Bridging the Gap Between End-to-End and Two-Step Text Spotting

Mingxin Huang[1] Hongliang Li[1] Yuliang Liu[2] Xiang Bai[2] Lianwen Jin[1,3*]
[1]South China University of Technology [2]Huazhong University of Science and Technology
[3]INTSIG-SCUT Joint Lab on Document Analysis and Recognition
eelwjin@scut.edu.cn

Abstract

Modularity plays a crucial role in the development and maintenance of complex systems. While end-to-end text spotting efficiently mitigates the issues of error accumulation and sub-optimal performance seen in traditional two-step methodologies, the two-step methods continue to be favored in many competitions and practical settings due to their superior modularity. In this paper, we introduce Bridging Text Spotting, a novel approach that resolves the error accumulation and suboptimal performance issues in two-step methods while retaining modularity. To achieve this, we adopt a well-trained detector and recognizer that are developed and trained independently and then lock their parameters to preserve their already acquired capabilities. Subsequently, we introduce a Bridge that connects the locked detector and recognizer through a zero-initialized neural network. This zero-initialized neural network, initialized with weights set to zeros, ensures seamless integration of the large receptive field features in detection into the locked recognizer. Furthermore, since the fixed detector and recognizer cannot naturally acquire end-to-end optimization features, we adopt the Adapter to facilitate their efficient learning of these features. We demonstrate the effectiveness of the proposed method through extensive experiments: Connecting the latest detector and recognizer through Bridging Text Spotting, we achieved an accuracy of 83.3% on Total-Text, 69.8% on CTW1500, and 89.5% on ICDAR 2015. The code is available at https://github.com/mxin262/Bridging-Text-Spotting.

1. Introduction

Text spotting, as a critical technology for reading text in natural scenes, has garnered significant attention in recent years, owing to its diverse real-world applications, including autonomous driving [65], intelligent navigation [48, 55],

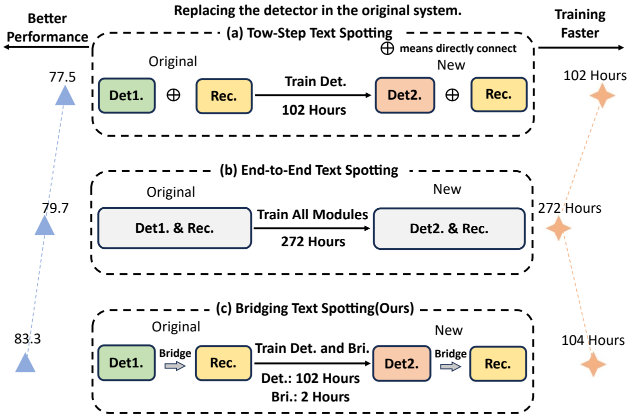


Figure 1. Comparison between the proposed paradigm with existing text spotting paradigms. Our pipeline achieves better performance with high modularity. We adopt the latest detector [68] and text spotter [64] to test the training time of the two-step and end-to-end methods, respectively. The training time is evaluated on the RTX-3090. Det1. means the original detector. Det2. means the new detector. Rec. means the text recognizer. Bri. mean the proposed Bridge.

and visual information extraction [4, 23, 52]. Traditional two-step text spotting separates text detection and recognition into two independent models [27]. In the first step, the focus is on detecting the text regions within a natural scene image. Once the text regions have been located, the second step involves cropping these regions within the image and employing a recognition model to recognize the text contained within these cropped regions.

Recently, many researchers have integrated text detection and recognition within an end-to-end trainable framework, aiming to address the issues of sub-optimal performance and error propagation [15, 26, 31, 36]. Dominant end-to-end text spotting methods mainly follow the detection-by-recognition paradigm [26, 29, 31, 33, 35]. In these approaches, text detectors are initially used to locate text instances, followed by a Region-of-Interest (RoI) oper-

[*] Corresponding author.

ation that extracts features from the shared backbone for recognition. These methods often require the incorporation of numerous heuristic designs involving RoI operations and post-processing steps [64, 69]. Inspired by the Transformer [53], recent advances [21, 22, 64, 69] develop a Transformer-based text spotting framework to avoid the RoI operation and post-processing steps. Besides, some researchers [15, 16, 64] attempt to enhance the synergy between the detection and recognition.

Despite the significant progress in end-to-end text spotting, many competitions [39, 51, 60, 67] and practical applications [6, 7, 25] still favor the two-step text spotting. One crucial reason for this preference is the high modularity inherent in two-step methods. Modularity refers to breaking up a complex system into discrete pieces. Highly modular text spotting systems allow simultaneous development and independent maintenance of detectors and recognizers, with the flexibility to adjust the amount of training data for each module based on specific requirements [1, 24]. To more intuitively illustrate the advantages of modularity, consider a scenario where we need to replace the detector in the original system. As depicted in Fig. 1, the two-step text spotting approach simply requires training a new detector, which can be completed in approximately 102 hours. In contrast, the end-to-end text spotting method necessitates retraining the entire system, which consumes roughly 272 Hours. Furthermore, when aiming to improve the performance of the detection by increasing training data, unlike two-step methods that only necessitate labeling for detection annotations, end-to-end methods require labeling for both detection and recognition annotations, resulting in higher resource consumption than two-step methods.

In this paper, we introduce a new paradigm for text spotting, termed Bridging Text Spotting, which addresses the issues of error accumulation and sub-optimal performance in two-step methods while retaining modularity. Specifically, Bridging Text Spotting adopts a well-trained detector and recognizer and then locks them to maintain their already acquired capabilities. Then, we propose a *Bridge* to integrate the locked detector and recognizer into a trainable framework by incorporating the large receptive field features from the detection into the locked recognizer. To prevent the recognizer from misinterpreting the detection feature as noise in the early stages of training, we initialize the weights of the input and output layer in the *Bridge* to zeros. Additionally, as the locked detector and recognizer do not inherently possess end-to-end optimization features, we adopt the Adapter [14] to facilitate their efficient learning of these features. When transitioning to a new scenario, the Bridging Text Spotting simply requires training a new detector and *Bridge* with the Adapter. It's worth noting that training the *Bridge* with the Adapter is efficient, as demonstrated in the bottom part of Fig. 1. Benefiting from the utilization of the well-trained detector and recognizer, *Bridge* with Adapter eliminates the need for extensive data and enables a rapid completion of the training process.

In conclusion, the main contributions are three-fold:

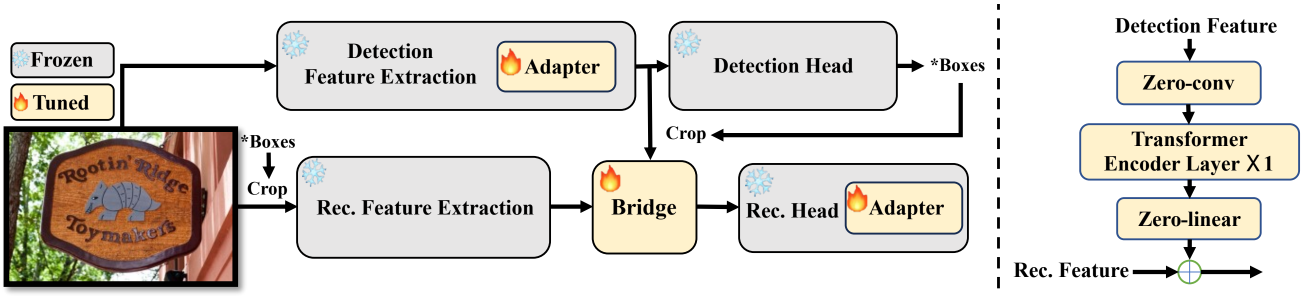
- We introduce a new paradigm for text spotting, termed Bridging Text Spotting, which addresses the issues of error accumulation and sub-optimal performance in two-step spotting while retaining modularity.
- We propose a *Bridge* with the Adapter that enables the well-trained detector and recognizer to learn the end-to-end optimization features based on their already acquired capabilities.
- Experiments demonstrate the effectiveness of the proposed Bridging Text Spotting: 1) Connecting the latest detector and recognizer through Bridging Text Spotting, we achieved an accuracy of 83.3% on Total-Text, 69.8% on CTW1500, and 89.5% on ICDAR 2015; 2) Bridging Text Spotting can achieve an average improvement of 4.4% across multiple combinations of detectors and recognizers.

2. Related Work

Over the past few decades, the advent of deep learning techniques has significantly advanced the field of scene text spotting. Text spotting methods can be broadly classified into two main categories: two-step text spotting and end-to-end text spotting.

Two-Step Text Spotting. Two-step text spotting involves performing detection and recognition through two separate models. The detection model initially locates the text regions, and then the recognition model recognizes the text within these regions. In recent years, Wang *et al.* [56] detect characters by a sliding-window-based detector and subsequently classify each character. Jaderberg *et al.* [18] introduce a method that first detects text instances by generating holistic text proposals with high recall and then recognizes the text content using a word classifier. Liao *et al.* propose TextBoxes++ [28], which incorporate a single-shot detector [27] and a text recognizer [49] in a two-step process. Two-step text spotting methods have a high modularity that allows independent development and maintenance of detectors and recognizers. However, the advancement of two-step text spotting faces constraints due to the accumulation of errors and sub-optimal performance issues [26, 31].

End-to-end Text Spotting. To solve the error accumulation and sub-optimal performance issues, researchers have recently attempted to integrate detection and recognition within an end-to-end trainable framework. Li *et al.* [26] integrate detection and recognition into a unified scene text spotting framework, primarily focusing on horizontal text. In order to handle oriented text, various sampling techniques, including RoIRotate [33] and Text-Align [12], have



(a) Overview

(b) Bridge

Figure 2. The overall architecture of bridging text spotting. Rec. means the recognition. Crop represents the crop operation. The predictions of the detector are used to crop the text regions.

been developed to convert the oriented text into a horizontal grid. The Mask TextSpotter series [29, 31, 42] further utilize character segmentation to handle the arbitrarily-shaped text. Concurrently, arbitrarily-shaped sampling techniques such as RoISlide [10] and BezierAlign [35, 36] are built to rectify the curve texts. Similarly, Wang *et al*. [54] rectify curve texts by the Thin-Plate-Spline [3]. In comparison, [46, 58] use the RoI Masking to connect the detector and recognizer. In order to enhance text recognition performance, Fang *et al*. [9] introduce a language model [8] for text spotting. Additionally, GLASS [47] introduces a global-to-local attention module aimed at improving the capability to read text under varying scales. SRSTS [59] reduces the dependence of recognition on detection through a self-reliant sampling recognition branch.

With the exceptional performance exhibited by the Transformer [53], researchers have started to explore its application in the field of text spotting. Zhang *et al*. [69] adopt a dual decoder framework to represent detection and recognition, respectively. TTS [22] add an RNN-based recognition head on Deformable DETR [70]. SwinTextSpotter [15, 17] propose a Recognition Conversion to enable the back-propagation of recognition information to the detector. To enhance the coherence of detection and recognition, Ye *et al*. [64] develop shared point queries for detection and recognition within a single decoder. SPTS [37, 43] and UNITS [21] treat text spotting as a sequence generation problem. ESTextSpotter [16] further proposes a framework to achieve explicit synergy between two tasks.

While end-to-end text spotting effectively addresses the issues of error accumulation and sub-optimal performance in traditional two-step methodologies, it faces a limitation in leveraging a substantial amount of data that solely comprises detection or recognition annotations. Additionally, end-to-end text spotting cannot directly utilize well-trained detectors and recognizers. Therefore, in many competitions and practical applications, two-step text spotting continues to be the preferred choice due to the high modularity [7, 25, 51, 60, 67].

3. Methodology

Bridging text spotting represents a fresh methodology in the realm of text spotting. It provides a novel solution to address the issues of error propagation and sub-optimal performance in two-step text spotting while preserving modularity. Within the framework of Bridging Text Spotting, the detector and recognizer can be independently developed and trained. Subsequently, they are unified via the proposed *Bridge*.

3.1. Overall Architecture

The overall architecture is depicted in Fig. 2. Initially, we employ a well-trained detector and recognizer, both developed and trained independently. This independent development and training provide flexibility in adjusting training data and structures for individual modules as needed. Subsequently, the parameters of both the detector and recognizer are locked to preserve their already acquired capabilities. Given a scene text image, we send it to the trained detector to locate text instances. Subsequently, the detection results are employed to crop the corresponding regions within the features from the detection backbone and the image. We directly use rectangles to crop the regions. These cropped regions in the image, denoted as $\mathbf{C_i}$, are then fed into the recognition backbone to extract the features. Then, the output from the recognition backbone is sent into the Bridge along with the cropped features $\mathbf{C_f}$ from the detection backbone. In the final step, the output from the Bridge is forwarded to the recognition head, which generates the final recognition results. Additionally, Adapter [14] is integrated into the detection feature extraction and recognition head, facilitating the learning of end-to-end optimization features in both the frozen detector and recognizer.

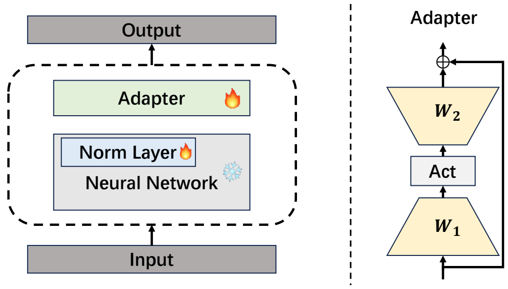


Figure 3. Illustration of Adapter. The neural network refers to the fundamental building blocks of a neural network, such as a multi-head attention block or a transformer block. $\mathbf{W_1}$ and $\mathbf{W_2}$ represent the linear layer. Act means the activation function. All normalization layers in the recognizer are used to tune.

3.2. Bridge

We propose a Bridge to connect the well-trained detector and recognizer, addressing the challenges of error accumulation and sub-optimal performance in two-step spotting while preserving modularity. Note that, we lock (freeze) the parameters of the well-trained detector and the recognizer to preserve their already acquired capabilities. Suppose $R(;\theta_{rb})$ is the recognizer's backbone with parameters θ_{rb} and $R(;\theta_{rh})$ is the recognizer's head with parameters θ_{rh}. Similarly, suppose $D(;\theta_{db})$ is the detector's backbone with parameters θ_{db} and $D(;\theta_{dh})$ is the detector's head with parameters θ_{dh}. Given the input image $\mathbf{I}$, the detection process is as follows:

$$\mathbf{F_{det}} = D(\mathbf{I};\theta_{db})\,, \tag{1}$$

$$\mathbf{P_{det}} = D(\mathbf{F_{det}};\theta_{dh})\,, \tag{2}$$

where $\mathbf{F_{det}}$ represents the features generated by the detection backbone. $\mathbf{P_{det}}$ represents the predictions of the detector. After obtaining the predictions from the detector, we proceed to extract the corresponding regions from the images and features generated by the detection backbone:

$$\mathbf{C_f} = Crop(\mathbf{F_{det}}, \mathbf{P_{det}})\,, \tag{3}$$

$$\mathbf{C_i} = Crop(I, \mathbf{P_{det}})\,, \tag{4}$$

$$\mathbf{F_i} = R(\mathbf{C_i};\theta_{rb})\,, \tag{5}$$

where $Crop$ represents the crop operation. $\mathbf{F_i}$ represents the recognition features generated by the recognition backbone. The recognition features $\mathbf{F_i}$ and the cropped features from detection backbone $\mathbf{C_f}$ are sent to the Bridge. Inspired by [66], we design a zero-initialized convolution and zero-initialized linear layers, whose weight and bias are both initialized to zeros, denoted $Z_c(;)$ and $Z_l(;)$. The process in the Bridge can be formulated as follows:

$$\mathbf{F_r} = \mathbf{F_i} + Z_l(Tr(Z_c((\mathbf{C_f} + PE) : \theta_{zc})) : \theta_{zl})\,, \tag{6}$$

where Tr represents the Transformer encoder. θ_{zc} and θ_{zl} denote the parameters of the zero-initialized convolution and linear layers, which include the weight $\mathbf{W}$ and bias $\mathbf{B}$. $\mathbf{F_r}$ denotes the output of Bridge. During the initial training stage, the weight and bias parameters of the zero-initialized convolution and linear layers are initialized to zero. Consequently, this causes the component $Z_c(;)$ and $Z_l(;)$ in eq. 6 to yield a result of zero. Then, eq. 6 is changed as follows:

$$\mathbf{F_r} = \mathbf{F_i}\,. \tag{7}$$

In this way, the recognition head will not be disturbed by the sudden addition of features in the initial training phase. Although the weight and bias parameters of the convolution and linear layer are initialized to zero, the gradients are non-zero. The gradient calculation of the convolution can be formulated as follows:

$$\begin{cases} \dfrac{\partial Z_c(\mathbf{C_f};\{\mathbf{W},\mathbf{B}\})}{\partial \mathbf{B}} = 1\,, \\ \dfrac{\partial Z_c(\mathbf{C_f};\{\mathbf{W},\mathbf{B}\})}{\partial \mathbf{W}} = \mathbf{C_f}\,, \end{cases} \tag{8}$$

where $\mathbf{C_f}$ is the feature extracted from the image, ensuring it is non-zero. The gradient calculation of the linear layer is similar to the convolution layer. As the training progresses, the weights and biases of the convolution and linear layers gradually adjust to transform the detection features into an adaptive form for the recognition head.

3.3. Adapter

To enhance the synergy between the detector and recognizer in achieving joint optimization, we adopt the Adapter to fine-tune the model, inspired by [14]. The architecture of the Adapter is illustrated in Fig. 3. We keep the parameters of the neural network locked (frozen), with the exception of the normalization layer. The purpose of the normalization layer is to adjust the mean and variance of the distribution of joint optimization features for the adapter. The adapter itself comprises two linear layers and an activation function [13], as represented by the following equation:

$$\mathbf{f_o} = \varphi(\mathbf{f_i}\mathbf{W_1}^T + \mathbf{B_1})\mathbf{W_2}^T + \mathbf{B_2} + \mathbf{f_i}\,, \tag{9}$$

where $\mathbf{W_1}$, $\mathbf{W_2}$, $\mathbf{B_1}$ and $\mathbf{B_2}$ represent the parameters of the linear layers. φ is the activation function. $\mathbf{f_i}$ and $\mathbf{f_o}$ represent the input and output.

3.4. Optimization

For optimization, we follow the original loss in the detector and recognizer. The loss function is the sum of detection loss $\mathcal{L}_{det}$ and recognition loss $\mathcal{L}_{rec}$, formulated as follows:

$$\mathcal{L}_{sum} = \lambda_{det}\mathcal{L}_{det} + \lambda_{rec}\mathcal{L}_{rec}\,, \tag{10}$$

where λ_{det} and λ_{rec} represents a trade-off hyper-parameters and set to 1 in this paper.

Table 1. Scene text spotting results on Total-Text and TextOCR. "None" refers to recognition without lexicon. "Full" lexicon contains all the words in the test set. DB+PARSeq, TESTR-det+MAERec, and DPText-DETR+DiG represent the two-step text spotting using the DB, TESTR's Detector, DPText-DETR as detector and PARSeq, MAERec, and DiG as the recognizer.

Method	Venue	Backbone	Detection			End-to-End		FPS
			P	R	F	None	Full	
DB [30]+PARSeq [2]	–	ResNet-18	89.3	78.4	83.5	69.1	79.1	28.2
TESTR-det [69]+MAERec [19]	–	ResNet-50	92.8	81.3	87.3	78.0	86.6	4.2
DPText-DETR [63]+DiG [62]	–	ResNet-50	91.2	86.3	88.7	77.5	87.6	7.1
Text Dragon [10]	ICCV'2019	VGG16	85.6	75.7	80.3	48.8	74.8	–
Boundary TextSpotter [54]	AAAI'2020	ResNet-50-FPN	88.9	85.0	87.0	65.0	76.1	–
Unconstrained [46]	ICCV'2019	ResNet-50-MSF	83.3	83.4	83.3	67.8	–	–
Text Perceptron [44]	AAAI'2020	ResNet-50-FPN	88.8	81.8	85.2	69.7	78.3	–
Mask TextSpotter v3 [29]	ECCV'2020	ResNet-50-FPN	–	–	–	71.2	78.4	–
ABCNet [35]	CVPR'2020	ResNet	–	–	64.2	75.7	17.9	
ABCNet v2 [36]	TPAMI'2022	ResNet-50-BiFPN	90.2	84.1	87.0	70.4	78.1	10
MANGO [45]	AAAI'2021	ResNet-50-FPN	–	–	–	72.9	83.6	4.3
PGNet [57]	AAAI'2021	ResNet-50-FPN	85.5	86.8	86.1	63.1	–	35.5
TESTR [69]	CVPR'2022	ResNet-50	93.4	81.4	86.9	73.3	83.9	5.3
TTS (poly) [22]	CVPR'2022	ResNet-50	–	–	–	78.2	86.3	–
SwinTextSpotter [15]	CVPR'2022	Swin-Tiny	–	–	88.0	74.3	84.1	–
ABINet++ [9]	TPAMI'2022	ResNet-50-BiFPN	–	–	-	77.6	84.5	10.6
SRSTS [59]	ACMMM'2022	ResNet-50	92.0	83.0	87.2	78.8	86.3	18.7
GLASS [47]	ECCV'2022	ResNet-50	90.8	85.5	88.1	79.9	86.2	3.0
SPTS v2 [37]	TPAMI'2023	ResNet-50	–	–	–	75.5	84.0	–
DeepSolo [64]	CVPR'2023	ResNet-50	93.1	82.1	87.3	79.7	87.0	17.0
UNITS [21]	CVPR'2023	Swin-Base	–	–	89.8	78.7	86.0	–
ESTextSpotter [16]	ICCV'2023	ResNet-50	92.0	88.1	90.0	80.8	87.1	4.3
DG-Bridge Spotter	–	ResNet-50	92.0	86.5	89.2	**83.3**	**88.3**	6.7

4. Experiments

4.1. Implementation Details

We use the DPText-DETR [63] as the well-trained detector and the DiG [62] as the well-trained recognizer in this paper, which is termed DG-Bridge Spotter. Official open-source weights for both the detector and recognizer are utilized. The adapter is incorporated into the Transformer encoder layer [70] in the detector and decoder in the recognizer. We also attempt other combinations of detectors and recognizers to verify the effectiveness of our method, as described in Sec. 4.3. We utilize the AdamW [41] optimizer to optimize the Bridge and Adapter in DG-Bridge Spotter. The Bridge and Adapter are tuned on the training data of the target set. We train the Bridge and Adapter for 10,000 iterations, employing a batch size of eight images. The inference speed is tested on a single NVIDIA GeForce RTX 3090. The data augmentation strategies are similar to those in prior works [35, 36, 69], as detailed below: 1) Random resizing is conducted with the shorter dimension ranging from 480 to 832 pixels, at intervals of 32, while the longer dimension is constrained within 1600 pixels. 2) Random cropping is used, ensuring that text is not cut off. For testing, the shorter dimension of the image is resized to 1000 pixels, while the longer dimension is constrained to a maximum of 1824 pixels.

4.2. Comparison with State-of-the-art Methods

We evaluate our method on several benchmarks, including the multi-oriented benchmark ICDAR2015 [20], the word-level annotated arbitrarily-shaped text benchmark Total-Text [5], and the line-level annotated arbitrarily-shaped text benchmark CTW1500 [34]. For various benchmarks, we replace only the detector and fine-tune the Bridge with the Adapter. In contrast, end-to-end text spotters require training the entire system. Unless otherwise stated, all values in the table are presented as percentages. We also present the results on the TextOCR [50] and HierText [38], in the supplementary material.

Total-Text. For the word-level annotated arbitrarily-shaped text benchmark Total-Text, the results are presented in Tab. 1. We find that our method achieves stronger improvement on the 'None' Lexicon than 'Full'. On the 'None' lexicon, our method correctly recognizes the results that are incorrect in the baseline, leading to higher performance. However, on the 'Full' lexicon, many incorrect results in the baseline can be corrected by the lexicon, resulting in comparable performance with ours. Therefore, our method shows better performance on the 'None' lexicon compared to that on the 'Full' lexicon. Some qualitative results are shown in Fig. 6, demonstrating that our method is capable of recognizing text even in highly curved scenarios.

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 [10]	84.5	82.8	83.6	39.7	72.4
Text Perceptron [44]	87.5	81.9	84.6	57.0	–
ABCNet [35]	–	–	–	45.2	74.1
ABCNet v2 [36]	85.6	83.8	84.7	57.5	77.2
MANGO [45]	–	–	–	58.9	78.7
ABINet++ [9]	–	–	–	60.2	80.3
TESTR [69]	92.0	82.6	87.1	56.0	81.5
SwinTextSpotter [15]	–	–	88.0	51.8	77.0
SPTS v2 [37]	–	–	–	63.6	**84.3**
DeepSolo [64]	–	–	–	64.2	81.4
ESTextSpotter [16]	91.5	88.6	90.0	64.9	83.9
DG-Brigde Spotter	92.1	86.2	89.0	**69.8**	83.9

CTW1500. For the line-level annotated arbitrarily-shaped text benchmark CTW1500, the results are shown in Tab. 7. Our method surpasses the state-of-the-art method by a notable margin of 4.9% in the ‘None’ metrics, unequivocally proving its efficacy in handling long text recognition. Our method outperforms SPTS v2 and ESTextSpotter in ‘None’ metrics while yielding comparable results in ‘Full’ metrics. On the ‘None’ lexicon, our method correctly recognizes the results that are incorrect in the baseline, leading to higher performance. This discrepancy highlights a prevalent issue in these methods, where a small number of characters within a text line are frequently misidentified, necessitating the use of a lexicon for correction. However, in many practical scenarios, lexicons are usually absent.

ICDAR2015. Since DPText-DETR [63] does not provide open-source weights for the ICDAR2015 dataset, we adopt a similar method the detector of TESTR [69] as our detector. It’s important to highlight that we have kept the recognizer DiG [62] unchanged. We refer to this combination of the detector of TESTR and the DiG-based recognizer as the TG-Bridge Spotter. The results are illustrated in Tab. 3. The proposed TG-Bridge Spotter outperforms the state-of-the-art method in all lexicons. Specifically, our method outperforms the ESTextSpotter [16] by 1.6%, 1.2% and 2.3% in terms of ‘Strong’, ‘Weak’, and ‘Generic’ metrics, respectively.

Table 3. Results on ICDAR 2015 dataset. “S”, “W”, “G” represent recognition with “Strong”, “Weak” or “Generic” lexicon, respectively.

Method	Detection			End-to-End		
	P	R	F	S	W	G
FOTS [33]	91.0	85.2	88.0	81.1	75.9	60.8
CharNet R-50 [61]	91.2	88.3	89.7	80.1	74.5	62.2
Boundary TextSpotter [54]	89.8	87.5	88.6	79.7	75.2	64.1
Unconstrained [46]	89.4	85.8	87.5	83.4	79.9	68.0
Text Perceptron [44]	92.3	82.5	87.1	80.5	76.6	65.1
Mask TextSpotter v3 [29]	–	–	–	83.3	78.1	74.2
ABCNet v2 [36]	90.4	86.0	88.1	82.7	78.5	73.0
MANGO [45]	–	–	–	81.8	78.9	67.3
PGNet [57]	91.8	84.8	88.2	83.3	78.3	63.5
ABINet++ [9]	–	–	–	84.1	80.4	75.4
TESTR [69]	90.3	89.7	90.0	85.2	79.4	73.6
TTS [22]	–	–	–	85.2	81.7	77.4
SwinTextSpotter [15]	–	–	–	83.9	77.3	70.5
SPTS v2 [37]	–	–	–	82.3	77.7	72.6
SRSTS [59]	96.1	82.0	88.4	85.6	81.7	74.5
GLASS [47]	86.9	84.5	85.7	84.7	80.1	76.3
DeepSolo [64]	92.8	87.4	90.0	86.8	81.9	76.9
ESTextSpotter [16]	92.5	89.6	91.0	87.5	83.0	78.1
TG-Bridge Spotter	93.8	87.5	90.5	**89.1**	**84.2**	**80.4**

Table 4. Ablation study on Total-Text. DA means using the adapter in the detector. RA means using the adapter in the recognizer. DG/TM/BP-Baseline represent the corresponding two-step pipelines.

Method	Bridge	DA	RA	Detection			E2E	FPS	Param
				P	R	F			
DG-Baseline	–	–	–	91.2	86.3	88.7	77.5	7.1	81.5M
DG-Baseline+	✓	–	–	91.2	86.3	88.7	81.5	7.1	85.0M
DG-Baseline+	✓	✓	–	91.7	86.7	89.1	82.1	7.0	85.2M
DG-Bridge Spotter	✓	✓	✓	92.0	86.5	89.2	**83.3**	6.7	86.0M
TM-Baseline	–	–	–	92.8	81.3	87.3	78.0	4.2	77.4M
TM-Baseline+	✓	–	–	92.8	81.3	87.3	80.8	4.2	80.9M
TM-Baseline+	✓	✓	–	92.3	83.6	87.8	81.1	4.0	81.1M
TM-Bridge Spotter	✓	✓	✓	92.4	82.7	87.3	**81.9**	3.6	81.5M
BP-Baseline	–	–	–	89.3	78.4	83.5	69.1	28.2	37.6M
BP-Baseline+	✓	–	–	89.3	78.4	83.5	70.7	28.0	40.4M
BP-Baseline+	✓	✓	–	88.8	79.7	84.0	70.8	27.0	40.5M
BP-Bridge Spotter	✓	✓	✓	89.1	79.1	83.8	**72.5**	26.5	40.9M

4.3. Ablation Studies

We conduct ablation studies on Total-Text with three combinations of detectors and recognizers to verify the validity of our method. (1) we adopt the DPText-DETR [63] and the DiG [62], termed DG-Bridge Spotter. (2) we attempt another combination that involves the detector of TESTR [69] and MAERec [19], termed TM-Bridge Spotter. (3) we also utilize the combination of segmentation-based text detector DBNet [30] as the detector and a fast decoding recognizer PARSeq [2], dubbed BP-Bridge Spotter. We choose the ResNet18 [11] as the backbone of the DBNet.

Ablation Study of The Bridge. To evaluate the effectiveness of the proposed Bridge and Adapter, we conduct ablation studies on the Total-Text. As shown in Tab. 4, Bridge significantly improves text spotting performance in all three

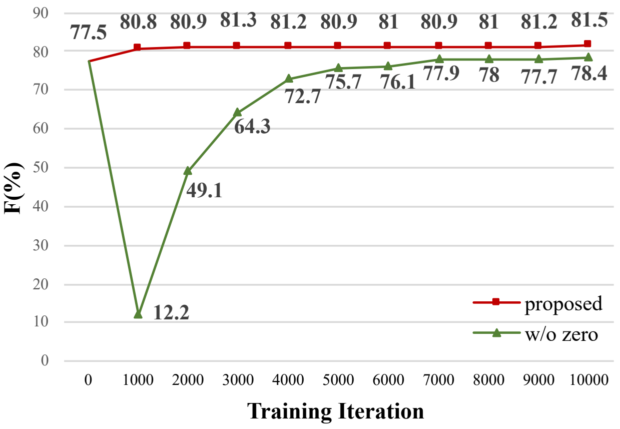


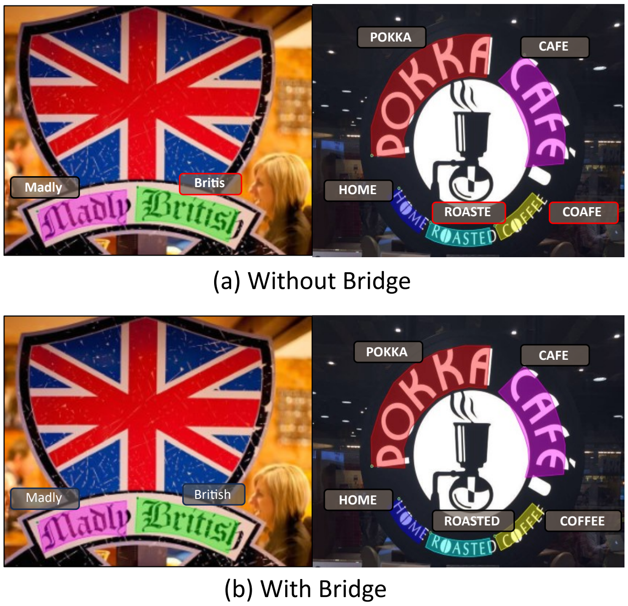
Figure 4. Ablative study of the zero-initialized weight in Bridge. "F" indicates F-measure in end-to-end text spotting results on Total-Text.

Table 5. Ablation study about different Transformer layers in the Bridge.

Num layer	Detection			E2E	FPS	Param
	P	R	F			
0	91.3	86.0	88.6	82.3	2.9	84.2M
1	**92.0**	86.5	**89.2**	**83.3**	2.9	86.0M
3	91.1	**86.6**	88.8	82.6	2.9	89.6M
6	91.8	86.5	89.1	82.9	2.9	94.9M

combinations, with a 4.0% increase in DG-Bridge Spotter, a 2.8% increase in TM-Bridge Spotter, and 1.6% increase in BP-Bridge Spotter. The Bridge successfully merges detection features with large receptive fields and recognition features with high resolution, directing this combined information into a locked recognition head. This process enhances performance significantly, with only a slight reduction in speed and requiring minimal increases in parameters. Moreover, we offer a more intuitive comparison to highlight the efficacy of the Bridge, as depicted in Fig. 5. In scenarios without the Bridge, if the detection results do not align accurately with the text boundaries, it can readily result in inaccurate recognition results. This situation is commonly known as error accumulation. In contrast, the Bridge effectively alleviates this issue by utilizing features with large receptive fields, thereby improving the performance.

Ablation Study of The Adapter. The influence of Adapter is depicted in Tab. 4. Through the integration of the Adapter into both the detector and recognizer, three combinations effectively acquire end-to-end optimization features, thereby improving overall performance. Furthermore, it introduces only a modest number of additional parameters and has a minimal impact on speed.



(a) Without Bridge

(b) With Bridge

Figure 5. Effectiveness of Bridge. Red boxes indicate recognition errors due to inaccurate detection results. Zoom in for best view.

Ablation Study of the Zero-initialized Weight in Bridge. To further investigate the efficacy of the proposed zero-initialized architecture, we conduct an analysis of the alternative Bridge structure by substituting the zero-initialized convolution and linear layers with counterparts initialized using Gaussian weights. In order to control variables, we do not add Adapter in this experiment. The results are illustrated in Fig. 4. The findings reveal that the zero-initialized weight within the Bridge facilitates rapid adaptation of the recognition head to features with large receptive fields, leading to a substantial performance boost (from 77.5% to 81.5%). In contrast, employing a Gaussian-initialized structure initially disrupts the training of the recognition head, resulting in only marginal eventual improvement (from 77.5% to 78.4%).

Ablation Study of The Number of Transformer Layers. To comprehensively validate the impact of the number of Transformer layers in the Bridge, we conduct experiments on Total-Text to compare the performance with varying numbers of Transformer layers in the Bridge. As presented in Tab. 5, we observe that setting the number to 1 resulted in the model achieving the best performance. Despite reducing the number of parameters by $1.8M$, the detection performance dropped by 0.6%, and the text spotting performance decreased by 1%. Further increasing the number of Transformer layers does not lead to performance improve-

Table 6. Comparison of different paradigms on Total-Text. Two-step finetune represents fine-tuning the detector and recognizer, respectively, without Bridge. Row3+Using C_f represents using the C_f as the input of the recognizer. Row4+Using C_i represents using C_f and C_i as the input of the recognizer.

Method	Detection			E2E	FPS	Param
	P	R	F			
Two-step	91.2	86.3	88.7	77.5	7.1	81.5M
End-to-end	91.2	86.1	88.6	75.6	7.5	83.1M
Two-step finetune	89.4	85.7	87.5	78.8	7.1	81.5M
Row3+Using C_f	91.2	86.5	88.8	66.8	7.5	83.1M
Row4+Using C_i	91.0	87.7	89.3	79.8	7.1	83.1M
Ours	92.0	86.5	89.2	**83.3**	6.7	85.2M

ment, but it does increase the model's parameters.

Comparison with Existing Paradigm. To further validate the effectiveness of our method, we conduct experiments on Total-Text, comparing it with two-step and end-to-end text spotting. This experiment uses the DPText-DETR as the detector and DiG as the recognizer. The results are illustrated in Tab. 6. In the case of end-to-end text spotting, we made a modification by adjusting the first layer of the recognition backbone. This adjustment enables a seamless passage of cropped features from the detection backbone to the recognizer. We observed that when using the same detector and recognizer, two-step one outperforms end-to-end one in text spotting metrics, despite their comparable detection performance. This superiority can be attributed to the high modularity of the two-step method. This modularity enables the recognizer to be trained independently with a larger dataset, potentially surpassing the impact of error accumulation and sub-optimization. Additionally, we made an effort to load pre-trained weights in a two-step method to initialize both the detector and recognizer. We also make efforts to fine-tune the detector and recognizer directly on TotalText and connect them in a two-step manner, as illustrated in the third row of Tab. 6. Due to its limitations in effectively mitigating the issues of sub-optimal performance and error accumulation, this approach offers only marginal performance improvements. Subsequently, we load the pre-trained weight of the detector and recognizer and use the C_f as the input of the recognizer, referred to as Row3+Using C_f. The C_f and C_i are detailed in the Sec. 3.1. As shown in Tab. 6, the results indicate a drop in performance. This is because, during pre-training, the inputs of the recognizer in Row3+Using C_f are the images. Consequently, when the input transitions to features in detection, the recognizer misinterprets them as noise, disrupting the fine-tuning process of the recognizer. We further utilize the summation of C_f and C_i as the input of the recognizer, referred to as Row4+Using C_i. The results demonstrate the effectiveness of integrating both detection and recognition features.

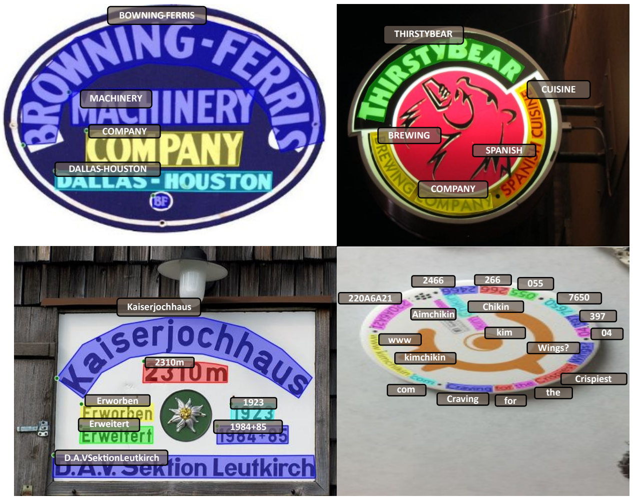


Figure 6. Qualitative results of DG-bridge Spotter on CTW1500 (left column) and Total-Text (right column). Zoom in for best view.

5. Conclusion

In this paper, we introduce a new paradigm for text spotting, termed Bridging Text Spotting, to address the issues of sub-optimal performance and error accumulation in the two-step text spotting while retaining modularity. The proposed Bridge effectively connects the well-trained detector and recognizer through a zero-initialized neural network. The Adapter enables the well-trained detector and recognizer to efficiently learn end-to-end optimization features, thereby improving performance. Extensive experimental results demonstrate the effectiveness of Bridging Text Spotting: 1) Bridging Text Spotting with the latest detector and recognizer outperforms the previous state-of-the-art end-to-end method on various challenge benchmarks. 2) Bridging Text Spotting can consistently enhance performance across various combinations of detectors and recognizers. The proposed method provides an effective way of integrating two distinct modules for end-to-end optimization. In the future, how to connect multi-task modules using our approach to create a robust multi-task system is worthy of further study.

Acknowledgement This research is supported in part by National Natural Science Foundation of China (Grant No.: 62441604, 61936003), National Key Research and Development Program of China (2022YFC3301703)

Appendix

A. Performance on TextOCR and HierText

We conduct experiments on more challenging benchmarks on TextOCR [50] and HierText [38] to verify the effectiveness of our methods. For TextOCR, we adopt a detector similar to the GLASS [47] and use the DiG [62] as the recognizer. For HierText, we utilize the DBNet++ [32] as the detector and MAERec [19] as the recognizer. As shown in Tab. 7, the results demonstrate the effectiveness of our method.

Table 7. End-to-end text spotting results on TextOCR and HierText.

Method	TextOCR	HierText
MaskTextSpotter v3 [29]	50.8	–
GLASS [47]	67.1	–
HTS [40]	–	75.6
Ours	68.5	76.1

References

[1] Carliss Young Baldwin and Kim B Clark. *Design rules: The power of modularity*. MIT press, 2000. 2

[2] Darwin Bautista and Rowel Atienza. Scene text recognition with permuted autoregressive sequence models. In *European Conference on Computer Vision*, pages 178–196, Cham, 2022. Springer Nature Switzerland. 5, 6

[3] Fred L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(6):567–585, 1989. 3

[4] Haoyu Cao, Changcun Bao, Chaohu Liu, Huang Chen, Kun Yin, Hao Liu, Yinsong Liu, Deqiang Jiang, and Xing Sun. Attention where it matters: Rethinking visual document understanding with selective region concentration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19517–19527, 2023. 1

[5] 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. 5

[6] Yuning Du, Chenxia Li, Ruoyu Guo, Xiaoting Yin, Weiwei Liu, Jun Zhou, Yifan Bai, Zilin Yu, Yehua Yang, Qingqing Dang, et al. PP-OCR: A practical ultra lightweight OCR system. *arXiv preprint arXiv:2009.09941*, 2020. 2

[7] Yuning Du, Chenxia Li, Ruoyu Guo, Cheng Cui, Weiwei Liu, Jun Zhou, Bin Lu, Yehua Yang, Qiwen Liu, Xiaoguang Hu, et al. PP-OCRv2: Bag of tricks for ultra lightweight OCR system. *arXiv preprint arXiv:2109.03144*, 2021. 2, 3

[8] Shancheng Fang, Hongtao Xie, Yuxin Wang, Zhendong Mao, and Yongdong Zhang. Read like humans: Autonomous, bidirectional and iterative language modeling for scene text recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7098–7107, 2021. 3

[9] Shancheng Fang, Zhendong Mao, Hongtao Xie, Yuxin Wang, Chenggang Yan, and Yongdong Zhang. ABINet++: Autonomous, bidirectional and iterative language modeling for scene text spotting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–18, 2022. 3, 5, 6

[10] 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*, pages 9076–9085, 2019. 3, 5, 6

[11] 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

[12] 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 the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5020–5029, 2018. 2

[13] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016. 4

[14] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR, 2019. 2, 3, 4

[15] Mingxin Huang, Yuliang Liu, Zhenghao Peng, Chongyu Liu, Dahua Lin, Shenggao Zhu, Nicholas Yuan, Kai Ding, and Lianwen Jin. SwinTextSpotter: Scene text spotting via better synergy between text detection and text recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4593–4603, 2022. 1, 2, 3, 5, 6

[16] Mingxin Huang, Jiaxin Zhang, Dezhi Peng, Hao Lu, Can Huang, Yuliang Liu, Xiang Bai, and Lianwen Jin. ESTextSpotter: Towards better scene text spotting with explicit synergy in transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19495–19505, 2023. 2, 3, 5, 6

[17] Mingxin Huang, Dezhi Peng, Hongliang Li, Zhenghao Peng, Chongyu Liu, Dahua Lin, Yuliang Liu, Xiang Bai, and Lianwen Jin. Swintextspotter v2: Towards better synergy for scene text spotting. *arXiv preprint arXiv:2401.07641*, 2024. 3

[18] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Reading text in the wild with convolutional neural networks. *International journal of computer vision*, 116:1–20, 2016. 2

[19] Qing Jiang, Jiapeng Wang, Dezhi Peng, Chongyu Liu, and Lianwen Jin. Revisiting scene text recognition: A data perspective. In *Proceedings of the IEEE/CVF International*

Conference on Computer Vision, pages 20543–20554, 2023. 5, 6, 9

[20] 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. 5

[21] Taeho Kil, Seonghyeon Kim, Sukmin Seo, Yoonsik Kim, and Daehee Kim. Towards unified scene text spotting based on sequence generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15223–15232, 2023. 2, 3, 5

[22] Yair Kittenplon, Inbal Lavi, Sharon Fogel, Yarin Bar, R Manmatha, and Pietro Perona. Towards weakly-supervised text spotting using a multi-task transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4604–4613, 2022. 2, 3, 5, 6

[23] Jianfeng Kuang, Wei Hua, Dingkang Liang, Mingkun Yang, Deqiang Jiang, Bo Ren, and Xiang Bai. Visual information extraction in the wild: practical dataset and end-to-end solution. In *International Conference on Document Analysis and Recognition*, pages 36–53. Springer, 2023. 1

[24] Richard N Langlois. Modularity in technology and organization. *Journal of economic behavior & organization*, 49(1): 19–37, 2002. 2

[25] Chenxia Li, Weiwei Liu, Ruoyu Guo, Xiaoting Yin, Kaitao Jiang, Yongkun Du, Yuning Du, Lingfeng Zhu, Baohua Lai, Xiaoguang Hu, et al. PP-OCRv3: More attempts for the improvement of ultra lightweight OCR system. *arXiv preprint arXiv:2206.03001*, 2022. 2, 3

[26] 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

[27] 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. 1, 2

[28] 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

[29] Minghui Liao, Guan Pang, Jing Huang, Tal Hassner, and Xiang Bai. Mask textspotter v3: Segmentation proposal network for robust scene text spotting. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pages 706–722. Springer, 2020. 1, 3, 5, 6, 9

[30] 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*, pages 11474–11481, 2020. 5, 6

[31] M. Liao, P. Lyu, M. He, C. Yao, W. Wu, and X. Bai. Mask TextSpotter: An end-to-end trainable neural network for spotting text with arbitrary shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(2):532–548, 2021. 1, 2, 3

[32] Minghui Liao, Zhisheng Zou, Zhaoyi Wan, Cong Yao, and Xiang Bai. Real-time scene text detection with differentiable binarization and adaptive scale fusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 9

[33] 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*, pages 5676–5685, 2018. 1, 2, 6

[34] 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. 5

[35] 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, 3, 5, 6

[36] 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*, 44(11):8048–8064, 2021. 1, 3, 5, 6

[37] Yuliang Liu, Jiaxin Zhang, Dezhi Peng, Mingxin Huang, Xinyu Wang, Jingqun Tang, Can Huang, Dahua Lin, Chunhua Shen, Xiang Bai, et al. SPTS v2: Single-point scene text spotting. *arXiv preprint arXiv:2301.01635*, 2023. 3, 5, 6

[38] Shangbang Long, Siyang Qin, Dmitry Panteleev, Alessandro Bissacco, Yasuhisa Fujii, and Michalis Raptis. Towards end-to-end unified scene text detection and layout analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1049–1059, 2022. 5, 9

[39] Shangbang Long, Siyang Qin, Dmitry Panteleev, Alessandro Bissacco, Yasuhisa Fujii, and Michalis Raptis. ICDAR 2023 Competition on Hierarchical Text Detection and Recognition. *arXiv preprint arXiv:2305.09750*, 2023. 2

[40] Shangbang Long, Siyang Qin, Yasuhisa Fujii, Alessandro Bissacco, and Michalis Raptis. Hierarchical text spotter for joint text spotting and layout analysis. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 903–913, 2024. 9

[41] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 5

[42] 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)*, pages 67–83, 2018. 3

[43] Dezhi Peng, Xinyu Wang, Yuliang Liu, Jiaxin Zhang, Mingxin Huang, Songxuan Lai, Jing Li, Shenggao Zhu, Dahua Lin, Chunhua Shen, et al. SPTS: single-point text spotting. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 4272–4281, 2022. 3

[44] 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, pages 11899–11907, 2020. 5, 6

[45] 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. In *Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI)*, pages 2467–2476, 2021. 5, 6

[46] 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*, pages 4704–4714, 2019. 3, 5, 6

[47] Roi Ronen, Shahar Tsiper, Oron Anschel, Inbal Lavi, Amir Markovitz, and R Manmatha. GLASS: Global to local attention for scene-text spotting. In *European Conference on Computer Vision*, pages 249–266. Springer, 2022. 3, 5, 6, 9

[48] Xuejian Rong, Bing Li, J Pablo Munoz, Jizhong Xiao, Aries Arditi, and Yingli Tian. Guided text spotting for assistive blind navigation in unfamiliar indoor environments. In *International Symposium on Visual Computing*, pages 11–22. Springer, 2016. 1

[49] 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

[50] 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 the IEEE/CVF conference on computer vision and pattern recognition*, pages 8802–8812, 2021. 5, 9

[51] 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. 2, 3

[52] Jordy Van Landeghem, Rubèn Tito, Łukasz Borchmann, Michał Pietruszka, Pawel Joziak, Rafal Powalski, Dawid Jurkiewicz, Mickaël Coustaty, Bertrand Anckaert, Ernest Valveny, et al. Document understanding dataset and evaluation (dude). In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19528–19540, 2023. 1

[53] 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. 2, 3

[54] 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*, pages 12160–12167, 2020. 3, 5, 6

[55] Hsueh-Cheng Wang, Chelsea Finn, Liam Paull, Michael Kaess, Ruth Rosenholtz, Seth Teller, and John Leonard. Bridging text spotting and slam with junction features. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3701–3708. IEEE, 2015. 1

[56] Kai Wang, Boris Babenko, and Serge Belongie. End-to-end scene text recognition. In *2011 International Conference on Computer Vision*, pages 1457–1464. IEEE, 2011. 2

[57] 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. 5, 6

[58] 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. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 3

[59] Jingjing Wu, Pengyuan Lyu, Guangming Lu, Chengquan Zhang, Kun Yao, and Wenjie Pei. Decoupling recognition from detection: Single shot self-reliant scene text spotter. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 1319–1328, 2022. 3, 5, 6

[60] Weijia Wu, Yuzhong Zhao, Zhuang Li, Jiahong Li, Mike Zheng Shou, Umapada Pal, Dimosthenis Karatzas, and Xiang Bai. ICDAR 2023 Competition on Video Text Reading for Dense and Small Text. In *International Conference on Document Analysis and Recognition*, pages 405–419. Springer, 2023. 2, 3

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

[62] Mingkun Yang, Minghui Liao, Pu Lu, Jing Wang, Shenggao Zhu, Hualin Luo, Qi Tian, and Xiang Bai. Reading and writing: Discriminative and generative modeling for self-supervised text recognition. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 4214–4223, 2022. 5, 6, 9

[63] 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*, 2023. 5, 6

[64] Maoyuan Ye, Jing Zhang, Shanshan Zhao, Juhua Liu, Tongliang Liu, Bo Du, and Dacheng Tao. DeepSolo: Let transformer decoder with explicit points solo for text spotting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19348–19357, 2023. 1, 2, 3, 5, 6

[65] Chongsheng Zhang, Yuefeng Tao, Kai Du, Weiping Ding, Bin Wang, Ji Liu, and Wei Wang. Character-level street view text spotting based on deep multi-segmentation network for smarter autonomous driving. *IEEE Transactions on Artificial Intelligence*, pages 1–1, 2021. 1

[66] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023. 4

[67] Rui Zhang, Yongsheng Zhou, Qianyi Jiang, Qi Song, Nan Li, Kai Zhou, Lei Wang, Dong Wang, Minghui Liao, Mingkun

Yang, et al. Icdar 2019 robust reading challenge on reading chinese text on signboard. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1577–1581. IEEE, 2019. 2, 3

[68] Shi-Xue Zhang, Chun Yang, Xiaobin Zhu, and Xu-Cheng Yin. Arbitrary shape text detection via boundary transformer. *IEEE Transactions on Multimedia*, 2023. 1

[69] 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. 2, 3, 5, 6

[70] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: Deformable transformers for end-to-end object detection. *ICLR*, pages 1–9, 2021. 3, 5