DEER: Detection-agnostic End-to-End Recognizer for Scene Text Spotting

Seonghyeon Kim* Seung Shin Yoonsik Kim Han-Cheol Cho Taeho Kil
Jaeheung Surh Seunghyun Park Bado Lee Youngmin Baek
NAVER Clova

Abstract

Recent end-to-end scene text spotters have achieved great improvement in recognizing arbitrary-shaped text instances. Common approaches for text spotting use region of interest pooling or segmentation masks to restrict features to single text instances. However, this makes it hard for the recognizer to decode correct sequences when the detection is not accurate i.e. *one or more characters are cropped out. Considering that it is hard to accurately decide word boundaries with only the detector, we propose a novel Detection-agnostic End-to-End Recognizer, DEER, framework. The proposed method reduces the tight dependency between detection and recognition modules by bridging them with a single reference point for each text instance, instead of using detected regions. The proposed method allows the decoder to recognize the texts that are indicated by the reference point, with features from the whole image. Since only a single point is required to recognize the text, the proposed method enables text spotting without an arbitrarily-shaped detector or bounding polygon annotations. Experimental results present that the proposed method achieves competitive results on regular and arbitrarily-shaped text spotting benchmarks. Further analysis shows that DEER is robust to the detection errors. The code and dataset will be publicly available.*

1. Introduction

End-to-end scene text spotting has recently gained a lot of attention due to its simplicity and performance gain. These works also have many practical applications in various areas, such as information extraction, image retrieval, and visual question answering. Commonly, an end-to-end text spotting pipeline consists of a text detector and a recognizer. The detector outputs a box or polygon representation to localize the text instances within an image, and the recognizer takes each localized text region as an input to decode the characters within each patch of the image.

(a) Outputs of DEER on challenging examples.

(b) Outputs of a segmentation based model with some incorrect predictions.

(c) Outputs of DEER when the reference point is obtained from (b).

Figure 1. (a) is the successful outputs of DEER on challenging samples from other papers [2, 13]. + indicates the sampled reference point. (b) is the outputs of a comparison model with some incorrect predictions. The red outline indicates detection output. (c) is the output of DEER when the reference point is obtained from (b)'s detection output that involves incorrect detection.

*Correspondence to kim.seonghyeon@navercorp.com

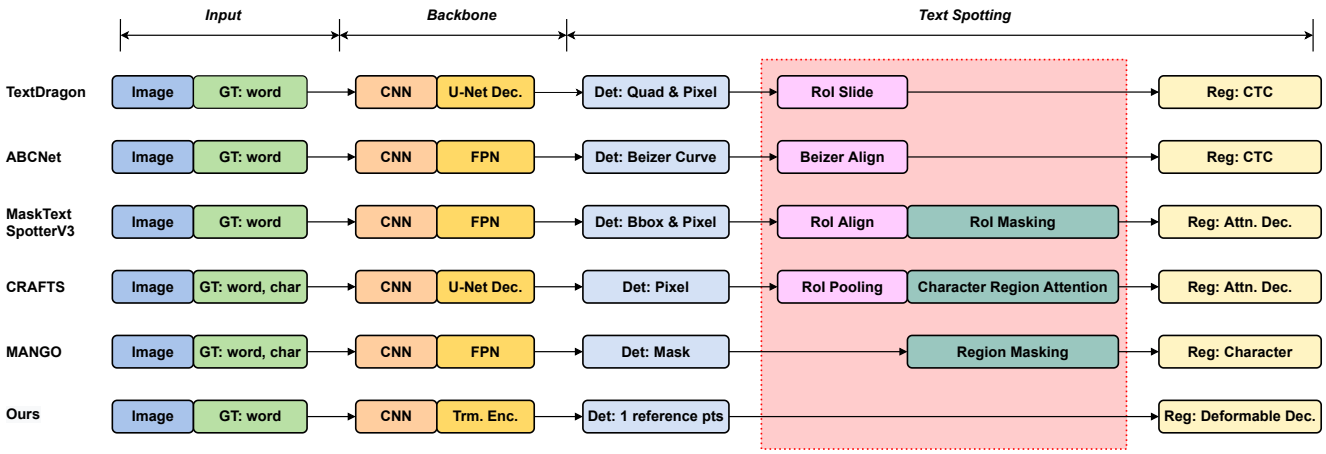


Figure 2. An overview of various end-to-end text spotting methods. The proposed method is different from other alternatives in that it does not use any pooling or masking technique.

Previous scene text spotting pipelines used a rather tightly coupled framework between detector [1, 14, 16, 23] and recognizer [5, 6, 10, 24–26, 36]. Specifically, cropped $images$ from the detector are fed to the recognizer, which inevitably means that the recognition performance is critically dependent on that of the detector and image patches it outputs. Recently, end-to-end text spotting methods [2, 13, 17, 18] proposed the use of a more loosely coupled framework by utilizing ROI pooling or masking to extract $features$ and restrict the input region for the recognizer to contain a single word. Although taking the localized features for the recognizer could lessen the dependency of the recognizer on the cropped regions from detectors, the errors from the detector are still accumulated, and thus, these errors can cause recognition failure cases as shown in Figure 1b. Moreover, feature pooling and masking still require data with bounding boxes (or bounding polygons) to train end-to-end text spotting models, even if the final application does not require exact bounding box information.

With the advancement of recent end-to-end Transformer-based approaches in the field of object detection [3, 4, 35, 38], it is becoming more evident that exact region information, sophisticated ground truth assignment, and feature pooling are not strictly required to recognize individual objects in the image. From these observations, we propose a novel Detection-agnostic End-to-End Recognizer, denoted as DEER, that drastically relieves the dependency on the exactness of detection results. Instead of relying on detectors to extract accurate text regions, we let the detector localize a single reference $point$ for each text instance. Then, we design a text decoder to comprehensively recognize texts around the corresponding reference point. Specifically, given a reference point, the text decoder learns to determine the attending region of that specific text instance while decoding the text sequence. Since DEER only requires the detector to localize a single reference point, it allows the model to use a much wider variety of detection algorithms and annotations. Moreover, this approach can naturally handle rotated and curved text instances without pooling operations and polygon-type annotations.

As shown in Figure 1a, DEER successfully recognizes challenging samples mentioned in [2, 13]. The samples present rotated text, text-in-text, and text with granularity issues. Thus, DEER achieves comparable performance to that of the state-of-the-art models. Moreover, we validate the detection agnostic property of DEER by investigating diverse ablation studies. Figure 1c shows detection agnostic property of DEER, where the reference point from other detection output is used together with the decoder in DEER. Although the detected region is not accurate and the reference point is also biased, DEER correctly recognizes the words.

2. Related Works

In this section, we review various end-to-end text spotting models and study the latest object detection and segmentation models that inspired the development of the proposed method.

2.1. End-to-end Scene Text Spotting

Early end-to-end text spotting models, like TextBoxes++ [14] for instance, separately trained the text detector [15] and recognizer [24], and joined them afterwards. Following studies [2, 13, 17, 18] found that jointly training both detector and recognizer improved the final performance. Because the performance of these end-to-end models heavily relies on detection results, many

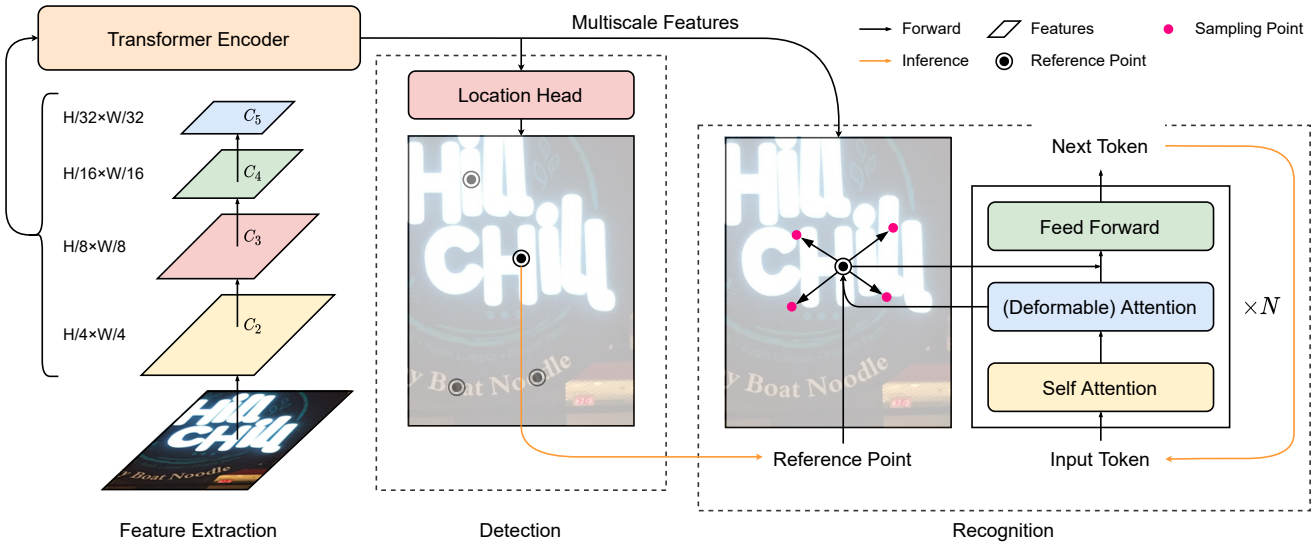


Figure 3. Overview of the proposed method. Refined feature tokens from the encoder is used as an input to the location head and the text decoder. During inference, center points from the location head is used as reference points.

works focused on developing sophisticated detection and feature pooling/masking algorithms. For example, ABCNet [18] localized arbitrary-shaped text instances by using a Bezier-curve fitting algorithm. TextDragon [8] used an RoI sliding algorithm to group a series of local features along the centerline. Another line of work utilized segmentation as its technique. For instance, MaskTextSpotterV3 [13] adopted hard RoI masking to remove features not related to the target text instances. MANGO [20] used an image-level mask attention, which doesn't need RoI operation.

As shown in Figure 2, while existing methods include additional modules to restrict the input features, the proposed method performs text spotting without these modules. The proposed method is different from other end-to-end approaches in that the detector provides only a reference point to the recognizer. The recognizer exploits any information from the reference point and the entire input to decode the output text sequence.

2.2. End-to-end Object Detection and Segmentation

Recent Transformer-based object detection and instance segmentation models have achieved impressive results. DETR [3] showed competitive results without using sophisticated hand-crafted components, such as spatial anchor-boxes or non-maximum suppression. Deformable DETR [38] improved the training speed of DETR by employing deformable attention and solved the issue of performance on small objects by using multi-scale features. Efficient DETR [35] further accelerated the training speed by using well-initialized reference points and object queries generated from the initial dense object detection stage. For panoptic segmentation, Panoptic SegFormer [12] achieved state-of-the-art performance by using two-stage decoders, namely a location decoder and a mask decoder.

These studies showed that an explicit detection proposal is not required to achieve high recognition performance. Based on this observation, we adopt the Transformer architecture and the concept of reference points (or location queries) for the end-to-end text spotting task, relaxing the dependency between detector and recognizer.

3. DEER

The overall pipeline of DEER is shown in Figure 3. DEER consists of a backbone, Transformer encoder, location head, and text decoder. First, the Transformer encoder combines the multi-scale feature maps generated by the backbone. Then, the location head predicts the reference points of text instances and (optionally) bounding boxes. Finally, the text decoder generates the character sequences in each text instance specified by their reference points.

In the forward phase, an input image $X \in \mathbb{R}^{H \times W \times 3}$ is fed into the backbone, and the feature maps C_2, C_3, C_4, C_5 are extracted. The resolution of each extracted feature corresponds to $1/4$, $1/8$, $1/16$, $1/32$, respectively. We then apply a fully-connected layer and group normalization [33] on these feature maps to project into 256 channels. Then, we flatten and concatenate them into feature tokens with size $(L_2 + L_3 + L_4 + L_5) \times 256$, where L_i corresponds to the flattened length of C_i, which is $\frac{H}{2^i} \times \frac{W}{2^i}$. Then, using this as its input, the Transformer encoder outputs the refined

features. We use features of size L_2, which corresponds to C_2 as the input to the location head. Finally, by using the reference points and the refined feature outputs, the text decoder generates the character sequences within the text instance autoregressively.

3.1. Transformer Encoder

Using high-resolution, multi-scale features is beneficial for text recognition. Since the computation cost of self-attention increases quadratically with input lengths, it is impractical to use Transformer on the concatenation of multi-scale features. Therefore, previous approaches that employ Transformer encoders for object detection have used lower-resolution features like C_5.

Unlike these previous methods, we use deformable attention, which scales linearly with input lengths. Due to the efficiency of deformable attention, our encoder is able to refine high-resolution concatenated features to generate multi-scale feature tokens F.

Deformable Attention Deformable attention [38] is a crucial component for both the encoder and decoder, due to its efficiency and location-awareness. Deformable attention is calculate by

$$\begin{aligned}&\text{DeformAttn}_h(A_{hqk}, p_{\text{ref}}, \Delta p_{hqk}) \\ &= W_h^o \left[\sum_{k=1}^{K} A_{hqk} \cdot W_h^k x(v, p_{\text{ref}} + \Delta p_{hqk}) \right], \quad (1)\end{aligned}$$

where $x(v, p)$ is a bilinear interpolation that extracts features from value features v at position p. K is the number of sampled key points and k is the key index. h is the index for the attention head, and $W_h^o \in \mathbb{R}^{C \times C_m}$, $W_h^k \in \mathbb{R}^{C_m \times C}$ are linear projections. p_{ref}, Δp_{hqk}, and A_{hqk} correspond to reference points, sampling offsets, and attention weights, respectively. p_{ref}, Δp_{hqk}, and A_{hqk} are computed by applying linear projections to the query features, and a softmax is applied on A_{hqk}. In the encoder, we use fixed reference points with normalized coordinates on $[0, 1] \times [0, 1]$.

Instead of using predefined reference points, as shown in [35], image-specific reference points can be used to accelerate model training. Also, the reference points positioned in each object can be used to decode specific object instances. We have used this feature to decode specific text instances.

3.2. Location Head

Utilizing location information is beneficial for recognizing and distinguishing objects. Inspired by previous approaches [12, 35], we use a location head to predict reference points (*i.e.*, center position of text instance) for the text decoder. Additionally, it provides a segmentation map to extract the bounding polygon of text instances, which is required only for computing evaluation metrics for our purposes. We adopted differentiable binarization (DB) [16] to extract bounding polygon of text instances, which is required for the evaluation.

Specifically, we extract feature tokens of size L_2 from F, which corresponds to C_2, then reshaped them to $(H/4, W/4)$. Then, to obtain binary and threshold maps we apply a separated segmentation head which consists of transposed convolution, group normalization, and relu. In the inference phase, the center coordinates of text instances are obtained from the detected bounding polygons and are used as the reference points.

3.3. Text Decoder

The recognition branch, which consists of a Transformer decoder, predicts the character sequences within the text instance autoregressively. The query Q for the text decoder is composed of the character embedding, positional embedding, and reference point q_{ref}. The keys K and values V of the text decoder are the feature tokens F from the Transformer encoder. We pass the queries through self-attention, deformable attention, and feed-forward layers. Inspired by [12, 37], we also add a regular cross attention instead of deformable attention to the F alternatingly.

During training, N_t text boxes are sampled from the image and the computed center coordinates are used as reference points for the decoder. This allows for the independent training of the location head and text decoder. In the training stage, we use the points computed by the ground truth regions as reference points for the text decoder. For evaluation, the center coordinate from the detection branch is used as the reference point. To reduce the gaps between coordinates from the ground truth (training) and model prediction (evaluation), the center coordinates are perturbed in the training stage using the equation below:

$$\begin{aligned}q_{\text{ref}} &= p_c + \frac{\eta}{2} \min(\|p_{\text{tl}} - p_{\text{tr}}\|, \|p_{\text{tl}} - p_{\text{bl}}\|), \\ \eta &\sim \text{Uniform}(-1, 1) \quad (2)\end{aligned}$$

where p_c is the centroid of the ground truth polygon, and p_{tl}, p_{tr}, p_{bl} correspond to the coordinates of the top-left point and adjacent top-right and bottom-left points. In the inference phase, the center point of the text regions extracted from the detection stage is used as the reference point.

3.4. Optimization

The loss function L used for training is defined as below:

$$L = L_r + \lambda_s L_s + \lambda_b L_b + \lambda_t L_t, \quad (3)$$

where L_r is the autoregressive text recognition loss, and L_s, L_b, L_t are the losses from differentiable binarization [16]. Each denotes a loss for the probability map, binary map,

and threshold map. L_r is computed with a softmax cross entropy between the predicted probability of character sequences and the ground truth text labels of corresponding text box. Following differentiable binarization, we apply a binary cross entropy with hard negative minining for L_s, dice loss for L_b, and L_1 distance loss for L_t.

During inference, we only use the probability map from the location head. The probability map is binarized with a specified threshold, and connected components are extracted from the binary map. As stated in a previous work [16], the size of the extracted regions is smaller than the actual text regions. Therefore, the extracted regions are dilated using the Vatti clipping algorithm with offset D, $D = \frac{A \times r}{L}$, where A is the area of a polygon region, L is the perimeter of a polygon, and r is the pre-defined dilation factor. After extracting the polygon from each dilated region, we calculate the center coordinates and give it as the reference point to the decoder. Finally, the decoder greedily predicts character sequences in the corresponding text region.

4. Experiments

4.1. Datasets

TextOCR [27] is a scene text dataset which contains arbitrary shaped text instances. It is annotated with polygons and contains 24,902 training images and 3,232 testing images.

ICDAR 2015 [11] is a scene text dataset with quadrilateral text instances. It includes 1,000 training images and 500 testing images.

TotalText [7] is a scene text dataset with various text instances. It includes rotated and curved text instances, and is annotated using polygon bounding boxes. The dataset is composed of 1,255 training and 300 testing images.

4.2. Implementation details

The model is pretrained on TextOCR dataset for 400k steps with initial 10k warmup steps. We use Adam optimizer with a cosine learning rate scheduler. For pretraining, the learning rate is set to $3e-4$, and the weight decay is set to $1e-6$. The pretrained model is then fine-tuned on each benchmark dataset for 10k steps with a learning rate of $5e-5$. We use a batch size of 32 and sample 2 text instances on each image to train the recognizer.

For training data augmentation, we apply random rotation between -90 and 90 degrees, random resize between 50% and 300%, safe random crop up to 640 pixels while preserving the bounding box, and color jitter with a probability of 0.8. For inference, the longer side of the input image is resized to 1280 or 1920 pixels while keeping the aspect ratio.

4.3. Experimental Results

On the ICDAR 2015 dataset, the end-to-end performance is evaluated under *Strong*, *Weak*, and *Generic* contextualization settings. Each uses a separate lexicon to match the closest word produced by the model. Also, texts of length less than 3 are ignored, and special characters in the first or last position of a word are removed. The performance of DEER on ICDAR 2015 dataset is listed in Table 1. With a score of 75.6, the model achieves state-of-the-art results under the generic contextualization setting. Note that the proposed method shows a lower detection score compared to previous works [30, 34], but it achieves higher end-to-end performance. This result indicates that the recognition performance is more robust to detection error compared to previous works.

For the TotalText dataset, we measure the end-to-end performance in two ways. *None* indicates that the model does not use any lexicon for evaluation, and *Full* denotes that the model fully exploits a lexicon with the ground truth labels. Table 2 shows the quantitative results. DEER achieves competitive results without feature pooling or masking.

For qualitative results, we show eight examples in Figure 4. The yellow cross represents the reference point. Using only a single point as a cue, DEER is able to correctly recognize texts in complex and dense scenes. Specifically, as shown in the first and second examples, we see that DEER successfully handles complex layouts such as text-in-text, crossing texts, rotated texts, and diverse font sizes. DEER can also handle text instances in perspective and arbitrarily shaped texts. We also acknowledge that our text decoder does not highly depend on detected text regions. As can be seen in Figure 1c, our model correctly predicts text sequence by attending to the extracted full image features even if detection results are inaccurate. We plan to present more visual results as supplementary material.

We visualize the attention map in Figure 5 to analyze the roles of deformable attention and plain attentions. Figure 5a shows that deformable attention attends to the feature around the reference points in the first layer. As the layer progresses, it attends to the features around the entire text instances. On the other hand, plain attention in Figure 5b attends to the entire text instances in the scene at the first layer. For this reason, the attention maps of each text instance are overlapped. In higher layers, plain attention attends to each character in the text specified by the reference point.

4.4. Ablation study

To analyze whether our proposed method is robust to the detection errors, we conduct ablation experiments on ICDAR 2015 and TotalText datasets. No lexicon is used throughout these experiments.

Method	Detection			E2E			
	Recall	Precision	F-measure	Strong	Weak	Generic	None
TextNet* [28]	85.41	89.42	87.37	78.66	74.90	60.45	-
E2E TextSpotter [9]	86.00	87.00	87.00	82.00	77.00	63.00	-
MaskTextSpotter* [19]	81.00	91.60	86.00	79.30	73.00	62.40	-
FOTS* [17]	87.92	91.85	89.84	83.55	79.11	65.33	-
TextDragon [8]	83.75	92.45	87.88	82.54	78.34	65.15	-
CharNet* [34]	**90.47**	92.65	91.55	85.05	81.25	71.08	67.24
Qin *et al*. [22]	87.96	91.67	89.78	**85.51**	81.91	69.94	-
CRAFTS [2]	85.30	89.00	87.10	83.10	**82.10**	74.90	**74.90**
MaskTextSpotter v3 [13]	-	-	-	83.30	78.10	74.20	-
Boundary [29]	87.50	89.80	88.60	79.70	75.20	64.10	-
TextPerceptron [21]	82.50	92.30	87.10	80.50	76.60	65.10	-
ABCNet v2* [18]	86.00	90.40	88.10	83.00	80.70	75.00	-
MANGO [20]	-	-	-	85.40	80.10	73.90	-
PGNetPGNet [31]	84.80	91.80	88.20	83.30	78.30	63.50	-
Li *et al*. [30]	-	-	**91.60**	84.23	78.04	65.53	-
Ours	86.23	**93.72**	89.82	82.71	79.07	**75.64**	71.72

Table 1. Experiment results on IC15. Models with * use multi-scale inference, and strong/weak/generic columns mean recognition with strong, weak, and generic lexicons, respectively.

Method	Detection			E2E	
	Recall	Precision	F-measure	None	Full
TextNet [28]	59.45	68.21	63.53	54.02	-
Mask TextSpotter [19]	55.00	69.00	61.30	52.90	71.80
TextDragon [8]	75.70	85.60	80.30	48.80	74.80
CharNet* [34]	85.00	88.00	86.50	69.20	-
PAN++ [32]	81.00	89.90	85.30	68.60	78.60
Qin *et al*. [22]	85.00	87.80	86.40	70.70	-
CRAFTS [2]	**85.40**	89.50	**87.40**	**78.70**	-
Mask TextSpotter v3 [13]	-	-	-	71.20	78.40
ABCNet v2* [18]	84.10	90.20	87.00	73.50	80.70
MANGO [20]	-	-	-	72.90	**83.60**
Li *et al*. * [30]	59.38	63.25	61.25	58.56	-
Ours	81.44	**90.44**	85.71	74.84	81.34

Table 2. Experiment results on TotalText. Models with * use multi-scale inference. E2E result with Full and None types mean that recognition is done with/without a lexicon, respectively.

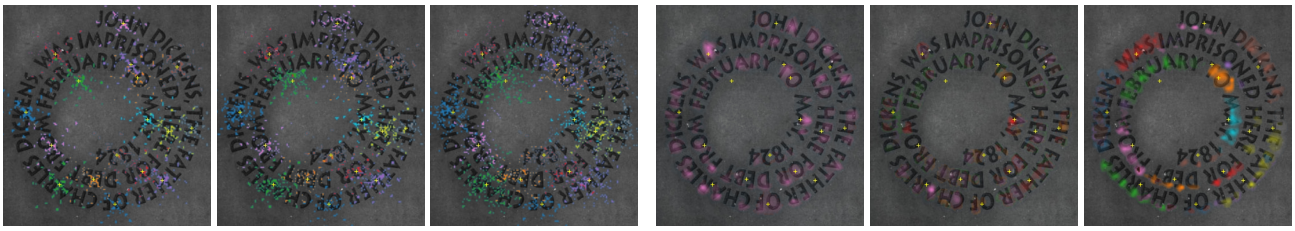
Effectiveness of reference point perturbation Table 3 shows DEER trained with or without the perturbation of the center coordinates p_{ref} using Equation 2. The model trained with perturbation shows slightly better performance on both datasets.

Robustness against detection error When detection results are not accurate, it is not guaranteed that the reference point p_{ref} will be placed at the center point of the polygon. To simulate this error, we deliberately move the ground truth reference point from the center point towards the top-left point by β. Therefore, the new center point is $p'_{\text{ref}} = (1-\beta)p_{\text{ref}} + \beta p_{\text{tl}}$, where p_{ref} is the original center point and p_{tl} is the top-left point of the bounding polygon.

Figure 6 shows the end-to-end F1-score on TotalText when β changes from 0.0 to 0.4. While the noisy reference points deteriorate the overall performance, the proposed model trained with reference point perturbation is much more robust to detection errors than the one trained without it.

Alternative selection of reference point For a text instance

Figure 4. Qualitative results of DEER on IC15 [11] and TotalText [7] datasets. The reference point is presented as +. Please zoom in for better visualization.



(a) Visualized deformable attention map.

(b) Visualized plain attention map.

Figure 5. Visualization of (a) deformable attention from layers 1, 3, and 5, (b) plain attentions from layers 2, 4, and 6, respectively. Each text instance is color-coded.

Dataset	Perturbation	Detection	E2E
ICDAR 2015	✓	89.82	71.72
		89.11	70.63
TotalText	✓	85.71	75.32
		84.19	74.91

Table 3. **Pertubation** – Applying perturbation on reference point p_{ref} during training is beneficial for the model performance, and it suggests that the model could be trained with noisy supervisions.

Dataset	Sampling	Detection	E2E
ICDAR 2015	Center	89.82	71.72
	Inner	88.58	71.01
TotalText	Center	85.71	75.32
	Inner	84.19	75.47

Table 4. **Center point *vs*. Inner point** – The proposed method could be trained to use a more divers set of points inside the text polygon. This suggest that the model could recognize text sequences correctly even when the reference point is located outside of the text instance.

with extreme shape, its reference point obtained by simply taking the average of the polygon vertices might be located outside the bounding polygon. In this experiment, we apply an alternative reference point sampling method, namely the *Inner* method. For training, we simply use a random reference point inside the annotation polygon. For inference, we take the midpoint of the center cross section of the polygon.

Robustness against perturbing the reference points and alternative strategy for the sampling shows that DEER could be trained with noisy annotations.

Table 4 compares the performance of the proposed model using the Inner method to the original one. The

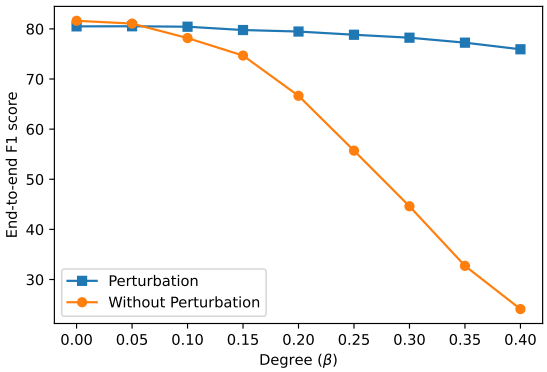


Figure 6. Moving the reference point during inference from the center point of the bounding boxes to their top-left points by degree. A model trained with perturbation is robust to noise caused by the moving of reference points.

	Detection	Recall	Precision	F1
ICDAR 2015		74.39	75.22	74.80
	✓	72.85	73.55	73.20
TotalText		80.19	80.81	80.50
	✓	78.87	76.91	79.24

Table 5. **Detection Supervision** – Training the model to detect text instances is beneficial for the recognition of the text sequences.

original sampling method performs comparably to the Inner method. This result indicates that reference points do not have to necessarily be located inside the text area for correct recognition; therefore, the proposed method is robust to recognizing text instances with extreme shapes.

Training with detection supervision The proposed model does not directly use the region information of the text instances except for the reference points. However, the detection could guide the features to be beneficial for the recognizer. To verify this idea, we train DEER without detection loss. The model under this setting does not produce any detection results. Therefore, during evaluation, we use ground truth annotations to obtain the reference points.

Table 5 shows that the detection supervision benefits the recognizer. However, we expect that using extended training and a larger dataset could reduce the gap, as detection supervision is only used for guidance during training. We also believe that high-performing end-to-end text spotting models can be trained through just point supervision, eliminating the need for expensive polygon annotations.

Figure 7. Structured error patterns in IC15. In the left image, DEER predicts 'SALE' on ambiguous texts with red backgrounds as the word 'SALE' frequently occurs on red backgrounds. On the right, as word 'off' frequently occurs with the word 'SALE', 'OFF' is predicted instead of 'ON,' which is near the word 'SALE'.

5. Discussions

Allowing the recognizer to utilize the entire feature map mitigates the need for accurate detection of text instances, but it could be inefficient when the detection result contains the text instances neatly and the text is unambiguous.

Although it is not required to detect regional information of the text instances, it is required to predict it for the evaluation used in end-to-end text spotting metrics. We suspect that evaluation metrics with relaxed requirements on detecting precise regions could allow interesting research directions.

As the recognizer crucially depends on the detection of reference points and the recognition of each text instance is independent, it is not possible to detect missing text instances or duplicated reference points referring to the same instances.

Using surrounding features that are not restricted within text boundaries and utilizing contextual information could be beneficial to recognize ambiguous characters. However, it could enable the model to utilize spurious cues for recognition. It could make the model have structured error patterns in the recognition results, as shown in Figure 7. Thus it is possible for the model to amplify the bias in the trained datasets.

6. Conclusions

We propose DEER, detection-agonistic end-to-end recognizer for scene text spotting. By relaxing the coupling between the detection of the text bounding polygon and the recognition of text instances, the proposed model allows for the recognition of arbitrary-shaped text instances without the need for rather expensive polygon annotations and elaborate pooling mechanisms. Our experiments show that it is possible to have competitive performances on scene text spotting without using polygon proposals and pooling operations. We hope that the proposed detection-agnostic architecture could help construct scene text spotting models

that are more flexible and easily trained, and offer insights to other vision tasks that require the generation of structured outputs.

References

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

[2] Youngmin Baek, Seung Shin, Jeonghun Baek, Sungrae Park, Junyeop Lee, Daehyun Nam, and Hwalsuk Lee. Character region attention for text spotting. In *Proceedings of European Conference on Computer Vision*, pages 504–521. Springer, 2020. 1, 2, 6

[3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Proceedings of European Conference on Computer Vision*, pages 213–229. Springer, 2020. 2, 3

[4] Ting Chen, Saurabh Saxena, Lala Li, David J Fleet, and Geoffrey Hinton. Pix2seq: A language modeling framework for object detection. *arXiv preprint arXiv:2109.10852*, 2021. 2

[5] Zhanzhan Cheng, Fan Bai, Yunlu Xu, Gang Zheng, Shiliang Pu, and Shuigeng Zhou. Focusing attention: Towards accurate text recognition in natural images. In *Proceedings of International Conference on Computer Vision*, pages 5086–5094, 2017. 2

[6] Zhanzhan Cheng, Yangliu Xu, Fan Bai, Yi Niu, Shiliang Pu, and Shuigeng Zhou. Aon: Towards arbitrarily-oriented text recognition. In *Proceedings of Computer Vision and Pattern Recognition*, pages 5571–5579, 2018. 2

[7] Chee Kheng Ch'ng and Chee Seng Chan. Total-text: A comprehensive dataset for scene text detection and recognition. In *Proceedings of International Conference on Document Analysis and Recognition*, volume 1, pages 935–942. IEEE, 2017. 5, 7

[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 International Conference on Computer Vision*, pages 9076–9085, 2019. 3, 6

[9] Tong He, Zhi Tian, Weilin Huang, Chunhua Shen, Yu Qiao, and Changming Sun. An end-to-end textspotter with explicit alignment and attention. In *Proceedings of Computer Vision and Pattern Recognition*, pages 5020–5029, 2018. 6

[10] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Reading text in the wild with convolutional neural networks. *International Journal of Computer Vision*, 116(1):1–20, 2016. 2

[11] 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 *Proceedings of International Conference on Document Analysis and Recognition*, pages 1156–1160, 2015. 5, 7

[12] Zhiqi Li, Wenhai Wang, Enze Xie, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, Tong Lu, and Ping Luo. Panoptic segformer. *arXiv preprint arXiv:2109.03814*, 2021. 3, 4

[13] 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 European Conference on Computer Vision*, pages 706–722. Springer, 2020. 1, 2, 3, 6

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

[15] Minghui Liao, Baoguang Shi, Xiang Bai, Xinggang Wang, and Wenyu Liu. Textboxes: A fast text detector with a single deep neural network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017. 2

[16] Minghui Liao, Zhaoyi Wan, Cong Yao, Kai Chen, and Xiang Bai. Real-time scene text detection with differentiable binarization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11474–11481, 2020. 2, 4, 5

[17] 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 Computer Vision and Pattern Recognition*, pages 5676–5685, 2018. 2, 6

[18] 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 Computer Vision and Pattern Recognition*, pages 9809–9818, 2020. 2, 3, 6

[19] 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 European Conference on Computer Vision*, pages 67–83, 2018. 6

[20] 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. *arXiv preprint arXiv:2012.04350*, 2020. 3, 6

[21] 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. 6

[22] Siyang Qin, Alessandro Bissacco, Michalis Raptis, Yasuhisa Fujii, and Ying Xiao. Towards unconstrained end-to-end text spotting. In *Proceedings of International Conference on Computer Vision*, pages 4704–4714, 2019. 6

[23] Anhar Risnumawan, Palaiahankote Shivakumara, Chee Seng Chan, and Chew Lim Tan. A robust arbitrary text detection system for natural scene images. *Expert Systems with Applications*, 41(18):8027–8048, 2014. 2

[24] 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. 2

[25] Baoguang Shi, Xinggang Wang, Pengyuan Lyu, Cong Yao, and Xiang Bai. Robust scene text recognition with automatic rectification. In *Proceedings of Computer Vision and Pattern Recognition*, pages 4168–4176, 2016. 2

[26] Baoguang Shi, Mingkun Yang, Xinggang Wang, Pengyuan Lyu, Cong Yao, and Xiang Bai. Aster: An attentional scene text recognizer with flexible rectification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2018. 2

[27] Amanpreet Singh, Guan Pang, Mandy Toh, Jing Huang, Wojciech Galuba, and Tal Hassner. Textocr: Towards large-scale end-to-end reasoning for arbitrary-shaped scene text. In *Proceedings of Computer Vision and Pattern Recognition*, pages 8802–8812, 2021. 5

[28] 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 *Proceedings of Asian Conference on Computer Vision*, pages 83–99. Springer, 2018. 6

[29] 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. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12160–12167, 2020. 6

[30] Peng Wang, Hui Li, and Chunhua Shen. Towards end-to-end text spotting in natural scenes. *Transactions on Pattern Analysis and Machine Intelligence*, 2021. 5, 6

[31] 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. *arXiv preprint arXiv:2104.05458*, 2021. 6

[32] Wenhai Wang, Enze Xie, Xiang Li, Xuebo Liu, Ding Liang, Yang Zhibo, Tong Lu, and Chunhua Shen. Pan++: Towards efficient and accurate end-to-end spotting of arbitrarily-shaped text. *Transactions on Pattern Analysis and Machine Intelligence*, 2021. 6

[33] Yuxin Wu and Kaiming He. Group normalization. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XIII*, volume 11217 of *Lecture Notes in Computer Science*, pages 3–19. Springer, 2018. 3

[34] Linjie Xing, Zhi Tian, Weilin Huang, and Matthew R Scott. Convolutional character networks. In *Proceedings of International Conference on Computer Vision*, pages 9126–9136, 2019. 5, 6

[35] Zhuyu Yao, Jiangbo Ai, Boxun Li, and Chi Zhang. Efficient detr: Improving end-to-end object detector with dense prior. *arXiv preprint arXiv:2104.01318*, 2021. 2, 3, 4

[36] Fangneng Zhan and Shijian Lu. ESIR: end-to-end scene text recognition via iterative image rectification. In *Proceedings of Computer Vision and Pattern Recognition*, pages 2059–2068, 2019. 2

[37] Xizhou Zhu, Dazhi Cheng, Zheng Zhang, Stephen Lin, and Jifeng Dai. An empirical study of spatial attention mechanisms in deep networks. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 6687–6696. IEEE, 2019. 4

[38] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *International Conference on Learning Representations*, 2021. 2, 3, 4