TPD-STR: Text Polygon Detection with Split Transformers

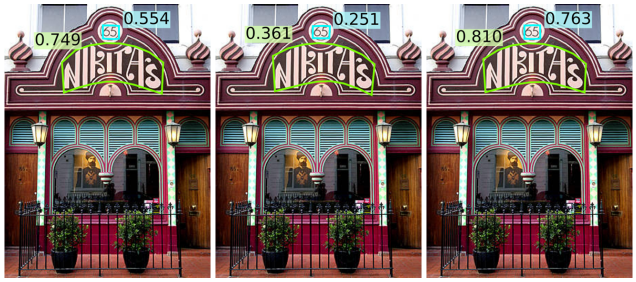
Sangyeon Kim[1,*] Sangkuk Lee[1,2,*] Jeesoo Kim[1] Nojun Kwak[2,†]

[1]NAVER WEBTOON AI [2]Seoul National University

{sangyeon.kim0, sangkuklee, jeesookim}@webtoonscorp.com nojunk@snu.ac.kr

Abstract

Regressing text in natural scenes with polygonal representations is challenging due to shape prediction difficulties. To address this, we introduce Text Polygon Detection with Split Transformers (TPD-STR), which directly regresses polygonal points. TPD-STR incorporates the Decoder Split (DS) architecture to separate polygonal point regression and textness classification, and the Positional Information Propagation (PIP) module to enhance classification. Both modules are effective and compatible with existing methods. TPD-STR achieves state-of-the-art (SOTA) performance among regression-based methods, surpassing segmentation-based methods on MSRA-TD500 without external data. Adding DS and PIP to existing models further improves performance. Experiments demonstrate the model's ability to detect text instances effectively.



(a) w/ Recognition Branch (b) w/o Recognition Branch (c) Ours

Figure 1. Example of lowered textness classification scores when the recognition branch is removed from TESTR [30] (See the numbers). Although TESTR architecturally enabled polygonal detection, without character annotation, making the character recognition branch unavailable, the performance degrades seriously (a) (b). On the other hand, without the recognition branch, our proposed TPD-STR detects polygons very well (c).

1. Introduction

Scene text detection is one of the important tasks in computer vision, acting as a foundation technology for many practical applications such as character recognition, instant translation, scene parsing, blind navigation, alt text[1] and so on. Like other computer vision fields, scene text detection has made great progress by the introduction of deep learning methodologies. In particular, the emergence of transformers [20] has driven a new paradigm of scene text detection by showing outstanding performances. DEtection TRansformer (DETR) [3] is a detector built upon a transformer. Its competitive performance without using any anchor has allowed many derivatives of DETR in the field of scene text detection [17, 19, 27]. However, most DETR-based methods do not directly yield polygonal predictions but focus on finely refining the structure of arbitrary texts [27] or partially use only a few components of DETR [17, 19] to achieve the desired performance indirectly. Unfortunately, rectangular-shaped bounding box detection is not enough to cover the complexity of scene text detection data since they contain various samples with deformed, multi-directional, and curved shapes. Furthermore, it is structurally challenging for neural network models to get good polygonal predictions for the texts in the wild.

Recent studies [25, 30] have proposed a regression-based method using D-DETR [32] to directly regress polygonal points. However, these methods have limitations as scene text detectors. Text Spotting TRansformers (TESTR) [30] is a spotter that simultaneously learns detection and recognition, which is different from the general detector settings as it uses additional recognition labels. DPText-DETR [25] used a structure similar to [30] and proposed a method that better predicts curved text by using additional labels for point positions. Although these two studies report the state-of-the-art performance on some detection benchmark datasets, it is not a fair comparison as they use additional data. In fact, we observed that using only polygonal ground truth for the detection task, without additional data such as recognition labels or point positional labels, seriously degrades performance of these models compared to official

*Equal contribution.
†Corresponding author.
[1]https://en.wiktionary.org/wiki/alt_text

results. This is shown in Tab. 1 (Compare 2nd to 3rd rows of 'Regression-based' (reproduced) and 5th to 6th rows of 'Regression-based' (official)).

In particular, the performance degradation in TESTR [30] originates from the malfunction of the classification head, which determines the presence of texts (textness). While the polygonal point regression head works well enough, as shown in Fig. 1, the textness confidence score significantly drops when the recognition branch is removed. In other words, putting the classification and regression head together is not a desirable strategy for text detection. This phenomenon has also been described in many previous studies, and various solutions have been suggested [1, 2, 5, 6, 11, 19, 24, 27, 33]. CRAFT [1], a CNN-based bottom-up method, utilizes a confidence map by matching the text length included in the text recognition data with the feature map. However, this method has limitations because it requires additional text recognition data. In the recent transformer-based method such as [19] and [27], the confidence score is calculated using a rough text segmentation mask. This approach has also been applied to many previous works [2, 5, 6, 11, 24, 33], but additional training of the segmentation branch requires pre-processing with additional data, making the whole procedure more complex.

In this paper, we propose a simple yet effective transformer-based text detector that can directly regress polygonal points without any need for additional data, data processing, or additional modules other than the basic D-DETR [32] components. To tackle the performance degradation of regression-based methods, first, we separate the two tasks of 'polygonal point regression' and 'textness classification' by splitting the decoder. Through various analyses and studies that will be discussed later, we verify that the performance degradation is caused by the problem of 'shared suboptimal features' that occurs while performing two tasks with different characteristics using shared features. In other words, if the 'polygonal point regression' (reg) branch and the 'textness classification' (cls) branch of the D-DETR [32] structure are trained upon the same attention modules, it is impossible to find the optimal solution for each task. Therefore, we design the 'Decoder Split (DS)' to prevent each branch (reg, cls) from sharing attention weights, ultimately finding better respective features rather than sharing suboptimal features. Secondly, unlike the regression branch, the classification branch can benefit from the information from the regression branch since structural information can be a clue for determining the textness. Therefore, we design the 'Positional Information Propagation (PIP)', which propagates positional information from the regression branch to the classification branch. Through benchmark experiments that will be presented later, it is confirmed that PIP works better on curved text datasets composed of complex polygons. Our contributions can be summarized as follows:

- To tackle the performance degradation of regression-based methods, we propose 'Decoder Split (DS)' which splits the decoder in half and lets each part conduct *polygonal point regression* and *textness classification*, respectively.

- We also introduce the 'Positional Information Propagation (PIP)' module, which transfers positional information from the regression branch to the classification branch. This enables the classification branch to better utilize the geometric information of polygonal points from the regression branch to accurately determine the corresponding textness.

- Our TPD-STR, which directly regresses polygonal points, achieves SOTA among regression-based methods and shows SOTA or competitive performance against segmentation-based methods on standard benchmarks.

- Furthermore, we demonstrated that the DS and PIP modules can be easily incorporated into other methods to further boost performance.

2. Related Work

2.1. Scene Text Detection

Segmentation-based methods [6, 11, 14] create segmentation maps from text boundaries, roughly indicating text regions in images. As scene text shapes vary, these methods effectively handle the problem using segmentation maps. Although the segmentation map is valuable in covering the arbitrary shape of scene texts, model inference requires a binary map with a specific threshold, introducing heuristics. To address arbitrary text shapes, TextSnake [14] predicts score maps of text regions and text center lines with geometry attributes. On the other hand, DB [11] adaptively selects thresholds for binarization to simplify post-processing. FreeReal [6] adopts a student-teacher framework and employs DB [11] as a baseline model.

Regression-based methods [26, 31] directly detect text bounding points, still relying partly on segmentation maps. Though they streamline network architecture, effectively handling scene texts with diverse shapes is still challenging. Using text score maps for thresholding, EAST [31] predicts quadrilateral bounding boxes with diverse orientations. LOMO [26] employs regression to generate quadrangle text proposals, refining and reconstructing text representations including region, center line, and border offsets.

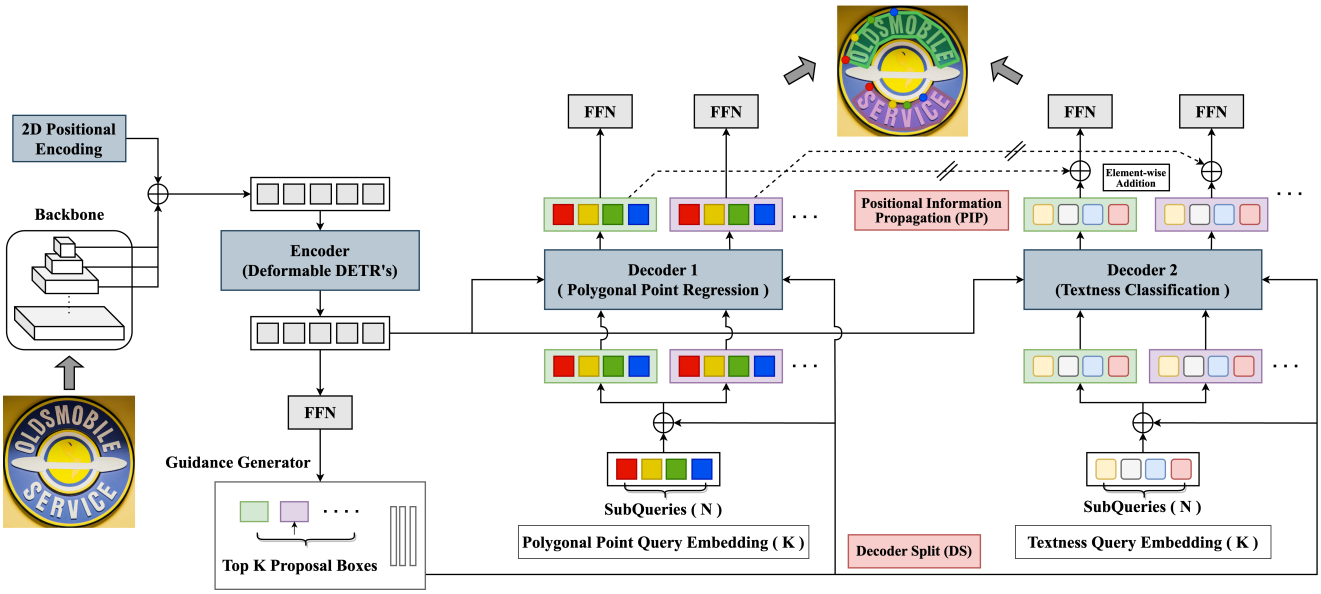


Figure 2. The overall architecture of our method is based on D-DETR [32], and the subquery proposed in [30] is applied for polygonal point regression. Specifically, we propose a decoder split (DS) architecture and positional information propagation (PIP). For visualization, among $N = 16$ polygonal points, four detected points for each polygon are colored in the output image, and the corresponding subqueries are colored accordingly. The double slash sign (//) means preventing gradient flows from the textness classification branch to the polygonal point regression branch.

2.2. Transformers in Scene Text Detection

Recently, there has been growing interest in transformer-based architectures for scene text detection, leading to methods like [2, 19, 27] These approaches use rough text segmentation to regress text instance polygonal points. Specifically, [19] employs score maps for text point features, grouping, and regressing polygonal points. Meanwhile, TextBPN++ [27] refines multiple segmentation maps to regress polygonal points. SRFormer [2] incorporates segmentation branches to its decoder layers. However, these methods need additional segmentation maps for accurate regression of arbitrarily shaped text instances, indicating a need for research in detectors handling such texts without extra maps.

Initially designed for object detection, Deformable DETR (D-DETR) [32] enhanced DETR [3] with a deformable attention mechanism, excelling in small objects. Building on D-DETR's advantages, transformer-based scene text detectors [17, 25, 30] regress polygonal points. However, they have limits. For instance, [17] struggles with curved or complex-shaped text, and DPText-DETR [25] falters without the use of point positional labels. Unlike these, TESTR [30] relies on recognition labels for detection but it degrades without them. In contrast, we propose an effective scene text detector that identifies text instances without needing segmentation maps, point positional labels, or recognition labels.

3. Method

We first introduce the overall architecture of our TPD-STR and then explain our two contributions, Decoder Split (DS) and Positional Information Propagation (PIP) in detail. The overall architecture of our TPD-STR is shown in Fig. 2.

3.1. Overall Architecture

Encoder and Decoder. Our TPD-STR encoder is built upon the D-DETR model [32], and incorporates the decoder's approach of [30] that utilizes subqueries to estimate the polygon coordinates. Our method can detect objects in an arbitrary shape since each subquery corresponds to an individual point that surrounds the text. Given K initial queries $\mathcal{P} = \{P^1, \cdots, P^K\}$, each query $P = (p_1, \cdots, p_N)$ represents a text instance by holding N subqueries to predict N bounding points ($N = 16$ is used in our experiments.). After the encoder E of D-DETR encodes an image x into features, the decoder D turns $\mathcal{P}$ into textness confidence $\hat{\mathcal{B}}$ and predicted coordinates $\hat{\mathcal{P}}$:

$$(\hat{\mathcal{B}}, \hat{\mathcal{P}}) = FFN(D(E(x), \mathcal{P})). \quad (1)$$

Note that $\hat{\mathcal{B}}$ and $\hat{\mathcal{P}}$ share the same queries $\mathcal{P}$.

Meanwhile, in our method, we separate the prediction of textness and regression, as will be discussed later, and

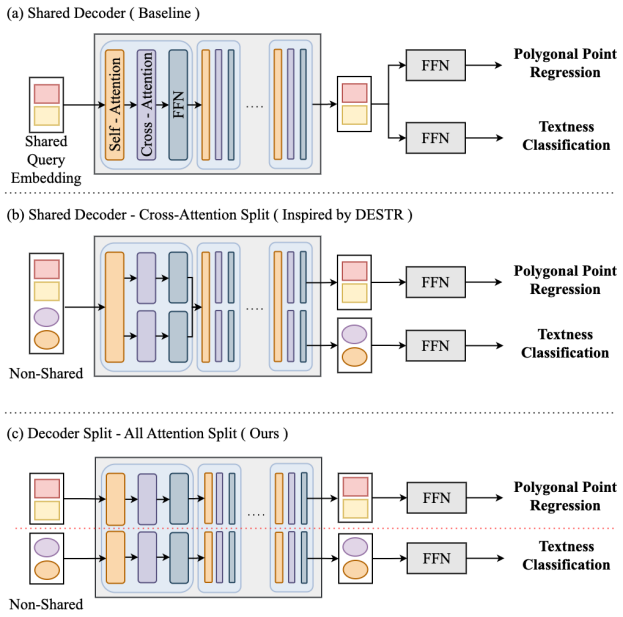


Figure 3. Illustration of various attention split architectures. Tab. 2 shows that our Decoder Split (c) performs best among the three.

Eq. (1) can be rewritten as below:

$$\hat{\mathcal{B}} = FFN_c(D_c(E(x), \mathcal{B})), \tag{2}$$

$$\hat{\mathcal{P}} = FFN_r(D_r(E(x), \mathcal{P})), \tag{3}$$

where $\mathcal{B}$ refers to the query used for textness classification and D_c and D_r respectively represent decoders from classification and regression branches. Accordingly, both branches are followed by their own feed-forward networks FFN_c and FFN_r. Once the points are predicted, they are connected in a clockwise order starting from the left most corner resulting in the final polygonal prediction.

Guidance Generator. In our method, the guidance generator module is utilized to support polygon estimation by detecting text instances as bounding boxes using the encoder output. This concept was first introduced in D-DETR and later used similarly in [8, 30]. In this paper, we follow the implementation of [30] as it is. Encoder E is followed by a linear layer and a normalization layer, producing $\mathbf{C} = \{(c_x, c_y, c_{emb})^1, \cdots (c_x, c_y, c_{emb})^K)\}$. Then, c_{emb} is used to make initial queries as follows:

$$P^k = c^k_{emb} + (p_1, \cdots, p_N), \quad k \in [K], \tag{4}$$

and c_x and c_y are utilized later as reference point coordinates for deformable attention.

3.2. Task Separation with Decoder Split (DS)

In general, locating scene texts has been quite challenging, and many works have proposed additional methods to enhance localization accuracy. In the field of object detection, one of the best working methods is separating the task using an additional branch. For example, DESTR [8] separates the classification and regression branches apart, and this brings a performance gain. This also means that using shared suboptimal features instead of focusing on their own better respective features results in performance degradation. Furthermore, detecting polygonal-shaped objects demands more sophisticated representations than simple bounding box prediction, and it is natural to assume that splitting the task into classification and regression would make better features for each task. Inspired by this, we try to solve the problem of performance degradation in regression-based methods with a transformer by applying the concept of task separation. With this simple yet effective idea, we have tackled the performance limitations of existing methods. Note that, unlike other existing methods, our method does not require complex architectures or additional data such as segmentation masks [2, 5, 6, 19, 27], text recognition labels [30], or point positional labels [25].

Finally, based on the task separation hypothesis, we propose a dual decoder architecture that splits the decoder for polygonal point regression (D_r in Eq. (3)) and text classification (D_c in Eq. (2)) as shown in Fig. 2. In the text polygon detection problem, different features should be attended to the polygonal point regression task and the textness classification task. In this respect, our proposed method is expected to be quite effective. To confirm this quantitatively and qualitatively, we conduct experiments on various decoder split architectures shown in Fig. 3 and the results will be discussed in Sec. 4.4.1.

3.3. Positional Information Propagation (PIP)

As described in the previous section, task separation with decoder split allows the regression and classification branches to find their own respective features. In our experiment, which will be described later, we have verified that this helps improve the detection performance. However, unlike the regression branch, the classification branch can benefit from the information from the regression branch since structural information can be a clue for determining the textness.

Based on this, we formulate the features from the polygonal point regression branch to be transferred to the features from the textness classification branch using element-wise addition, as shown in Fig. 2. After extracting features h_r and h_c from D_r and D_c, we substitute h_c with $h'_c = h_c + h_r$. In this simple way, the classification branch can receive abundant polygonal structural information from the regression branch and make a better prediction of the textness classification score, resulting in better detection performance. The related benchmarks and visualizations are shown in Tab. 2 and Fig. 4, respectively.

	Method	Backbone	Additional Data	Multi-Oriented Dataset						Curved Datasets					
				ICDAR 2015			MSRA-TD500			Total-Text			CTW1500		
				P	R	F	P	R	F	P	R	F	P	R	F
Segmentation-based	TextSnake [14]	VGG16	Seg. Mask	84.9	80.4	82.6	83.2	73.9	78.3	82.7	74.5	78.4	67.9	85.3	75.6
	CRAFT [1]	VGG16	Seg. Mask	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 [21]	Res18	Seg. Mask	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 [11]	Res50	Seg. Mask	**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 [22]	Res50	Seg. Mask	87.6	86.1	86.9	-	-	-	86.9	83.9	85.4	84.1	83.7	83.9
	DRRG [28]	VGG16	Seg. Mask	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 [9]	Res50	Seg. Mask	89.1	**87.3**	88.2	90.4	82.7	86.4	-	-	-	-	-	-
	TextBPN [29]	Res50	Seg. Mask	-	-	-	86.6	84.5	85.6	90.7	85.2	87.9	86.5	83.6	85.0
	Tang *et al.* [19] (RBox, Bezier)	Res50	Seg. Mask	90.9	**87.3**	89.1	-	-	-	90.7	85.7	88.1	88.1	82.4	85.2
	RFN [5]	Res50	Seg.Mask	-	-	-	88.4	**87.8**	88.1	-	-	-	-	-	-
	TextBPN++ [27]	Res50	Seg. Mask	-	-	-	89.2	85.4	87.3	85.3	**91.8**	88.5	83.8	87.3	85.5
	SRFormer [2]	Res50	Seg. Mask, Pos. Label	-	-	-	-	-	-	**92.2**	87.9	**90.0**	**91.6**	**87.7**	**89.6**
	FreeReal [6]	Res50	Seg. Mask, LSVT+	-	-	**90.0**	-	-	**90.1**	-	-	88.9	-	-	87.9
Regression-based	Raisi *et al.* [17]	Res50	-	89.8	78.3	83.7	90.9	83.8	87.2	-		-	-	-	-
	TESTR (Polygon) - w/o Rec. Label (Reproduced*)	Res50	-	90.9	82.6	86.5 (-3.5)	88.9	82.7	85.7	88.7	82.1	85.3 (-1.6)	88.6	84.2	86.3 (-0.8)
	DPText-DETR - w/o Pos. Label (Reproduced*)	Res50	-	-	-	-	-	-	-	**88.9**	83.2	86.0 (-3.0)	**89.5**	86.3	87.9 (-0.9)
	TPD-STR (Ours) - w/o additional data	Res50	-	**91.1**	**87.7**	**89.4**	**93.0**	**88.8**	**90.9**	88.0	**89.3**	**88.7**	89.4	**87.7**	**88.5**
	TESTR [30] (Polygon) - w/ Rec. Label (Official)	Res50	Rec. Label	90.3	89.7	90.0	-	-	-	**93.4**	81.4	86.9	**92.0**	82.6	87.1
	DPText-DETR [25] - w/ Pos. Label (Official)	Res50	Pos. Label	-	-	-	-	-	-	91.8	86.4	89.0	91.7	86.2	88.8
	DPText-DETR + Ours (DS, PIP)	Res50	Pos. Label	-	-	-	-	-	-	89.4	**89.0**	**89.2**	90.6	**87.5**	**89.0**

Table 1. Quantitative results of scene text detection on ICDAR 2015, MSRA-TD500, Total-Text, and CTW1500. The best results are in bold. Reproduced* means the result reproduced through learning without additional data such as segmentation masks, text recognition labels, and point positional labels. Negative numbers in red indicate performance degradation when reproduced without using additional data. LSVT+ denotes the use of additional pre-training datasets, including the 430K LSVT [18] dataset.

Note that our proposed method propagates information only from the regression branch to the classification branch. Therefore, the gradient flow in the opposite direction (from classification branch to regression branch) is blocked during the backpropagation process. If the gradient flows in the opposite direction, the information of each branch would be blended through the backprop, and the effect of task separation (with Decoder Split) may be weakened. We verify this hypothesis through the result in Tab. 2.

3.4. Point Regression and Textness Classification

In this section, we provide a detailed explanation about the overall training process. Given a set of ground truth text polygons $\mathcal{Y} = \{Y^1, \cdots, Y^M\}$, each consisting of N points, *i.e.* $Y^i = (y_1^i, \cdots, y_N^i)$, a query P finds its own match by minimizing the following cost of the Hungarian algorithm:

$$C_i(Y^i, P^{\sigma(i)}) = \lambda_{cls} FL(b^{\sigma(i)}) + \lambda_{coord} \sum_{n=1}^{N} \|y_n^i - p_n^{\sigma(i)}\|_1 \quad (5)$$

$$\text{where } FL(x) = -\alpha(1-x)^\gamma log(x) + (1-\alpha)x^\gamma log(1-x). \quad (6)$$

Here, $\sigma(i)$ is an injective function that represents bipartite matching for the i-th ground truth while b^j represents the confidence of textness for the j-th query. Focal loss FL is adopted from [30] to efficiently handle the classification result. Although $b^{\sigma(i)}$ and $p_n^{\sigma(i)}$ come from different branches, they can still be included in Eq. (5) at the same time since c_x and c_y are constantly fed to the offset of deformable attention modules from both branches.

Using the Hungarian algorithm, we can obtain a set of query indices that match to the M ground truths, $\Omega = \{\sigma(1), \cdots, \sigma(M)\}$. Accordingly, we can compute the regression loss of positive instances as below:

$$\mathcal{L}_{reg}^i = \sum_{n=1}^{N} \|y_n^i - p_n^{\sigma(i)}\|_1. \quad (7)$$

Similarly, we can get the textness classification loss using focal loss, and indices from Ω are utilized as follows:

$$\begin{aligned} \mathcal{L}_{cls}^j = &-\mathbb{1}_{\{j \in \Omega\}} \alpha (1-\hat{b}^j)^\gamma log(\hat{b}^j) \\ &- \mathbb{1}_{\{j \notin \Omega\}} (1-\alpha)(\hat{b}^j)^\gamma log(1-\hat{b}^j), \end{aligned} \quad (8)$$

where $\mathbb{1}$ is the indicator function, and α and γ are the hyperparameters. Finally, the total loss function is defined as follows:

$$\mathcal{L}_{total} = \lambda_{reg} \sum_{i=1}^{M} \mathcal{L}_{reg}^i + \lambda_{cls} \sum_{j=1}^{K} \mathcal{L}_{cls}^j \quad (9)$$

where λ_{reg} and λ_{cls} are the weighting factors for polygonal point regression and textness classification, respectively. Note that M and K are the numbers of ground truth text polygons and the number of queries, respectively.

4. Experiments

4.1. Datasets

SynthText 150k is a synthesized text dataset created in ABCNet [12]. Out of 149,050 images, 94,723 images include straight text instances, while the remaining 54,327 images contain curved text instances. **MLT 2017** [16] is a multi-lingual (9 languages) and multi-oriented text dataset. It consists of 7,200 training, 1,800 validation, and 9,000 test images. This dataset exposes the model to characters

CAS	DS	PIP	Detach	ICDAR 2015			MSRA-TD500			Total-Text			CTW1500		
				P	R	F	P	R	F	P	R	F	P	R	F
-	-	-	-	90.9	82.6	86.5	88.9	82.7	85.7	88.7	82.1	85.3	88.6	84.2	86.3
✓	-	-	-	91.2	85.3	88.2	91.5	85.1	88.2	**89.1**	86.5	87.8	**89.8**	85.5	87.6
✓	-	✓	-	89.9	86.6	88.2	91.9	85.7	88.7	88.3	87.2	87.7	89.0	85.5	87.2
✓	-	✓	✓	90.8	85.4	88.0	91.4	85.7	88.5	**89.1**	86.4	87.7	89.7	85.9	87.8
-	✓	-	-	91.1	**87.7**	**89.4**	90.6	**89.0**	89.8	87.9	88.6	88.3	89.1	86.0	87.5
-	✓	✓	-	90.4	85.7	88.0	**93.2**	86.4	89.7	87.9	87.2	87.5	86.9	84.8	85.9
-	✓	✓	✓	**92.5**	86.0	89.2	93.0	88.8	**90.9**	88.0	**89.3**	**88.7**	89.4	**87.7**	**88.5**

Table 2. Overall ablation of split architectures and PIP. CAS and DS mean Cross-Attention Split architecture and Decoder Split architecture, respectively. 'Detach' means blocking gradient flows from the classification branch to the regression branch during backpropagation.

with unique appearances. **ICDAR 2015** [10] and **MSRA-TD500** [23] are both multi-oriented text datasets. The former contains 1,000 training and 500 testing images, while the other contains 300 training and 200 testing images. **Total-Text** [4] and **CTW1500** [13] are both curved text datasets. Total-Text includes 1,255 training and 300 testing images, while CTW1500 includes 1,000 training and 500 testing images.

As in [30], we first pretrain our models with training images of SynthText 150k, MLT17, and Total-Text. After that, we train the models with one of ICDAR 2015, MSRA-TD500, Total-Text, and CTW1500 as a fine-tuning step.

4.2. Implementation Details

In all experiments, ResNet-50 [7] is used as the backbone network. The transformer encoder E and decoders D_c, D_r have 6 layers, and utilize the multi-head deformable attention with 8 heads, capturing 4 sampling offsets. Text instances are represented with 100 queries ($K = 100$) and each query consists of 16 subqueries ($N = 16$). We use ADAMW [15] optimizer with a learning rate of 1×10^{-4} for pre-training, and a learning rate of 1×10^{-5} for fine-tuning with a batch size of 8. The focal loss parameters α and γ are set to 0.25 and 2, respectively. λ_{reg} is fixed at 5.0 for all experiments, while λ_{cls} is determined through hyper-parameter tuning. We use 4 NVIDIA A100 (80G) GPUs for training and 1 GPU for testing.

4.3. Comparison with State-of-the-Art Methods

We evaluate the performance of our proposed method on four standard scene text detection datasets: ICDAR 2015, MSRA-TD500, Total-Text, and CTW1500, comparing it against several state-of-the-art (SOTA) methods. As shown in Tab. 1, our TPD-STR achieves SOTA performance across all datasets without relying on additional data in regression-based methods. Notably, our method surpasses spotter-based models like TESTR, which rely on recognition labels, especially in curved datasets like Total-Text and CTW1500. Moreover, it performs exceptionally well without segmentation masks, which are commonly used in other methods. Even when compared to segmentation-based methods, our approach demonstrates competitive performance, achieving the best results on MSRA-TD500 with an F-score of 90.9%, and consistently delivering strong results on ICDAR 2015, Total-Text, and CTW1500 without the need for extra data.

	DS	PIP (w/ Detach)	Total-Text			CTW1500		
			P	R	F	P	R	F
DPText-DETR (Official)	-	-	91.8	86.4	89.0	**91.7**	86.2	88.8
DPText-DETR (Reproduced)	-	-	**92.1**	85.6	88.7	89.3	88.0	88.6
	✓	-	90.2	87.7	89.0	89.6	**88.3**	88.9
	✓	✓	89.4	**89.0**	**89.2**	90.6	87.5	**89.0**

Table 3. Results of applying DS and PIP to DPText-DETR, with all parameters and datasets set identically as the official implementation.

Additionally, we validate the effectiveness of our DS and PIP modules by integrating them into an existing method, DPText-DETR, which uses additional data. Incorporating our modules leads to performance improvements on all datasets, as shown in the last block of Tab. 1, with detailed ablation studies in Tab. 3. These results confirm that DS and PIP can be easily integrated into other methods, offering significant performance gains. Despite the simplicity of the D-DETR-based structure, our method achieves SOTA performance across various datasets (curved and multi-oriented), demonstrating its robustness and versatility.

4.4. Ablation Studies

4.4.1 Split Architectures.

In place of our decoder split architecture, there can be several candidates as shown in Fig. 3. Accordingly, we evaluate all the candidates for performance comparison. In Tab. 2, 'Cross-Attention Split' (CAS) means an architecture in which only cross-attention is divided in half: polygonal point regression and textness classification module, as shown in Fig. 3 (b). It has been proposed in DESTR [8], and shows a comparable performance. 'Decoder Split (DS)' is our proposed architecture using two entirely separate decoders for regression and classification tasks, respectively, as shown in Fig. 3 (c). 'Baseline' denotes an architecture using shared decoders, as shown in Fig. 3 (a) and its detection score is reported in the first row of Tab. 2. Our method

Figure 4. Qualitative results for scene text detection on ICDAR 2015, MSRA-TD500, Total-Text, and CTW1500.

(DS) tends to outperform CAS in all datasets in terms of F-score. It is noticeable that the more separated architecture within the decoder (DS) generally leads to better performance, with the performance order being Baseline $\leq$ CAS $\leq$ DS, as shown in the first, second, and fifth rows of Tab. 2.

We believe these results are reasonable because the task of polygon detection is more complex than that of box detection. Due to the complexity of polygon detection, we believe that separating decoders for polygonal point regression and textness classification helps each branch focus more on its own task.

4.4.2 Overall Ablation and Discussion.

In Tab. 2, an overall ablation study was conducted for each component we proposed in the previous section.

4.4.3 Information Flow Restriction.

As hypothesized in Sec. 3.3, blocking gradient propagation (Detach) between the classification and regression branches stabilizes the model by focusing the regression branch on localization, while the propagated positional information improves textness prediction in the classification branch. The last two rows in Tab. 2 validate this hypothesis.

4.4.4 Split Architectures and PIP.

As demonstrated in Tab. 2, the Decoder Split architecture combined with PIP regularly outperforms the Cross-Attention Split architecture across all datasets. Notably, PIP shows a more significant impact when paired with Decoder Split on datasets like MSRA-TD500, Total-Text, and CTW1500, highlighting the importance of task separation in leveraging positional information effectively.

In ICDAR 2015, where text instances are typically smaller, the Decoder Split proves particularly effective, but the benefits of PIP are less pronounced due to the limited ability to fully exploit positional information, as shown in Fig. 4 (a). This suggests that Decoder Split alone is better suited for small text detection, while the combination of Decoder Split and PIP excels in handling multi-oriented, curved and larger text instances, such as those in MSRA-TD500, Total-Text, and CTW1500.

4.4.5 Effects of Modules on Different Datasets.

As shown in Tab. 2, Decoder Split (DS) improves F-score by 2.9%p in ICDAR 2015, while PIP (with detach) shows a greater impact on Total-Text (3.4%p) and CTW1500 (2.2%p). In MSRA-TD500, DS achieves a 4.1%p increase,

with the combination of DS and PIP further boosting it to 5.2%p, indicating that PIP is beneficial for both curved and multi-oriented datasets.

These differences are attributed to dataset characteristics. For example, ICDAR 2015, with its small text instances, sees less impact from PIP, whereas MSRA-TD500 benefits from both DS and PIP, as it contains larger text and multi-oriented structures. This shows that PIP is especially effective for datasets containing larger or curved text, such as MSRA-TD500, Total-Text, and CTW1500.

4.4.6 Pluggability of DS and PIP.

To assess the pluggability of our proposed DS and PIP modules, we integrated them into DPText-DETR [25], a D-DETR-based text detector. We kept all the parameters and datasets identical to the official implementation, except for those required for the operation of the DS and PIP modules. As shown in Tab. 3, our method achieves better performance across all datasets when applied to DPText-DETR. Notably, although the best performance reported by DPText-DETR could not be fully reproduced (compare the first two rows in Tab. 3), our results represent the state-of-the-art performance among regression-based methods, as indicated in Tab. 1. However, since DPText-DETR uses additional data, we reported the results in the last block for a fair comparison in Tab. 1. These results demonstrate the flexibility and effectiveness of our proposed modules for improving the performance of existing text detectors.

4.5. Qualitative Results

4.5.1 Detection Performance.

We visualize scene text detection results for multi-oriented and curved text instances from ICDAR 2015, MSRA-TD500, Total-Text, and CTW1500. To highlight the improvement, we compare them with results from the baseline model (shared decoder in Fig. 3), which achieves F-scores of 86.5%, 85.7%, 85.3%, and 86.3%, respectively. As shown in Fig. 4, both models regress the boundaries of multi-oriented and curved text. However, the baseline struggles to detect blurry and small text in ICDAR 2015, while our method successfully detects them. Similarly, for MSRA-TD500, the baseline misses some smaller or less prominent instances, whereas our method accurately identifies them. In Total-Text and CTW1500, the baseline encounters difficulties with curved text due to lower confidence, but our method consistently detects all instances. By separating regression and classification tasks, our method exhibits greater robustness, particularly in handling both multi-oriented and curved text, as seen in MSRA-TD500, Total-Text, and CTW1500.

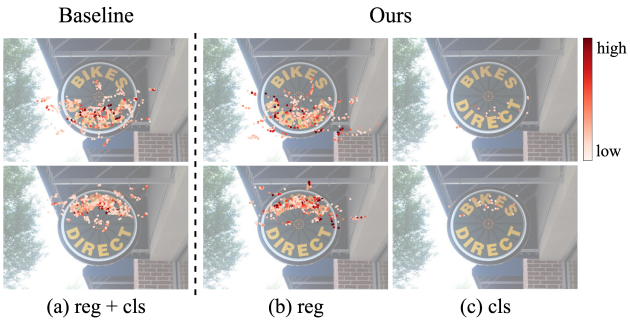


Figure 5. Visualization of deformable attention of the last layer in decoders. Baseline has a single composite decoder conducting regression and classification (reg + cls). Our method has two decoders for regression (reg) and classification (cls). A darker color implies a stronger attention value.

4.5.2 Attention Visualization.

We utilize deformable attention from D-DETR [32] to effectively handle multi-scaled text instances. This enables us to visualize the sampling locations and attention weights in the final layer of the transformer decoder for both branches, as shown in Fig. 5, where we compare them with the baseline model. In our method, the sampling locations for *reg* are more densely distributed along the text boundaries than in the baseline, and the attention intensity is more dispersed, clearly distinguishing text from non-text instances. Interestingly, the sampling locations for *cls* are coarser and more widely spread compared to the regression branch, highlighting how the two tasks perceive text context differently.

5. Conclusion

Our proposed method, TPD-STR, achieves state-of-the-art performance among regression-based methods across all benchmark datasets without requiring additional data, such as positional labels or segmentation maps. It also delivers competitive performance, achieving SOTA on MSRA-TD500, against segmentation-based methods. The Decoder Split (DS) architecture separates the tasks of polygonal point regression and textness classification, while the Positional Information Propagation (PIP) module improves classification by transferring geometric information. Unlike other methods that degrade without extra data, TPD-STR consistently delivers strong performance, and both DS and PIP can be easily integrated into other models. We believe TPD-STR has strong potential for practical applications in scene text detection without additional labeled data.

Acknowledgements. Nojun Kwak was supported by NRF (2021R1A2C3006659) and IITP grants (RS-2021-II211343, RS-2022-II220320) , all funded by the Korean Government.

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, 5

[2] Qingwen Bu, Sungrae Park, Minsoo Khang, and Yichuan Cheng. Srformer: Text detection transformer with incorporated segmentation and regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 855–863, 2024. 2, 3, 4, 5

[3] 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. Springer, 2020. 1, 3

[4] Chee-Kheng Ch'ng, Chee Seng Chan, and Cheng-Lin Liu. Total-text: toward orientation robustness in scene text detection. *International Journal on Document Analysis and Recognition (IJDAR)*, 23(1):31–52, 2020. 6

[5] Tongkun Guan, Chaochen Gu, Changsheng Lu, Jingzheng Tu, Qi Feng, Kaijie Wu, and Xinping Guan. Industrial scene text detection with refined feature-attentive network. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(9):6073–6085, 2022. 2, 4, 5

[6] Tongkun Guan, Wei Shen, Xue Yang, Xuehui Wang, and Xiaokang Yang. Bridging synthetic and real worlds for pre-training scene text detectors. In *Proceedings of the European conference on computer vision (ECCV)*, 2024. 2, 4, 5

[7] 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

[8] Liqiang He and Sinisa Todorovic. Destr: Object detection with split transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9377–9386, 2022. 4, 6

[9] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8813–8822, 2021. 5

[10] 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 *2015 13th international conference on document analysis and recognition (ICDAR)*, pages 1156–1160. IEEE, 2015. 6

[11] Minghui Liao, Zhaoyi Wan, Cong Yao, Kai Chen, and Xiang Bai. Real-time scene text detection with differentiable binarization. In *Proc. AAAI*, 2020. 2, 5

[12] 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. 5

[13] 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, 2019. 6

[14] Shangbang Long, Jiaqiang Ruan, Wenjie Zhang, Xin He, Wenhao Wu, and Cong Yao. Textsnake: A flexible representation for detecting text of arbitrary shapes. In *Proceedings of the European conference on computer vision (ECCV)*, pages 20–36, 2018. 2, 5

[15] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018. 6

[16] 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 *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 1, pages 1454–1459. IEEE, 2017. 5

[17] Zobeir Raisi, Mohamed A Naiel, Georges Younes, Steven Wardell, and John S Zelek. Transformer-based text detection in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3162–3171, 2021. 1, 3, 5

[18] Yipeng Sun, Zihan Ni, Chee-Kheng Chng, Yuliang Liu, Canjie Luo, Chun Chet Ng, Junyu Han, Errui Ding, Jingtuo Liu, Dimosthenis Karatzas, et al. Icdar 2019 competition on large-scale street view text with partial labeling-rrc-lsvt. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1557–1562. IEEE, 2019. 5

[19] Jingqun Tang, Wenqing Zhang, Hongye Liu, MingKun Yang, Bo Jiang, Guanglong Hu, and Xiang Bai. Few could be better than all: Feature sampling and grouping for scene text detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4563–4572, 2022. 1, 2, 3, 4, 5

[20] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. 1

[21] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8440–8449, 2019. 5

[22] 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 *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11753–11762, 2020. 5

[23] Cong Yao, Xiang Bai, Wenyu Liu, Yi Ma, and Zhuowen Tu. Detecting texts of arbitrary orientations in natural images. In *2012 IEEE conference on computer vision and pattern recognition*, pages 1083–1090. IEEE, 2012. 6

[24] Jian Ye, Zhe Chen, Juhua Liu, and Bo Du. Textfusenet: Scene text detection with richer fused features. In Christian Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 516–522. International Joint Conferences on Artificial Intelligence Organization, 7 2020. Main track. 2

[25] Maoyuan Ye, Jing Zhang, Shanshan Zhao, Juhua Liu, Bo Du, and Dacheng Tao. Dptext-detr: Towards better scene text detection with dynamic points in transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 3241–3249, 2023. 1, 3, 4, 5, 8

[26] Chengquan Zhang, Borong Liang, Zuming Huang, Mengyi En, Junyu Han, Errui Ding, and Xinghao Ding. Look more than once: An accurate detector for text of arbitrary shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10552–10561, 2019. 2

[27] Shi-Xue Zhang, Chun Yang, Xiaobin Zhu, and Xu-Cheng Yin. Arbitrary shape text detection via boundary transformer. *IEEE Transactions on Multimedia*, 26:1747–1760, 2023. 1, 2, 3, 4, 5

[28] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9699–9708, 2020. 5

[29] Shi-Xue Zhang, Xiaobin Zhu, Chun Yang, Hongfa Wang, and Xu-Cheng Yin. Adaptive boundary proposal network for arbitrary shape text detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1305–1314, 2021. 5

[30] Xiang Zhang, Yongwen Su, Subarna Tripathi, and Zhuowen Tu. Text spotting transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9519–9528, 2022. 1, 2, 3, 4, 5, 6

[31] 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

[32] 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*, 2020. 1, 2, 3, 8

[33] Yiqin Zhu, Jianyong Chen, Lingyu Liang, Zhuanghui Kuang, Lianwen Jin, and Wayne Zhang. Fourier contour embedding for arbitrary-shaped text detection. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3122–3130, 2021. 2