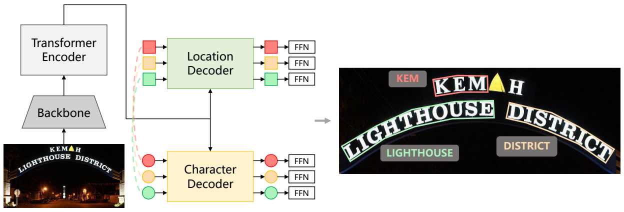

Figure 1. Illustration of the overall TESTR pipeline. The input image is passed through a feature backbone and Transformer encoder, and the multi-scale feature is shared across the location and character decoder, which predict the coordinates of control points and characters of the text instance respectively. The canonical representation of control points serves both polygon vertices and Bezier curve control points.

Classical text spotting methods [24, 38] often perform text detection and recognition in two separate steps. In the detection module, the regions of interest are proposed for text instance detection. After alignment, the features are then used in the text recognition module. In natural scenes, text-boxes often appear in arbitrary orientations [50] and are non-rectangular [24]. This poses further challenges for the algorithm development that typically requires a number of heuristics designs with intermediate and post-processing steps [11, 15, 29, 34].

Transformers [43] have achieved a remarkable success in natural language processing [4] and computer vision [6]. DEtection TRansformers (DETR) [2] have also made a profound impact to object detection by removing the proposal anchors and the non-maximum suppression processes needed in the sliding window based approaches [36]. LETR [49] extends DETR by adopting Transformers to directly detect geometric structures such as line segments beyond the bounding box representation.

Inspired by the DETR family models [2, 16, 49, 52], we propose TExt Spotting TRansformers (TESTR), a Transformer-based text spotting method that performs text detection and recognition in a unified framework. TESTR avoids the heuristics design and the intermediate stages required in many of the existing text spotting approaches.

The contribution of TESTR is listed as follows.

- We propose a single-encoder dual-decoder framework that jointly performs curved text instance detection and recognition using **Transformers beyond the stan-**

Code available at https://github.com/mlpc-ucsd/TESTR.

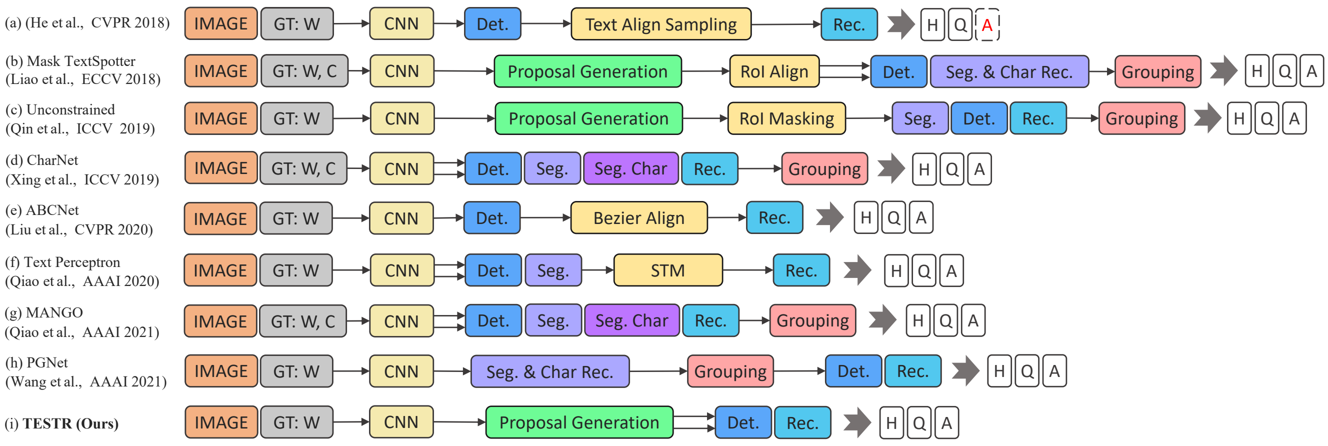

Figure 2. Overview of some end-to-end scene text spotting methods that are most relevant to ours. Inside the GT (ground-truth) box, 'W' and 'C' represent word-level annotation and character-level annotation. The 'H', 'Q', and 'A' represent that the method is able to detect horizontal, quadrilateral, and arbitrarily-shaped text, respectively. The dashed box represents the shape of the text which the method is unable to detect. Figure style from [24, 46].

dard bounding box representation. Our method, thanks to direct regression of the control points coordinates, is a holistic approach that requires neither heuristics-driven post-processing procedures, nor Region-of-Interest operations.

- We introduce a **box-to-polygon process** that achieves bounding-box guided polygon detection in the detection Transformers. Experimental results show an apparent performance boost.
- The canonical representation of control points makes our method appropriate for **both the polygonal and Bezier curve** annotations. TESTR achieves state-of-the-art performances on challenging datasets, *i.e.* Total-Text and CTW1500.

1.1. Related Works

Scene text spotting consists of text detection and recognition. Two-stage approaches are first developed to address the task, which train detection and recognition modules separately and simply join them during inference. Recent literature focuses on end-to-end methods, which tackles detection and recognition simultaneously through RoI operations during training. While these methods demonstrate satisfying performance, the text spotting task still remains a challenge due to the prevalence of arbitrarily-shaped texts. We will discuss related works from the perspectives of text detection, text recognition, regular text spotting, and arbitrarily-shaped text spotting. Figure 2 is an overview of exemplary works.

Text Detection. Early works [7, 19, 42] focus on horizontal text detection, which predict rectangular bounding boxes for the text instances. They pose apparent limitations as texts in the wild are mostly multi-oriented quadrilateral, curved, or even arbitrarily shaped. Efforts have been made to address these challenging cases. [18, 51] use both rotated boxes and quadrangles to achieve multi-oriented quadrilateral text detection. [1] enables the detection of arbitrarily-shaped texts through the prediction of character boxes. While achieving a dramatic performance boost, it requires expensive character-level annotations and post-processing to group the detected characters back to texts. [47] uses pairwise point representation for text regions, yet it is restricted to the sequential decoding of RNNs. [24, 26] introduce a novel Bezier curve representation of the curved texts, and significantly improve the detection performance on them.

Text Recognition. Classical works [30,32,45] adopt statistical approaches to classify characters and group them into words. Deep-learning based methods [14, 40] have ushered a new era for text recognition. CRNN [38] integrates CNN and RNN to perform text recognition. However, it is mainly applicable to regular texts and limited as to arbitrarily-shaped texts. [22, 39] use a spatial transformer to convert irregular texts into rectangular shapes, and then feed them into the feature extractor and sequence decoder for recognition.

Regular End-to-end Scene Text Spotting. To further enhance the performance of text spotting, [15] proposes an end-to-end trainable text spotting framework. RoI Pooling is introduced to bridge the gap between text detection and recognition. However, this method is limited to horizontal texts. Other literature conducts quadrilateral text spotting based on other specially crafted RoI operations, such as Text-Align [11] and RoI-Rotate [23], while remaining incapable of spotting arbitrarily-shaped texts.

Arbitrarily-shaped End-to-end Scene Text Spotting. In [41], quadrangle text region proposals are generated, followed by an RoI transform. While this method can rec-

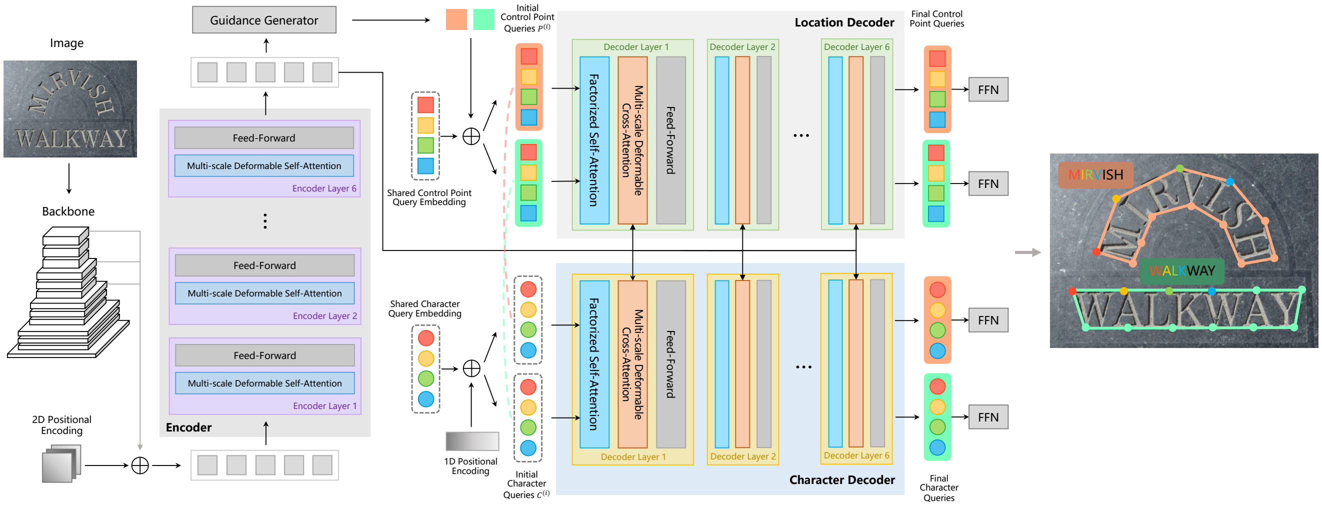
Image
Backbone
2D Positional Encoding
Guidance Generator
Feed-Forward
Multi-scale Deformable Self-Attention
Encoder Layer 6
Feed-Forward
Multi-scale Deformable Self-Attention
Encoder Layer 2
Feed-Forward
Multi-scale Deformable Self-Attention
Encoder Layer 1
Encoder
Initial Control Point Queries $P^{(i)}$
Shared Control Point Query Embedding
Location Decoder
Decoder Layer 1
Factorized Self-Attention
Multi-scale Deformable Cross-Attention
Feed-Forward
Decoder Layer 2
Decoder Layer 6
Final Control Point Queries
FFN
FFN
Shared Character Query Embedding
1D Positional Encoding
Initial Character Queries $C^{(i)}$
Factorized Self-Attention
Multi-scale Deformable Cross-Attention
Feed-Forward
Decoder Layer 1
Decoder Layer 2
Decoder Layer 6
Character Decoder
FFN
FFN
Final Character Queries
MIRVISH
WALKWAY

Figure 3. Overall architecture of Text Spotting Transformers (TESTR). First, the encoder performs multi-scale deformable self-attention across feature maps, and a guidance generator produces coarse bounding boxes from the features. These boxes are encoded and added on top of the learnable control point query embeddings to guide the learning of control points. Control point queries are fed through the location decoder and feed-forward networks (FFNs) to predict their coordinates. The character decoder, with shared reference points as the location decoder for the multi-scale cross-attention, predicts characters for the corresponding text instance. The framework is end-to-end trainable and performs detection and recognition in a unified way. Note that the control point and character queries with identical background color belong to the same text instance in the output image.

ognize irregular texts, its quadrilateral representation is not optimal for arbitrarily-shaped text regions. CharNet [48] performs character and text detection in a single pass, requiring character-level annotations. TextDragon [8] generates multiple local quadrangles around the text centerline, with RoISlide operation for feature warping and aggregation within the text instance. Though not requiring character-level supervision, it still needs to perform centerline detection, grouping, and sorting to convert local quadrangles to text boundaries.

Other literature focuses on segmentation-based methods for arbitrarily-shaped text spotting. Mask TextSpotter [29], built on Mask R-CNN [9], performs text- and character-level segmentation, requiring further grouping before getting final results. [35] proposes RoI masking that multiplies segmentation probability maps with features to suppress the background, whereas [17] uses binary maps to mitigate the inaccuracies in segmentation. While these approaches achieve fair performance, the mask representation is subject to post-processing such as polygon fitting and smoothing to obtain desirable boundaries. MANGO [33] develops Mask Attention module to retain global features for multiple instances, yet it still requires centerline segmentation to guide the grouping of the predictions.

Recent works try to develop appropriate representations that directly capture the text boundaries. ABCNet [24] and ABCNet v2 [26] introduce parametric Bezier curve representations for curved texts, and develop Bezier-Align for feature extracting. However, low-order Bezier curves exhibit limitations when representing heavily curved or wavy text shapes. [34] uses Shape Transform Module to generate fiducial points around the text boundaries and rectify irregular texts. PGNet [46] transforms the polygonal text boundaries to the centerline, border offset, and direction offset and perform multi-task learning for these objectives. While eliminating RoI operations, it still uses a specially designed polygon restoration process.

In contrast, our approach relies solely on Transformers, which is entirely free from RoI operations. With the outputs being coordinates of polygon vertices or Bezier control points for the text instance, along with the corresponding character sequence, no special post-processing is needed.

2. Method

TExt Spotting TRansformers (TESTR) is an end-to-end trainable framework that handles text detection and recognition in a unified manner. The overall architecture is shown in Figure 3. We first introduce the multi-scale deformable attention as in Deformable DETR [52], and elaborate on the key components of our model – dual decoders for detection and recognition, and box-to-polygon detection procedure.

2.1. Multi-Scale Deformable Attention

One obstacle for the text spotting task is the prevalence of small text instances in the images. Current literature tries to overcome this limitation by leveraging multi-scale feature maps, such as Feature Pyramid Network (FPN) [20]. To utilize such feature maps, we take the multi-scale deformable attention module in [52]. Given a set of L level multi-scale feature maps $\{U_l\}_{l=1}^{L}$, with each level as $U_l \in \mathbb{R}^{C\times H_l\times W_l}$, and $\mathbf{p}(q)$ as the normalized coordinates of the reference point for the query q, the multi-scale deformable attention can be expressed as

$$\begin{aligned}&\mathrm{MSDeformAttn}(q,\ \mathbf{p}(q),\ \{U_l\}_{l=1}^{L}) = \\ &\sum_{h=1}^{H}\mathbf{W}_h\left\{\sum_{l=1}^{L}\sum_{k=1}^{K}\mathbf{A}_{hlk}(q)\cdot\mathbf{W}'_h U_l\left[\phi_l\left(\mathbf{p}(q)\right)+\Delta\mathbf{p}_{hlk}(q)\right]\right\}\end{aligned} \tag{1}$$

where $h,\ l,\ k$ are indices for the attention head, input feature level, and sampling point respectively. $\mathbf{A}_{hlk}$ denotes the attention weight for query q, normalized with respect to K sampling points. ϕ_l maps the normalized coordinates to the scale of l-th level feature map, and $\Delta\mathbf{p}$ generates an appropriate sampling offset for the query. Both of them are added to form the sampling location for the feature map U_l. $\mathbf{W}'_h$ and $\mathbf{W}_h$ are trainable weight matrices that are similar to those present in the original multi-head attention.

Instead of relying on the original attention, which requires sampling of $H\times W$ points in the feature map, multi-scale deformable attention samples LK points, largely reducing computational overheads and enabling the capability to use multi-scale feature maps. We will illustrate its efficiency in the experiment section.

2.2. Dual Decoders

We formulate the holistic text spotting task as a set prediction problem. Given an image I, we need to output a set of point-character tuples, defined as $Y = \left\{\left(\mathbf{P}^{(i)},\ \mathbf{C}^{(i)}\right)\right\}_{i=1}^{K}$, where i is the index for each text instance, $\mathbf{P}^{(i)} = (p_1^{(i)},\ \cdots,\ p_N^{(i)})$ is the coordinates of N control points, and $\mathbf{C}^{(i)} = (c_1^{(i)},\ \cdots,\ c_M^{(i)})$ is the M characters of the text.

To tackle this problem, we propose a dual-decoder paradigm for predictions of different modalities, location decoder for detection (to predict $\mathbf{P}^{(i)}$) and character decoder for recognition (to predict $\mathbf{C}^{(i)}$).

Location decoder. We extend the queries in original DETR [2] to composite queries for predicting multiple control points for each instance. We have Q such queries, each corresponding to a text instance, as $\mathbf{P}^{(i)}$. Each query element is composed of subqueries p_j, where $\mathbf{P}^{(i)} = (p_1^{(i)},\ \cdots,\ p_N^{(i)})$. To capture the relationship between different text instances and between different subqueries within a single text instance in a structural way, we utilize factorized self-attention, inspired by [5]. The factorized self-attention is composed of an intra-group attention, which is a self-attention within subqueries belonging to each of the $\mathbf{P}^{(i)}$, and an inter-group attention, which is a self-attention across p_j of different queries.

The initial control point queries are fed into the location decoder. After the process of multi-layer decoding, the final control point queries are taken by a classification head predicting the confidence, and a 2-channel regression head outputting the normalized coordinates for each control point.

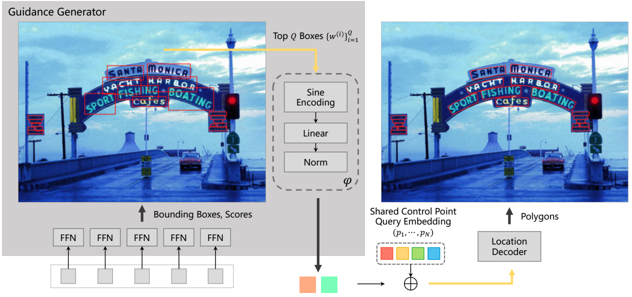

Figure 4. Illustration of the box-to-polygon detection process. The guidance generator predicts coarse bounding boxes and scores as shown in the left image. The coordinates of top Q boxes are fed to the differentiable encoding module φ and the encoded results are added to the shared control point query embeddings, which are taken by the location decoder for the final polygonal predictions.

The control points predicted here can either be N polygon vertices, or control points for Bezier curves, as in [24]. For the polygon points, we use the sequence that starts with the top left corner and moves in the clockwise order.

For the Bezier control points, Bernstein Polynomials [27] can be used to construct the parametric curve

$$c(t) = \sum_{j=1}^{N} p_j B_{(j-1),(N-1)}(t),\quad t\in[0,\ 1] \tag{2}$$

where Bernstein basis polynomials are defined as

$$B_{i,n}(t) = \binom{n}{i} t^i (1-t)^{n-i},\quad i = 0,\ 1,\ \cdots,\ n \tag{3}$$

Following [24], we use two cubic Bezier curves for a single text instance, corresponding to the two possibly curved sides of the text. One can sample across t to convert Bezier curves back to polygons.

Character decoder. The character decoder follows most of the location decoder, with control point queries replaced by character queries $\mathbf{C}^{(i)}$. The initial character queries comprise a learnable query embedding and 1D sine positional encoding, and are shared across different text instances. The character query $\mathbf{C}^{(i)}$ and control point query $\mathbf{P}^{(i)}$ with the same index belong to the same text instance, and therefore the reference points of the multi-scale deformable cross-attention are shared to ensure they get the identical contexts from the image feature. A classification head takes the final character queries to predict among multiple character classes.

2.3. Box-to-Polygon Detection Process

The decoder models the Bayesian inference process $P(Y|I) \propto P(I|Y)P(Y)$ for our set prediction problem, where $P(I|Y)$ captures the relationship between hypotheses (queries) and input I through cross-attention, while $P(Y)$ models the prior on configuration of Y through self-attention. We argue that when Y is complex, in our case of composite queries, $P(Y)$ is hard to learn. Hence, we propose a box-to-polygon detection approach, which takes the bounding boxes of text instances and uses them to guide the polygon detection. This process, employing information related to concrete image I to form input-specific priors, facilitates the training of polygon control point regression.

The framework begins with the guidance generator in Figure 3, which is a proposal generator outputting coarse bounding box coordinates and probabilities. Boxes with top-Q probabilities are selected and their coordinates are denoted as $\{w^{(i)}\}_{i=1}^{Q}$. The initial control point queries described in 2.2 are formed by:

$$\mathbf{P}^{(i)} = \varphi(w^{(i)}) + (p_1, \ \cdots , \ p_N) \tag{4}$$

where $(p_1, \ \cdots , \ p_N)$ is the control point query embedding, shared across Q queries, modeling the general relation between control points that is irrelevant to the specific bounding box location. φ is the sine positional encoding function followed by a linear and normalization layer, and therefore is fully differentiable. $\varphi(w^{(i)})$, as the encoded bounding box information, is shared across N subqueries within a single instance, modeling the overall location and scale of the text instance. $w^{(i)}$ is also used as the initial reference point for the multi-scale deformable cross-attention.

An illustration of this process is provided in Figure 4 with details of the guidance generator. Ablation studies in Section 3.4 demonstrate the significant improvement in the recognition accuracy brought by this process.

2.4. Training Losses

Bipartite matching. Since TESTR outputs a fixed number of predictions unlike the actual number (G) of ground truth instances, we need to find an optimal matching between them to calculate the loss. Specifically, we need to find an injective function $\sigma : [G] \mapsto [Q]$ that minimizes the following matching cost $\mathcal{C}$:

$$\arg\min_{\sigma} \sum_{i=1}^{G} \mathcal{C}(Y^{(i)}, \ \hat{Y}^{(\sigma(i))}) \tag{5}$$

where $\hat{Y}^{(j)} = (\hat{\mathbf{P}}^{(j)}, \ \hat{\mathbf{C}}^{(j)})$ is the prediction to be matched and $Y^{(i)}$ is the ground truth. For simplicity, we use the control point location to guide the learning of character decoding. Therefore, the matching cost is defined as a mixture of confidence and coordinate deviation. For i-th ground truth and its matched $\sigma(i)$-th query, the cost function is

$$\mathcal{C}(Y^{(i)}, \hat{Y}^{(\sigma(i))}) = \lambda_{\text{cls}} \text{FL}'(\hat{b}^{(\sigma(i))}) + \lambda_{\text{coord}} \textstyle\sum_{k=1}^{N} \left\| p_k^{(i)} - \hat{p}_k^{(\sigma(i))} \right\| \tag{6}$$

where $\hat{b}^{(\sigma(i))}$ is the probability for the only instance class – text, which also serves as the confidence score. FL$'$ is derived from the focal loss [21], and is defined as the difference of the positive and negative term: $\text{FL}'(x) = -\alpha(1-x)^{\gamma}\log(x) + (1-\alpha)x^{\gamma}\log(1-x)$. The second term in Equation 6 is the L-1 distance between ground truth and predicted control point coordinates.

The problem in 5 can be efficiently solved by the Hungarian algorithm [13]. We use the same bipartite matching scheme to match proposals in the guidance generator with ground truth boxes, which are bounding boxes for the control points.

Instance classification loss. We adopt focal loss as the classification loss of text instances. For the j-th query, the loss is defined as:

$$\begin{aligned} \mathcal{L}_{\text{cls}}^{(j)} = &- \mathbb{1}_{\{j \in \text{Im}(\sigma)\}} \alpha (1 - \hat{b}^{(j)})^{\gamma} \log(\hat{b}^{(j)}) \\ &- \mathbb{1}_{\{j \notin \text{Im}(\sigma)\}} (1-\alpha)(\hat{b}^{(j)})^{\gamma} \log(1 - \hat{b}^{(j)}) \end{aligned} \tag{7}$$

where $\text{Im}(\sigma)$ is the image of the mapping σ.

Control point loss. L-1 distance loss is used for control point coordinate regression:

$$\mathcal{L}_{\text{coord}}^{(j)} = \mathbb{1}_{\{j \in \text{Im}(\sigma)\}} \sum_{i=1}^{N} \left\| p_i^{(\sigma^{-1}(j))} - \hat{p}_i^{(j)} \right\| \tag{8}$$

Character classification loss. We deem the character recognition as a classification problem, where each class is assigned a specific character. Cross entropy loss is used here:

$$\mathcal{L}_{\text{char}}^{(j)} = \mathbb{1}_{\{j \in \text{Im}(\sigma)\}} \sum_{i=1}^{M} \left(-c_i^{(\sigma^{-1}(j))} \log \hat{c}_i^{(j)} \right) \tag{9}$$

The loss function for the dual decoders comprises the three aforementioned losses:

$$\mathcal{L}_{\text{dec}} = \sum_{j} \left(\lambda_{\text{cls}} \mathcal{L}_{\text{cls}}^{(j)} + \lambda_{\text{coord}} \mathcal{L}_{\text{coord}}^{(j)} + \lambda_{\text{char}} \mathcal{L}_{\text{char}}^{(j)} \right) \tag{10}$$

Bounding box intermediate supervision loss. To make the proposals in Section 2.3 more accurate, we also introduce intermediate supervision for them at the encoder side. The same bipartite matching scheme is used to match these bounding box proposals to the ground truth. We denote the matching here as σ', and the overall loss here is

$$\mathcal{L}_{\text{enc}} = \sum_{i} \left(\lambda_{\text{cls}} \mathcal{L}_{\text{cls}}^{(i)} + \lambda_{\text{coord}} \mathcal{L}_{\text{coord}}^{(i)} + \lambda_{\text{gIoU}} \mathcal{L}_{\text{gIoU}}^{(i)} \right) \tag{11}$$

where the classification loss $\mathcal{L}_{\text{cls}}^{(i)}$ and control point loss $\mathcal{L}_{\text{coord}}^{(i)}$ are identical to the ones used for polygon detection, except for the different matching σ' used. $\mathcal{L}_{\text{gIoU}}$ is the generalized IoU loss defined in [37] for bounding box regression.

The final loss for the entire model is simply the sum of the encoder and decoder loss.

3. Experiments

3.1. Datasets

Here we briefly introduce the datasets used in this paper.

SynthText 150k. Unlike existing *SynthText 800k* which contains mostly straight texts in quadrilateral annotations, SynthText 150k synthesized in [24] comes with 94,723 images containing mostly straight text and 54,327 with major curved texts in Bezier annotations.

ICDAR 2015. The ICDAR 2015 [12] is the official dataset for ICDAR 2015 Robust Reading Competition. It contains 1000 training images and 500 testing images, with horizontal and perspective texts with quadrilateral box annotation. The images were captured with hand-held cameras in the wild, therefore blurs and obscurities are frequent.

Total-Text. The Total-Text [3] is a popular curved text benchmark, with 1255 images for training and 300 for testing. Word-level polygon or Bezier annotations are used.

CTW1500. [25] is another important curved scene text benchmark, with 1000 training images and 500 testing images. Different from Total-Text, it contains both English and Chinese texts. As the proportion of Chinese texts is small, we ignore them during training.

We follow the standard evaluation protocols used in these datasets, which involve the calculation of IoU between predicted and ground truth polygons. The output of TESTR with Bezier annotations is converted back to polygons prior to evaluation.

3.2. Implementation Details

We use ResNet-50 [10] as the feature backbone for all the experiments. Multi-scale feature maps are directly drawn from the last three stages of ResNet without FPN. The parameters for the deformable Transformers are similar to [52], with $H = 8$ heads and $K = 4$ sampling points for the deformable attentions, and we use 6 layers of encoders and decoders.

Data augmentation. The data augmentation during training is conducted by 1) random resize with the shorter edge ranging from 480 to 896, and the longest edge kept within 1600; 2) instance-aware random crop, which ensures the cropped size larger than half of the original size and no texts being cut. During test time, we resize the shorter edge to 1600 while keeping the longest edge within 1892.

Pre-training. The model is pretrained on a mixture of SynthText 150k, MLT 2017 [31] and TotalText for 440k iterations. The base learning rate for the polygon variant is 1×10^{-4} and is decayed at the 340k-th iteration by a factor of 0.1. Learning rates are scaled by a factor of 0.1 for the linear projections used to predict reference points, sampling offsets of the multi-scale deformable attention and feature backbone. AdamW [28] is used as the optimizer, with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and weight decay of 10^{-4}. We use $Q = 100$ composite queries. The max text length M is 25, and number of polygon control points N is 16. The weighting factors for the losses are $\lambda_{\text{cls}} = 2.0$, $\lambda_{\text{coord}} = 5.0$, $\lambda_{\text{char}} = 4.0$, $\lambda_{\text{gIoU}} = 2.0$. We set $\alpha = 0.25$, $\gamma = 2.0$ for the focal loss. For the Bezier variant of the model, we have $N = 8$ control points, double the value of base learning rate, and half λ_{char} for the purpose of balancing. The pretraining process takes about 3 days on 8 RTX 2080Ti GPUs with the image batch size of 8.

Finetuning. The model is finetuned on specific datasets prior to evaluation to mitigate the variance across different datasets. For the Total-Text and ICDAR 2015 dataset, we finetune the model for 20k iterations, with the base learning rate scaled by 0.1. For CTW1500, to address the longer texts present in the dataset, the maximum text length M is set to 100, and therefore the model is finetuned for 200k iterations, larger than the ones needed for the other two.

3.3. Results

Here we present the benchmark of our model TESTR in polygonal or Bezier curve annotations.

Irregular texts. We test our method on two irregular text benchmarks: Total-Text and CTW1500, and the quantitative results are shown in Table 1 and 2.

In terms of text detection, the TESTR-Beizer outperforms the previous most accurate model by 1.0% on the F-score metric on the Total-Text dataset. The TESTR-Polygon has almost the same detection accuracy as ABCNet v2 and is free of Bezier annotations. On the CTW-1500 dataset, the F-score of TESTR surpasses that of ABCNet v2 by a large margin, with 1.6% for Bezier and 2.4% for polygonal annotations.

In the case of end-to-end text spotting, TESTR-Polygon significantly surpasses the best-reported results by 2.8% when equipped with full lexicons on CTW1500. On Total-Text, our method outperforms the previous best results by 0.4% without lexicons and by 0.3% with full lexicons.

Qualitative results on the two datasets are shown in the Figure 5 and 6. The results illustrate our method can handle both straight and curved texts well. The failure cases for TESTR with Bezier annotations are displayed, *e.g.* the last column of Figure 1, where it fails to generate the correct bounding polygon for the Bezier curves, while the polygon model variant succeeds. This observation is consistent with the quantitative results.

In summary, the results on Total-Text and CTW1500 demonstrate the effectiveness of our method for arbitrarily-shaped text spotting. Meanwhile, the overall performance of TESTR-Polygon is better than TESTR-Bezier mostly.

Regular texts. We evaluate our method on ICDAR2015 containing many perspective texts annotated with quadrilateral bounding boxes, and the results are shown in Table 3. In the detection stage, our method achieves state-of-the-art F-score. In the end-to-end text spotting, our method exhibits

Table 1. Scene text spotting results on Total-Text. “None” refers to recognition without lexicon. “Full” lexicon contains all the words in the test set.

Method	Backbone	Detection			End-to-End		FPS
		P	R	F	None	Full	
FOTS [23]	ResNet-50	52.3	38.0	44.0	32.2	–	–
Textboxes [19]	ResNet-50-FPN	62.1	45.5	52.5	36.3	48.9	1.4
Mask TextSpotter [29]	ResNet-50-FPN	69.0	55.0	61.3	52.9	71.8	4.8
CharNet [48]	ResNet-50-Hourglass57	87.3	85.0	86.1	66.2	–	1.2
Text Dragon [8]	VGG16	85.6	75.7	80.3	48.8	74.8	–
Boundary TextSpotter [44]	ResNet-50-FPN	88.9	85.0	87.0	65.0	76.1	–
Unconstrained [35]	ResNet-50-MSF	83.3	83.4	83.3	67.8	–	–
Text Perceptron [34]	ResNet-50-FPN	88.8	81.8	85.2	69.7	78.3	–
Mask TextSpotter v3 [17]	ResNet-50-FPN	–	–	–	71.2	78.4	–
ABCNet-MS [24]	ResNet-50-FPN	–	–	–	69.5	78.4	6.9
ABCNet v2 [26]	ResNet-50-FPN	90.2	84.1	87.0	70.4	78.1	10
MANGO [33]	ResNet-50-FPN	–	–	–	72.9	83.6	4.3
PGNet [46]	ResNet-50-FPN	85.5	**86.8**	86.1	63.1	–	35.5
TESTR-Bezier (ours)	ResNet-50	92.8	83.7	**88.0**	71.6	83.3	5.5
TESTR-Polygon (ours)	ResNet-50	**93.4**	81.4	86.9	**73.3**	**83.9**	5.3

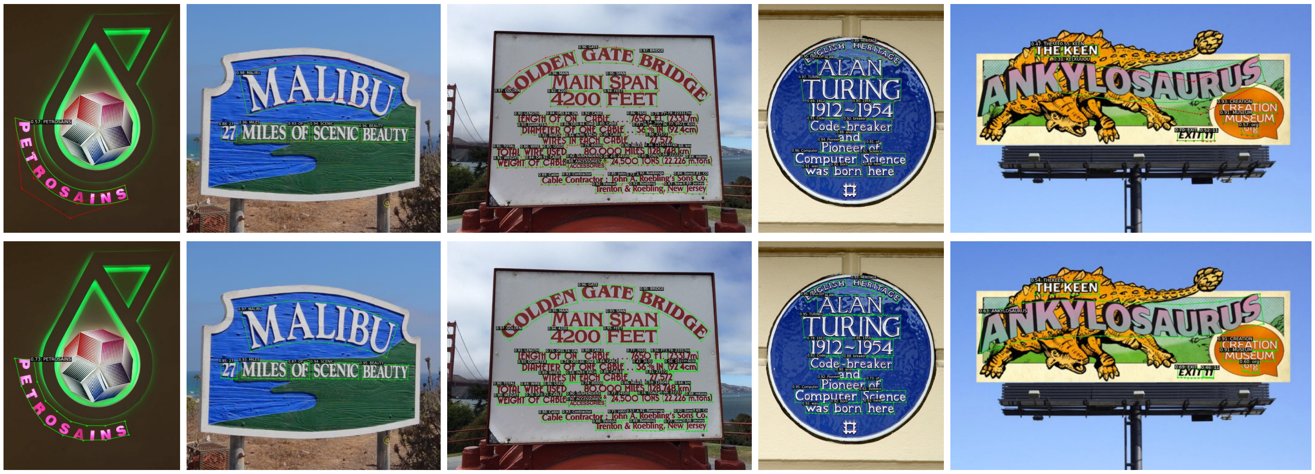

Figure 5. Qualitative results on Total-Text without lexicons. Top row: Bezier; bottom row: polygon annotations. The predictions are shown in green contours, with Bezier control points in red. The number before text is the confidence score. TESTR-Bezier fails to capture the shape of the “ANKYLOSAURUS” text in the last column, while the polygon variant succeeds. Zoom in for better visualization.

Table 2. End-to-end text spotting results on CTW1500. “None” represents lexicon-free, while “Full” indicates all the words in the test set are used.

Method	Detection			End-to-End	
	P	R	F	None	Full
Text Dragon [8]	84.5	82.8	83.6	39.7	72.4
Text Perceptron [34]	87.5	81.9	84.6	57.0	–
ABCNet [24]	–	–	–	45.2	74.1
ABCNet v2 [26]	85.6	**83.8**	84.7	57.5	77.2
MANGO [33]	–	–	–	**58.9**	78.7
TESTR-Bezier (ours)	89.7	83.1	86.3	53.3	79.9
TESTR-Polygon (ours)	**92.0**	82.6	**87.1**	56.0	**81.5**

remarkable performance in the lexicon-free setting, on par with Text Perceptron with generic lexicons. When lexicons are available, TESTR works best with the “Strong” type, obtaining competitive results compared with other methods. Qualitative results in the right column of Figure 6 show our method can recognize texts even in occluded scenes or from extreme viewing angles.

3.4. Ablation Studies

To illustrate the effectiveness of the proposed components, we conduct multiple ablation studies on Total-Text with polygonal annotations.

Box-to-polygon detection process. In our design of TESTR, the encoder performs multi-scale self-attention across feature maps, and a guidance generator produces coarse bounding boxes from the encoded features. These bounding boxes, encoded and added on top of the learnable control point query embeddings, are used to guide the learning of control point regression in the location decoder. We ablate this module by replacing the $\varphi(w^{(i)})$ term in Equation 4 with a learnable embedding vector to show how the bounding box guidance affects the results. The results

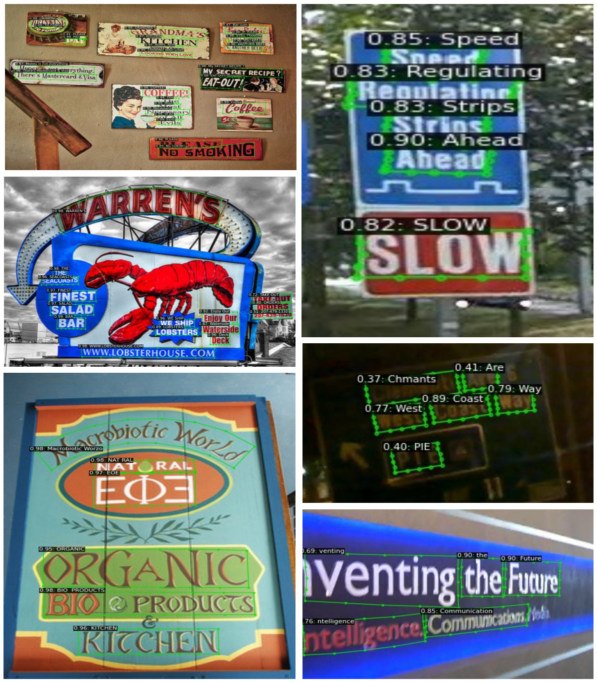

Figure 6. Qualitative results of TESTR on CTW1500 (left column) and ICDAR (right column) using polygonal annotations.

Table 3. Results on ICDAR 2015 dataset. “S”, “W”, “G”, “N” represent recognition with “Strong”, “Weak”, “Generic” or “None” lexicon respectively.

Method	Detection			End-to-End			
	P	R	F	S	W	G	N
He *et al.* [11]	87.0	86.0	87.0	82.0	77.0	63.0	–
TextNet [41]	89.4	85.4	87.4	78.7	74.9	60.5	–
FOTS [23]	91.0	85.2	88.0	81.1	75.9	60.8	–
CharNet R-50 [48]	91.2	88.3	89.7	80.1	74.5	62.2	60.7
Boundary TextSpotter [44]	89.8	87.5	88.6	79.7	75.2	64.1	–
Unconstrained [35]	89.4	85.8	87.5	83.4	**79.9**	68.0	–
Text Perceptron [34]	**92.3**	82.5	87.1	80.5	76.6	65.1	–
Mask TextSpotter v3 [17]	–	–	–	83.3	78.1	**74.2**	–
ABCNet v2 [26]	90.4	86.0	88.1	82.7	78.5	73.0	–
MANGO [33]	–	–	–	81.8	78.9	67.3	–
PGNet [46]	91.8	84.8	88.2	83.3	78.3	63.5	–
TESTR-Polygon (ours)	90.3	**89.7**	**90.0**	**85.2**	79.4	73.6	**65.3**

shown in Table 4 demonstrate that the box-to-polygon detection process could improve Precision, Recall, F-score by 0.5%, 3.6% and 2.2% in detection respectively, and significantly improve the end-to-end recognition results by 5.8%.

Multi-scale feature. Our method leverages multi-scale feature maps to overcome the challenge of the prevalent small text instances in the images. We conduct ablations by using only the feature map from the last stage of ResNet. Table 4 shows that adopting multi-scale features could improve Precision, Recall, F-score by 1.2%, 2.3% and 1.8% in detection respectively, and dramatically improve the end-to-end results by 10.8%. This indicates the text recognition task benefits much from features with larger scales.

Table 4. Ablation study on Total-Text using TESTR with polygonal output.

Multi-scale Features	Box Guidance	Detection			E2E
		P	R	F	
–	✓	92.2	79.1	85.1	62.5
✓	–	92.9	77.8	84.7	67.5
✓	✓	**93.4**	**81.4**	**86.9**	**73.3**

Input scale. To demonstrate the tradeoffs between speed and accuracy, we evaluate our model with the shorter side of the image resized to 720, 1000, 1280, 1600 respectively. The results are shown in Table 5. The F-score of both detection and end-to-end recognition increases with FPS decreasing as the input scale grows larger.

Table 5. Performance of TESTR with different input scales on Total-Text.

Model Type	Input	Detection			E2E	FPS
		P	R	F		
Bezier	720	91.5	81.5	86.2	62.6	11.6
	1000	91.2	**84.1**	87.5	69.4	7.9
	1280	92.3	83.7	87.8	70.9	5.8
	1600	**92.8**	83.7	**88.0**	**71.6**	5.5
Polygon	720	92.7	79.7	85.7	66.2	11.7
	1000	92.1	81.4	86.4	70.5	8.0
	1280	92.5	**81.5**	86.7	72.2	6.0
	1600	**93.4**	81.4	**86.9**	**73.3**	5.3

4. Discussions

Limitations and future work In our setting of TESTR, we assume a fixed number of polygon control points, which might not be optimal. For most perspective texts, quadrilaterals would be sufficient, while many more vertices would be required if texts come with higher curvature. In the future, we would like to investigate methods that adaptively determine the adequate number of polygon control points within our framework to better capture their shapes.

Conclusions In this paper, we have presented TESTR, a text spotting framework based on single-encoder dual-decoder Transformer architecture. By modeling the text detection and recognition in a holistic fashion, our model directly performs set prediction without heuristics-driven post-processing or Region-of-Interest operations. A bounding-box guided polygon detection procedure allows efficient detection of arbitrarily-shaped texts. In addition, our canonical representation of control points enables the model to function effectively for both polygonal and Bezier annotations. Experimental results on challenging curved or oriented text benchmarks, Total-Text and CTW1500, demonstrate the state-of-the-art performance of TESTR.

Acknowledgement We thank Intel Corporation for an award. We thank Weijian Xu, Yifan Xu, and Tianyi Xiong for valuable feedbacks.

References

[1] Youngmin Baek, Bado Lee, Dongyoon Han, Sangdoo Yun, and Hwalsuk Lee. Character region awareness for text detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9365–9374, 2019. 2

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229, 2020. 1, 4

[3] Chee Kheng Ch'ng, Chee Seng Chan, and Chenglin Liu. Total-text: Towards orientation robustness in scene text detection. *International Journal on Document Analysis and Recognition (IJDAR)*, 23:31–52, 2020. 6

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2019. 1

[5] Qi Dong, Zhuowen Tu, Haofu Liao, Yuting Zhang, Vijay Mahadevan, and Stefano Soatto. Visual relationship detection using part-and-sum transformers with composite queries. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3550–3559, October 2021. 4

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 1

[7] Boris Epshtein, Eyal Ofek, and Yonatan Wexler. Detecting text in natural scenes with stroke width transform. In *CVPR*, pages 2963–2970, 2010. 2

[8] Wei Feng, Wenhao He, Fei Yin, Xu-Yao Zhang, and Cheng-Lin Liu. Textdragon: An end-to-end framework for arbitrary shaped text spotting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 3, 7

[9] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 3

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 6

[11] Tong He, Zhi Tian, Weilin Huang, Chunhua Shen, Yu Qiao, and Changming Sun. An end-to-end textspotter with explicit alignment and attention. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5020–5029, 2018. 1, 2, 8

[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*, pages 1156–1160, 2015. 6

[13] H. W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955. 5

[14] Chen-Yu Lee and Simon Osindero. Recursive recurrent nets with attention modeling for ocr in the wild. In *CVPR*, pages 2231–2239, 2016. 2

[15] Hui Li, Peng Wang, and Chunhua Shen. Towards end-to-end text spotting with convolutional recurrent neural networks. In *Proceedings of the IEEE international conference on computer vision*, pages 5238–5246, 2017. 1, 2

[16] Ke Li, Shijie Wang, Xiang Zhang, Yifan Xu, Weijian Xu, and Zhuowen Tu. Pose recognition with cascade transformers. In *CVPR*, pages 1944–1953, 2021. 1

[17] Minghui Liao, Guan Pang, Jing Huang, Tal Hassner, and Xiang Bai. Mask textspotter v3: Segmentation proposal network for robust scene text spotting. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 3, 7, 8

[18] Minghui Liao, Baoguang Shi, and Xiang Bai. Textboxes++: A single-shot oriented scene text detector. *IEEE Transactions on Image Processing*, 27(8):3676–3690, Aug 2018. 2

[19] Minghui Liao, Baoguang Shi, Xiang Bai, Xinggang Wang, and Wenyu Liu. Textboxes: A fast text detector with a single deep neural network. In *Thirty-first AAAI conference on artificial intelligence*, 2017. 2, 7

[20] Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 3

[21] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 5

[22] Wei Liu, Chaofeng Chen, Kwan-Yee K Wong, Zhizhong Su, and Junyu Han. Star-net: a spatial attention residue network for scene text recognition. In *BMVC*, volume 2, page 7, 2016. 2

[23] Xuebo Liu, Ding Liang, Shi Yan, Dagui Chen, Yu Qiao, and Junjie Yan. Fots: Fast oriented text spotting with a unified network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 1, 2, 7, 8

[24] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9809–9818, 2020. 1, 2, 3, 4, 6, 7

[25] Yuliang Liu, Lianwen Jin, Shuaitao Zhang, Canjie Luo, and Sheng Zhang. Curved scene text detection via transverse and longitudinal sequence connection. *Pattern Recognition*, 90:337–345, June 2019. 6

[26] Yuliang Liu, Chunhua Shen, Lianwen Jin, Tong He, Peng Chen, Chongyu Liu, and Hao Chen. Abcnet v2: Adaptive bezier-curve network for real-time end-to-end text spotting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2021. 2, 3, 7, 8

[27] George G Lorentz. *Bernstein polynomials*. American Mathematical Soc., 2013. 4

[28] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 6

[29] Pengyuan Lyu, Minghui Liao, Cong Yao, Wenhao Wu, and Xiang Bai. Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018. 1, 3, 7

[30] Anand Mishra, Karteek Alahari, and CV Jawahar. Top-down and bottom-up cues for scene text recognition. In *CVPR*, pages 2687–2694, 2012. 2

[31] Nibal Nayef, Yash Patel, Michal Busta, Pinaki Nath Chowdhury, Dimosthenis Karatzas, Wafa Khlif, Jiri Matas, Umapada Pal, Jean-Christophe Burie, Cheng-lin Liu, et al. Icdar2019 robust reading challenge on multi-lingual scene text detection and recognition—rrc-mlt-2019. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1582–1587. IEEE, 2019. 6

[32] Lukáš Neumann and Jiří Matas. Real-time scene text localization and recognition. In *CVPR*, pages 3538–3545, 2012. 2

[33] Liang Qiao, Ying Chen, Zhanzhan Cheng, Yunlu Xu, Yi Niu, Shiliang Pu, and Fei Wu. Mango: A mask attention guided one-stage scene text spotter. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(3):2467–2476, May 2021. 1, 3, 7, 8

[34] Liang Qiao, Sanli Tang, Zhanzhan Cheng, Yunlu Xu, Yi Niu, Shiliang Pu, and Fei Wu. Text perceptron: Towards end-to-end arbitrary-shaped text spotting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11899–11907, 2020. 1, 3, 7, 8

[35] Siyang Qin, Alessandro Bissacco, Michalis Raptis, Yasuhisa Fujii, and Ying Xiao. Towards unconstrained end-to-end text spotting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 3, 7, 8

[36] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, pages 91–99, 2015. 1

[37] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 658–666, 2019. 5

[38] Baoguang Shi, Xiang Bai, and Cong Yao. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE transactions on pattern analysis and machine intelligence*, 39(11):2298–2304, 2016. 1, 2

[39] Baoguang Shi, Xinggang Wang, Pengyuan Lyu, Cong Yao, and Xiang Bai. Robust scene text recognition with automatic rectification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4168–4176, 2016. 2

[40] Bolan Su and Shijian Lu. Accurate scene text recognition based on recurrent neural network. In *Asian Conference on Computer Vision*, pages 35–48. Springer, 2014. 2

[41] Yipeng Sun, Chengquan Zhang, Zuming Huang, Jiaming Liu, Junyu Han, and Errui Ding. Textnet: Irregular text reading from images with an end-to-end trainable network. In *Asian Conference on Computer Vision*, pages 83–99. Springer, 2018. 2, 8

[42] Zhi Tian, Weilin Huang, Tong He, Pan He, and Yu Qiao. Detecting text in natural image with connectionist text proposal network. In *European conference on computer vision*, pages 56–72. Springer, 2016. 2

[43] 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*, pages 5998–6008, 2017. 1

[44] Hao Wang, Pu Lu, Hui Zhang, Mingkun Yang, Xiang Bai, Yongchao Xu, Mengchao He, Yongpan Wang, and Wenyu Liu. All you need is boundary: Toward arbitrary-shaped text spotting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(07):12160–12167, Apr. 2020. 7, 8

[45] Kai Wang, Boris Babenko, and Serge Belongie. End-to-end scene text recognition. In *ICCV*, pages 1457–1464, 2011. 2

[46] Pengfei Wang, Chengquan Zhang, Fei Qi, Shanshan Liu, Xiaoqiang Zhang, Pengyuan Lyu, Junyu Han, Jingtuo Liu, Errui Ding, and Guangming Shi. Pgnet: Real-time arbitrarily-shaped text spotting with point gathering network. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(4):2782–2790, May 2021. 2, 3, 7, 8

[47] Xiaobing Wang, Yingying Jiang, Zhenbo Luo, Cheng-Lin Liu, Hyunsoo Choi, and Sungjin Kim. Arbitrary shape scene text detection with adaptive text region representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6449–6458, 2019. 2

[48] Linjie Xing, Zhi Tian, Weilin Huang, and Matthew R. Scott. Convolutional character networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 3, 7, 8

[49] Yifan Xu, Weijian Xu, David Cheung, and Zhuowen Tu. Line segment detection using transformers without edges. In *CVPR*, pages 4257–4266, 2021. 1

[50] Cong Yao, Xiang Bai, Wenyu Liu, Yi Ma, and Zhuowen Tu. Detecting texts of arbitrary orientations in natural images. In *CVPR*, pages 1083–1090, 2012. 1

[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 *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 5551–5560, 2017. 2

[52] 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, 3, 4, 6