Visual Semantics Allow for Textual Reasoning Better in Scene Text Recognition

Yue He[1], Chen Chen[2], Jing Zhang[2], Juhua Liu[3]*, Fengxiang He[4], Chaoyue Wang[2], Bo Du[1]*

[1] National Engineering Research Center for Multimedia Software, Institute of Artificial Intelligence, School of Computer Science and Hubei Key Laboratory of Multimedia and Network Communication Engineering, Wuhan University, China
[2] School of Computer Science, Faculty of Engineering, The University of Sydney, Australia
[3] School of Printing and Packaging, and Institute of Artificial Intelligence, Wuhan University, China
[4] JD Explore Academy, China
{yuehe.cs, liujuhua, dubo}@whu.edu.cn, cche9000@uni.sydney.edu.au, {jing.zhang1, chaoyue.wang}@sydney.edu.au, hefengxiang@jd.com

Abstract

Existing Scene Text Recognition (STR) methods typically use a language model to optimize the joint probability of the 1D character sequence predicted by a visual recognition (VR) model, which ignore the 2D spatial context of visual semantics within and between character instances, making them not generalize well to arbitrary shape scene text. To address this issue, we make the first attempt to perform textual reasoning based on visual semantics in this paper. Technically, given the character segmentation maps predicted by a VR model, we construct a subgraph for each instance, where nodes represent the pixels in it and edges are added between nodes based on their spatial similarity. Then, these subgraphs are sequentially connected by their root nodes and merged into a complete graph. Based on this graph, we devise a graph convolutional network for textual reasoning (GTR) by supervising it with a cross-entropy loss. GTR can be easily plugged in representative STR models to improve their performance owing to better textual reasoning. Specifically, we construct our model, namely S-GTR, by paralleling GTR to the language model in a segmentation-based STR baseline,which can effectively exploit the visual-linguistic complementarity via mutual learning. S-GTR sets new state-of-the-art on six challenging STR benchmarks and generalizes well to multi-linguistic datasets. Code is available at https://github.com/adeline-cs/GTR.

Introduction

Scene Text Recognition (STR) remains a fundamental and active research topic in computer vision for its wide applications (Zhang and Tao 2020). However, this task remains challenging in real-world deployment, since the recognition results are highly influenced by various factors, such as complex background, irregular shapes, diverse textures.

Existing methods mainly treat STR as a visual recognition (VR) task and perform character-level recognition on input images, including visual-text sequence translation-based methods (Yang et al. 2017; Shi et al. 2018; Baek et al. 2019; Li et al. 2019; Litman et al. 2020) and semantic segmentation-based methods (Liao et al. 2019; Wan et al. 2020a). Although these methods obtain reasonable performance on identifying individual characters, they ignore vital global textual representations, making it extremely hard to give robust recognition outcomes in real-world scenarios.

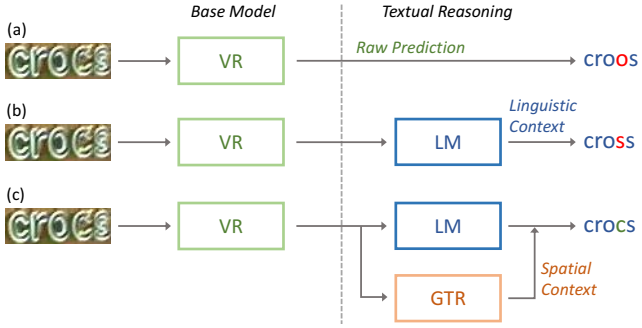

Figure 1: Diagram of different STR pipelines. (a) The VR model. (b) The VR model with an LM. (c) The proposed pipeline by adding GTR in parallel with LM. GTR performs textual reasoning based on the visual semantics generated by VR to address the irregular and blurry text. The ground truth label is "crocs" and wrong predictions are marked in red.

For global textual modeling, existing efforts (Qiao et al. 2020; Yu et al. 2020; Fang et al. 2021) have been made to leverage a language model (LM) (Jaderberg et al. 2014a) to optimize the joint probability of the character sequence predicted by the VR model. Though this strategy can correct mistaken predictions with linguistic context, it is hard to generalize to arbitrary texts and ambiguous cases. As shown in Fig. 1(b), for the irregular and blurry text, even LM could not make correct predictions. Other than linguistic cues, spatial context could also contribute to global textual modeling of character sequences but few methods explore in this direction. Hence, existing models have difficulty in producing satisfactory results on irregular texts in diverse fonts and shapes as well as with blur and occlusions.

In this paper, we fill this gap with a novel **G**raph-based **T**extual **R**easoning (GTR) model for introducing spatial context into the text reasoning stage. Given the character instances recognized by the VR model as well as the derived order relations between them, we first set up a two-level graph to establish the local-to-global dependency. In

*Corresponding author. This work was done during Yue He's internship at JD Explore Academy.

the first level, we construct a subgraph for pixels within each character instance based on their spatial similarity. And for the second level, 1-st level subgraphs are merged into a complete graph by linking their root nodes, which represent the geometric center of all nodes within each subgraph. Accordingly, we devise a graph convolutional neural network for context reasoning and optimizing the joint probability of the character sequence.

Our proposed GTR is an easy-to-plug-in module and can seamlessly function with other context modalities. Specifically, we put GTR parallel to the LM to produce joint features for text reasoning (as shown in Fig. 1(c)). To produce high-quality cross-modality representations, we design a mutual learning protocol to enforce the consistency between predictions from LM and GTR and employ a dynamic fusion strategy (Yue et al. 2020) to deeply combine visual and linguistic features. Based on these designs, GTR can largely boost the text reasoning performance comparing to existing representative methods with LM only.

We incorporate all aforementioned designs into a segmentation-based STR baseline and propose S-GTR, a unified framework of **S**egmentation baseline with **GTR**. We evaluate S-GTR on multiple datasets with both regular and irregular text material in different languages. Experimental results show that our S-GTR outperforms previous methods and obtains state-of-the-art performance on six challenging benchmarks. In summary, the contribution of this work is threefold:

- We propose a novel graph-based textual reasoning model named GTR to refine coarse text sequence predictions with spatial context. It is a complementary design to the popular reasoning fashion with LM only in existing representative methods, and can further improve their performance by acting as an easy-to-plug-in module.
- To make GTR work with LM compatibly, we further employ a mutual learning protocol and propose a dynamic fusion strategy to produce consistent linguistic and visual representations and high-quality joint prediction.
- We put all our designs in a unified framework (S-GTR) of segmentation baseline with GTR. Extensive experimental results indicate our S-GTR successfully sets new state-of-the-art for regular and irregular text recognition tasks as well as shows a superiority on both English and Chinese text materials.

Related Work

Arbitrary-shaped Scene Text Recognition. Existing STR methods for recognizing texts of arbitrary shapes can be divided into two main categories, *i.e.*, rectification-based methods and segmentation-based methods. The former methods (Gao et al. 2018; Yang et al. 2017; Cheng et al. 2018) use the spatial transformer network (Jaderberg et al. 2015) to normalize text images into the canonical shape, $i.e.$, horizontally aligned characters of uniform heights and widths. These methods, however, are limited by the predefined text transformation set and hard to generalize to real-world examples. The latter methods (Liao et al. 2019; Wan et al. 2020a) follow the common processing fashion in the text detection task (Ye et al. 2021) and formulate the recognition task as an instance segmentation problem, where texts are explicitly modeled into instance masks. In this way, it can efficiently represent irregular texts in different fonts, scales, orientation, and shapes, as well as with occlusions. For this reason, we choose to build our GTR model upon a base instance segmentation-based recognition model. In addition, since the segmentation probability maps embed useful semantics and spatial context, the propose GTR model can efficiently exploit them for text reasoning.

Semantic Context Reasoning. To further enhance the text recognition performance, some methods resort to linguistic context to improve raw outputs from the VR model. For example, (Cheng et al. 2017) employ CNN to yield bags of N-grams of text string for output reasoning. Some later methods (Wang et al. 2020; Wojna et al. 2017) leverage RNN to strengthen context dependencies between characters.Recently, some methods adopt semantic context reasoning to achieve high performance. SEED (Qiao et al. 2020) proposes to use word embedding from FastText (Bojanowski et al. 2017), which relies on semantic meaning of a word instead of its visual appearance. SRN (Yu et al. 2020) uses transformer-based models where global semantic information as well as long-range word dependency is modelled by self-attention. It it computationally efficient due to the parallel nature of transformer architecture like (Xu et al. 2021), but their non-differentiable semantic reasoning block imposes a significant limitation. Based on SRN, ABINet (Fang et al. 2021) adopts the iterative correction for enhancing semantic reasoning. Beyond semantic reasoning, we propose a graph-based context reasoning model that supplements the language model to exploit both visual spatial context and linguistic context to improve the visual recognition results.

Graph-structure Data Reasoning. Considerable efforts have been made to design graph convolutional neural networks (GCN) for modelling graph-structured data (Kipf and Welling 2017; Chen et al. 2019; Wang et al. 2019). For example, in the text detection task, (Zhang et al. 2020) adopts a GCN to link characters that belong to the same word. GTC (Hu et al. 2020) utilizes a GCN to guide CTC (Graves et al. 2006) for scene text recognition. Specifically, it models the sliced visual features as the graph nodes, captures their dependency, and merges features of the same instance for prediction. PREN2D (Yan et al. 2021) adopts a meta-learning framework to extract visual representations via GCN. In this paper, we devise a two-level graph network based on GCN to perform spatial context reasoning within and between character instances to refine the visual recognition results.

Methodology

Overview

The full framework of S-GTR is shown in the Figure 2, which comprises a segmentation-based VR model, an LM, and our proposed GTR. Given the input image $\mathbf{X} \in \mathbb{R}^{H \times W \times 3}$, the segmentation-based VR first produces a segmentation map $\mathbf{M} \in \mathbb{R}^{H \times W \times C}$, where C is the number of character classes. The segmentation map $\mathbf{M}$ is decoded

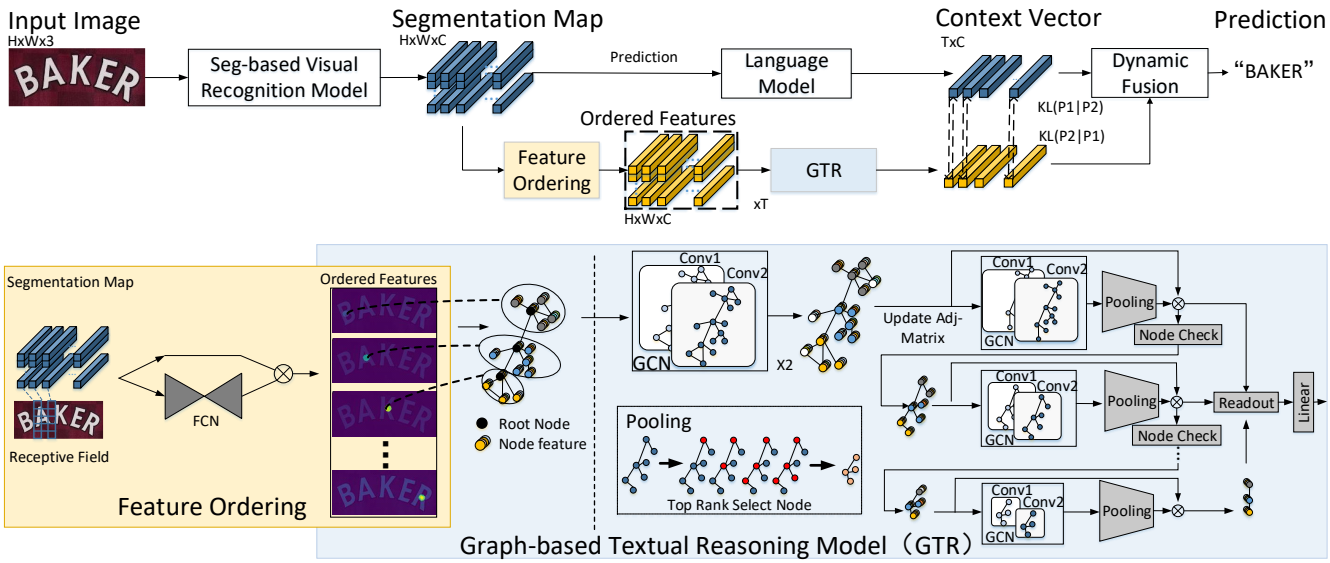

Figure 2: Overview of the proposed S-GTR model. It consists of a VR model, a LM, and the proposed GTR. GTR is stacked on the top of the VR model and in parallel with the LM. The detailed structure of GTR as well as a pre-processing step, *i.e.*, feature ordering, are also shown in the bottom part of this figure. More details can be found in Section .

to a preliminary text sequence prediction $\mathbf{T} \in \mathbb{R}^{T\times C}$ and further processed by LM for generating linguistic context vector $\mathbf{L} \in \mathbb{R}^{T\times C}$. T is the pre-defined maximum length of output sequence.

Our proposed GTR is stacked in parallel with LM, taking the segmentation map $\mathbf{M}$ as input. Firstly, we transform the map $\mathbf{M}$ with a feature ordering module to build an ordered feature tensor $\mathbf{V} \in \mathbb{R}^{T\times H\times W\times C}$, which comprises T attention maps and represents the relations between geometry features and text order information. Next, we build a sub-graph for each attention map and then connect all sub-graphs sequentially into a full graph. The graph is then deeply encoded with a GCN to produce the spatial context vector $\mathbf{S} \in \mathbb{R}^{T\times C}$. Finally, the coarse sequence prediction $\mathbf{T}$, the linguistic context $\mathbf{L}$ and the spatial context $\mathbf{S}$ are combined via dynamic fusion and the refined text is predicted.

GTR: Graph-based Textual Reasoning

Given the segmentation map $\mathbf{M}$, we employ a fully convolutional network (FCN) to obtain a series of attention maps related to the character order and use them to attend $\mathbf{M}$ via element-wise multiplication to get the ordered feature tensor $\mathbf{V} \in \mathbb{R}^{T\times H\times W\times C}$, as shown in the bottom left part of Figure 2. Based on $\mathbf{V}$, GTR firstly builds sub-graphs for all character instances and connects them sequentially. Then, the graph is encoded with a GCN and pooling operation to produce spatial context.

Graph Generation We build the two-level graph from the ordered feature tensor $\mathbf{V}$ to model the local-to-global dependency. We first connect pixels belonging to the same character in the 1-st level sub-graph. Specifically, for the i-th ordered feature map $V_i \in \mathbb{R}^{H\times W\times C}$, we first choose pixels having the same estimation to the i-th character in the text

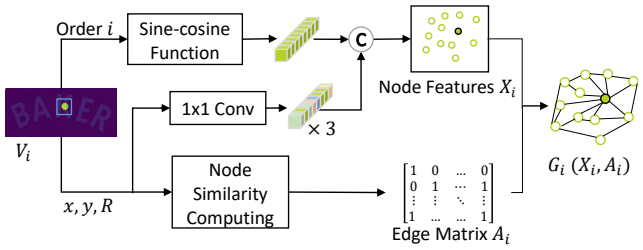

Figure 3: Illustration of the subgraph building process, including node feature generation and edge matrix compute.

sequence predicted by VR. These pixels are collected as a set $P_i = \{(x, y, R)_j\}$, where R is the C-dim feature vector in the position (x, y) of V_i and j is the pixel index. Note that we also add a root node with average x, y and R to the set. Then we construct the node feature vector $X_{i,j}$ by embedding x, y and R with three different 1×1 convolutions and i with sine and cosine functions. The four parts of embeddings are concatenated to form node features.

Next, the adjacent matrix is built according to node similarities. We compute the node similarity as a product between the position similarity E_p and the feature similarity E_f, which is defined as:

$$E_p(p, q) = 1 - \frac{D\left((x_p, y_p), (x_q, y_q)\right)}{max\left(H, W\right)}, \tag{1}$$

$$E_f(p, q) = \frac{R_p \cdot R_q}{\|R_p, \| \, \|R_q\|}, \tag{2}$$

$$E(p, q) = E_p(p, q) \cdot E_f(p, q), \tag{3}$$

where p and q are two nodes from the set P_i. The position similarity E_p is negatively proportional to the Euclidean dis-

tance between two pixels whereas the feature similarity E_f is the cosine similarity between pixel features. The product of $E_p(p,q)$ and $E_f(p,q)$ is the overall similarity E between node p and q. Then, we use the 1-hop rule (Wang et al. 2019) to build the adjacent matrix A_i. Specifically, we connect each node in V_i to other nodes that have the top-8 largest similarity and delete the connections to the nodes outside the 1-hop cluster.

After constructing sub-graphs $G_i(X_i, A_i)$, we connect them into the 2-level full graph by linking their root nodes in sequence order. The full graph is denoted as $G(X, A)$.

Spatial Context Reasoning Given a graph $G(X, A)$, we try to use the graph convolutional network to perform two-stage spatial context reasoning by following (Zhang et al. 2020; Kipf and Welling 2017).

The first stage is spatial reasoning. After obtaining the feature matrix X and the adjacency matrix A, we use a graph convolutional network to output a transformed node feature matrix Y. This process can be described as follows:

$$Y^l = \sigma([X^l; KX^l]W^l), l = 1...L, \tag{4}$$

$$K = \tilde{D}^{\frac{-1}{2}}\tilde{A}\tilde{D}^{\frac{-1}{2}}. \tag{5}$$

Here, l denotes the layer index, $L = 2$, $X(l) \in \mathbb{R}^{N\times d_i}$, $Y(l) \in \mathbb{R}^{N\times d_o}$, d_i and d_o are the dimension of input and output node features, and N is the number of nodes. $[;]$ represents concatenation. W^l is a layer-specific trainable weight matrix. σ denotes a non-linear activation function. K is an aggregation matrix of size $N \times N$, which is computed according to (Kipf and Welling 2017). Note that $X^{l+1} = Y^l$, $i.e.$, the output feature matrix Y^l is used as the input of the $l + 1$th layer.

After spatial reasoning, we perform the contextual reasoning. Denoting the output graph feature matrix from the aforementioned graph convolutional network as X_c^l, we compute a new adjacency matrix A_c based on X_c^l. Then, we calculate G according to Eq. (5) based on A_c. Next, we use a graph convolutional network to output a transformed node feature matrix Y_c^l as follows:

$$Y_c^l = \sigma((GX_c^l)W_c^l), l = 1...L, \tag{6}$$

where W_c^l is a layer-specific trainable weight matrix.

Then, we perform root node check to make sure the selected root node is the underlying reliable root node, $i.e.$, the center of the character instance. In this way, it achieves the balance between the edges with near easy nodes and distant hard nodes by satisfying the following criterion:

$$G_{iou} = \frac{G_r \cap G_s}{G_r \cup G_s} < \varepsilon, \tag{7}$$

where G_r and G_s are two subgraphs for a same character, given that s is a randomly selected node as the root node while r is always the center of character. $G_r \cap G_s$ and $G_r \cup G_s$ are the intersection and the union of 1-hop neighbors of G_r and G_s, respectively. In our experiments, ε is set to 0.75.

Next, we use a readout layer like (Lee, Lee, and Kang 2019) to aggregate node features to a fixed-size representation. The output feature of this layer is calculated as follows:

$$x_i^* = [x_i; max(x_j^*|j \in N(x_i))], \tag{8}$$

where x_j^* is the updated feature at jth node, which is also calculated according to Eq. (8), $i.e.$, x_i^* is calculated in a recursive manner. $N(x_i)$ denotes the neighboring node set of node i.

After we obtain the updated node features, we discard 50% nodes that are most distant to the root node, $i.e.$, pooling the graph into a smaller new one. We iteratively repeat the feature update and pooling process until only a node exists in the subgraph, resulting in a node sequence. Finally, the feature representations of the node sequence are passed to a linear layer for classification. We adopt the softmax cross-entropy loss for optimizing graph convolutional neural network. Similar to (Wang et al. 2019), we only back-propagate the gradient for nodes in the 1-hop neighborhood during training.

S-GTR: A Simple Baseline for STR

We incorporate our GTR to a popular segmentation-based VR model with LM, resulting in a simple baseline for STR, $i.e.$, S-GTR, as shown in Figure 2. Specifically, the VR model is designed following (Wan et al. 2020b), and the LM is based on SRN.We devise manifold training strategies to make GTR better support the the STR task.

Context Consistency Since we have two different types of reasoning features, namely linguistic context and spatial context. To prevent S-GTR from overly relying on one of them and avoid inconsistent reasoning cues to provide ambiguous results, we propose a mutual learning strategy to enforce the consistency between the two types of context features. Specifically, we compute the Kullback Leibler (KL) divergence between $\mathbf{L}$ from LM and $\mathbf{S}$ from GTR.

Dynamic Fusion Following (Yue et al. 2020) that uses a dynamic fusion module to fuse information from multiple domains, we extend it in S-GTR to combine three different text sequences from VR, LM and GTR.Formally,

$$\begin{aligned} Q_i &= Sigmoid(W_0[\mathbf{T}_i; \mathbf{L}_i; \mathbf{S}_i]), \\ Z_i &= Q_i \odot (W_1[\mathbf{T}_i; \mathbf{L}_i; \mathbf{S}_i]), \end{aligned} \tag{9}$$

where $\mathbf{T}_i$, $\mathbf{L}_i$, $\mathbf{S}_i$ are prediction vectors for the i-th character. W_0 and W_1 are two learnable linear transformations and $\odot$ indicates the element-wise multiplication operation. Z_i is the final output of S-GTR for the i-th character.

Mean Teacher-based Syn-to-Real Adaptation To mitigate the domain shift issue when using both synthetic and real datasets for training, we adopt the popular mean teacher framework (Tarvainen and Valpola 2017) in the area of domain adaptation. Specifically, a teacher network with the identical architecture as the segmentation-based VR model ($i.e.$, student network) is built and its weights are the exponential moving average of those of the student network.

Loss Function The overall loss contains three parts, including sequence prediction loss $\mathcal{L}_{\text{Seq}}$, the LM-GTR consistency loss $\mathcal{L}_{\text{CC}}$, and the mean-teacher training loss $\mathcal{L}_{\text{MT}}$:

$$L = \lambda_{\text{Seg}} * \mathcal{L}_{\text{Seg}} + \lambda_{\text{CC}} * \mathcal{L}_{\text{CC}} + \lambda_{\text{MT}} * \mathcal{L}_{\text{MT}}. \tag{10}$$

$\mathcal{L}_{Seg}$ contains a cross-entropy loss for character classification and a smooth L1 loss for order segmentation. $\mathcal{L}_{\text{CC}}$ is the

Methods	Training Data	Regular			Irregular			Params	Time
		IIIT5k	SVT	IC13	SVTP	IC15	CUTE	($\times 10^6$)	(ms)
CRNN (Shi, Bai, and Yao 2016)	ST + MJ	78.2	80.9	86.7	-	-	-	8.3	6.8
ASTER (Shi et al. 2018)	ST + MJ	93.4	89.5	91.8	78.5	76.1	79.5	22	73.1
TRBA (Baek et al. 2019)	ST + MJ	87.9	87.5	92.3	79.2	77.6	74.0	49.6	27.6
Textscanner* (Wan et al. 2020a)	ST + MJ	93.9	90.1	92.9	84.3	79.4	83.3	57	56.8
GTC (Hu et al. 2020)	ST + MJ	95.5	92.9	94.3	86.2	82.5	92.3	-	-
SCATTER (Litman et al. 2020)	ST + MJ	93.7	92.7	93.9	86.9	82.2	87.5	-	-
SEED (Qiao et al. 2020)	ST + MJ	93.8	89.6	92.8	81.4	80.0	83.6	-	-
SRN (Yu et al. 2020)	ST + MJ	94.8	91.5	95.5	85.1	82.7	87.8	49.3	26.9
RobustScanner (Yue et al. 2020)	ST + MJ	95.3	88.1	94.8	79.5	77.1	90.3	-	-
Base2D (Yan et al. 2021)	ST + MJ	95.4	93.4	95.9	86.0	81.9	89.9	59.0	61.6
PREN2D (Yan et al. 2021)	ST + MJ	95.6	94.0	96.4	87.6	83.0	91.7	-	67.4
ABINet-LV† (Fang et al. 2021)	ST + MJ	96.3	93.0	97.0	88. 5	85.0	89.2	36.7	22.0
Seg-Baseline	ST + MJ	94.2	90.8	93.6	84.3	82.0	87.6	34.0	14.0
S-GTR	ST + MJ	**95.8**	**94.1**	**96.8**	**87.9**	**84.6**	**92.3**	42.1	18.8
GTR + CRNN$^{[CTC]}$	ST + MJ	87.6	82.1	90.1	68.1	68.2	78.1	15.2	12.8
GTR + TRBA$^{[1DATT]}$	ST + MJ	93.2	90.1	94.0	80.7	76.0	82.1	54.2	32.9
GTR + SRN$^{[Transformer]}$	ST + MJ	96.0	93.1	96.1	87.9	83.9	90.7	54.3	31.6
GTR + Base2D$^{[2DATT]}$	ST + MJ	96.1	94.1	96.6	88.0	85.3	92.6	64.1	65.7
GTR + ABINet-LV$^{\dagger[Transformer]}$	ST + MJ	96.8	94.8	97.7	89.6	86.9	93.1	41.6	30.9
SAR(Li et al. 2019)	ST + MJ + R	95.0	91.2	94.0	86.4	78.0	89.6	-	-
Textscanner* (Wan et al. 2020a)	ST + MJ + R	95.7	92.7	94.9	84.8	83.5	91.6	57	56.8
RobustScanner (Yue et al. 2020)	ST + MJ + R	95.4	89.3	94.1	82.9	79.2	92.4	-	-
ABINet (Fang et al. 2021)	ST + MJ + R	97.2	95.5	97.7	90.1	86.9	94.1	-	-
S-GTR	ST + MJ + R	**97.5**	**95.8**	**97.8**	**90.6**	**87.3**	**94.7**	42.1	18.8

Table 1: Results of our S-GTR, SOTA methods and their variants with our GTR on six regular and irregular STR datasets. "R" denotes the real datasets. "*" means using character-level annotations during training. "†" means the batch size is set to 384 for a fair comparison. The superscripts in the second group of rows denote the type of different methods, $i.e.$, "CTC": CTC-based method, "1DATT": 1D attention-based method, "2DATT": 2D attention-based method, and "Transformer": Transformer-based method. Details can be found in Section .

KL loss for context consistency. $\mathcal{L}_{\text{MT}}$ is the MSE loss on the segmentation maps from teacher and student networks. λ_{Seg} and λ_{CC} are both set to 1.0. λ_{MT} is set to 1.0 when using synthetic datasets for training. After getting accurate feature representations, it is reduced to 0 gradually.

Experiments

Experimental Settings

Datasets Following (Yu et al. 2020), we use two public synthetic datasets , $i.e.$, SynthText (ST) (Gupta, Vedaldi, and Zisserman 2016) and MJSynth (MJ) (Jaderberg et al. 2014b, 2016) and a real datasets (R) (Baek, Matsui, and Aizawa 2021) for training. We test the trained model on six benchmarks including three regular scene-text datasets, $i.e.$, ICDAR2013 (Karatzas et al. 2013), IIIT5K (Mishra, Alahari, and Jawahar 2012), SVT (Wang, Babenko, and Belongie 2011), and three irregular text datasets, $i.e.$, ICDAR2015 (Karatzas et al. 2015), SVTP (Phan et al. 2013) and CUTE (Risnumawan et al. 2014). The evaluation metric is the standard word accuracy.

Implementation Details We train the model with ADAM optimizer on two synthetic datasets for 6 epochs and then transferred to the real dataset for the other 2 epochs. The total batch size is 256, equally distributed on four NVIDIA V100 GPUs. For the pre-training stage on synthetic datasets, the learning rate is set to 0.001 and divided by 10 at the 4-th and 5-th epochs. Then, we utilize the mean teacher training framework on the real dataset for the remaining 2 epochs. The detailed training setting for this stage is deferred to the supplementary material.

Our model recognize 63 types of characters, including "0-9", "a-z", and "A-Z". The max decoding length of the output sequence T is set to 25. We follow the standard image pre-processing that randomly resizing the width of original images into 4 scales, *i.e.*, 64, 128, 192 and 256, and then padding the images to the resolution of 64×256. We adopt multiple data augmentation strategies including random rotation, perspective distortion, motion blur, and adding Gaussian noise to the image.

Performance Analysis

Comparison with State-of-the-Art We compare the proposed S-GTR with state-of-the-art methods, and the results are summarized in Table 1, where the inference speed as well as the number of model parameters are also reported. As can be seen, the proposed S-GTR achieves the highest recognition accuracy and $3\times$ faster inference speed compared with the second best method PREN2D (Yan et al. 2021). In addition, when real data is utilized for training, S-GTR achieves more impressive results on all the six benchmarks, validating the effectiveness of the proposed GTR for

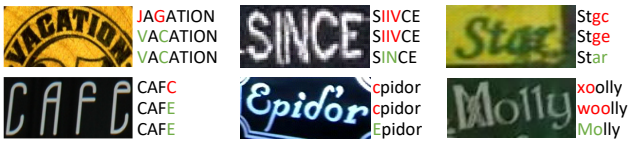

Figure 4: Results on some test images. Each image is along with three texts, which are predicted by VR, the Seg-baseline, and the proposed S-GTR model, respectively.

textual reasoning and the benefit of real data.

Plugging GTR in Different Models To further verify the effectiveness of GTR, we plug our GTR module into four representative types of STR methods, including CTC-based method ($e.g.$, CRNN (Shi, Bai, and Yao 2016)), 1D attention-based method ($e.g.$, TRBA (Baek et al. 2019)), 2D attention-based method ($e.g.$, Base2D (Yan et al. 2021)), and transformer-based methods ($e.g.$, SRN (Yu et al. 2020) and ABINet-LV (Fang et al. 2021)). For the 1D attention-based method, the prediction result of VR is a 1D semantic vector. Therefore, we adopt the 2D feature map from the layer before the prediction layer as input of GTR after feature ordering. The results are shown in the second group of rows in Table 1. As can be seen, after using GTR, the performance of all these models can be improved further. For example, the average recognition accuracy on all the available test sets is increased by 3.77%, 3.20%, 2.78%, 1.69%, and 1.65% for CRNN, TRBA, SRN, Base2D, and ABINet-LV, respectively. These results demonstrate the compatibility of our GTR to typical models.

Ablation Study

Ablation Study Results of S-GTR All the models in this ablation study have the same training configurations as used in S-GTR. To investigate the impact of different modules in S-GTR, we first train a baseline VR model, which utilizes neither LM and nor GTR. As shown in Table 2, without LM and GTR, the baseline VR model observes a significant performance drop. Compared with the baseline, a gain of 3.45% can be observed by using LM, since it introduces the global linguistic textual cues for textual reasoning and corrects some linguistically implausible predictions. The proposed GTR module exploits the visual-spatial context information to refine the output of the VR model and increases the average accuracy by 4.06%. When using both LM and GTR together, the best average performance of 90.96% can be obtained. These two modules both contribute to the improvement of S-GTR over the baseline, demonstrating that the linguistic cues and spatial context from visual semantics are complementary to each other. It is also noteworthy that the GTR module brings more gains than LM.

Note that there is no use of mutual learning in the experiments in Table 2. After comparing the results in its last row ($i.e.$, S-GTR without mutual learning) with the results of S-GTR in Table 1, which is also trained on the "ST+MJ" datasets but with mutual learning, we can find that mutual learning contributes to a better average recognition accuracy of 91.92%. It demonstrates that enforcing the consistency between the context features from LM and GTR is necessary to better exploit the complementary between these two different types of textual reasoning.

Seg-baseline VR	Seg-baseline LM	GTR	IIIT5k	SVT	IC15	CUTE
✓			91.8	86.6	77.7	84.8
✓	✓		94.2	90.8	82.0	87.6
✓		✓	94.0	91.2	82.8	88.4
✓	✓	✓	95.1	93.2	84.1	91.3

Table 2: Ablation study of the components in S-GTR.

#GCN	Adj	Pool	IIIT5k	CUTE	Pa(M)	T(ms)
2	{0,1}	Graph	95.8	92.3	42.1	18.8
1	{0,1}	Graph	94.3	90.8	39.5	16.1
3	{0,1}	Graph	96.0	92.6	44.9	22.5
2	[0,1]	Graph	95.9	92.4	42.2	20.3
2	{0,1}	AVG	94.8	89.8	38.1	15.7

Table 3: Ablation study of GTR. "#" means the number of GCN layers in the first stage of GTR. "Adj" is the value type of adjacency matrix, $i.e.$, discrete value 0 or 1 and continuous value in [0,1], respectively. "AVG" denotes employing average pooling on graph feature rather than the graph pooling described in Section .

For the qualitative analysis of different models, we present some test images and their corresponding text predictions from the basic VR model (top), Seg-baseline with LM (middle), and the proposed S-GTR (bottom) in Figure 4. We can see that LM can correct some mistaken predictions from the basic VR model by exploiting the global linguistic context. However, it is still challenging to generalize to arbitrary texts and some ambiguous cases. Compared to it, S-GTR produces satisfactory results on irregular texts in different fonts, scales, orientations, and shapes, owing to its better textual reasoning ability by exploiting both linguistic cues and spatial context from visual semantics.

Influence of Different Settings in GTR Although the proposed GTR module has shown its effectiveness in improving the STR performance on multiple benchmarks, we would also like to analyze the influence of different settings in GTR. In this section, we evaluate the performance of GTR variants with respect to different numbers of GCN layers in the first stage, different value types of adjacency matrix, and different pooling strategies. As shown in Table 3, with the increase of the number of GCN layers, the recognition accuracy, the number of parameters, and inference time all increase as well. To achieve a trade-off between recognition accuracy and model complexity, we choose 2 layers as the default setting. Besides, we find that there is almost no performance gain when using continuous values in the adjacency matrix compared to discrete values, while the inference time increases by 7.98%. Therefore, we choose the discrete value as the default value type. We further compare the graph pooling with the average pooling in the stage of contextual reasoning. The results show that graph pooling outperforms average pooling significantly since it can capture

Fusion	IIIT5k	SVT	IC13	SVTP	IC15	CUTE
Add	94.8	93.2	95.0	84.9	83.3	90.8
Concat	95.0	93.4	95.4	85.1	83.4	91.3
D-fuse	95.8	94.1	96.8	87.9	84.6	92.3

Table 4: Empirical study of the fusion strategy in S-GTR. "Add": element-wise addition. "Concat": Concatenation. "D-fuse": Dynamic fusion.

Figure 5: Visualization of feature maps from VR, GTR and S-GTR (from the second column to the last column).

the local-to-global dependency for reasoning. Therefore, we choose it as the default pooling strategy.

Impact of Fusion Strategy We also investigate the impact of fusion strategy in S-GTR when fusing the linguistic context from LM and spatial context from GTR. In addition to the proposed dynamic fusion, we consider other two choices, $i.e.$, element-wise sum and concatenation. The results are reported in Table 4. As can be seen, while the concatenation fusion strategy performs better than element-wise addition, it still falls behind the proposed dynamic fusion strategy. We suspect that the benefit may come from the learnable fusion weights which are absent in the other two non-parametric cases.

Further Visualization and Analysis

Visual Inspection Result For the qualitative analysis, we visualize the feature maps from the penultimate layer in VR, GTR and S-GTR. As shown in Figure 5, compared to the feature maps from VR, the feature maps from GTR are more strongly activated on the target characters owing to the textual reasoning of spatial context. Besides, the feature maps from S-GTR cover the target characters more precisely than GTR. These results imply that the S-GTR can learn more discriminative features by attending to the target characters and discard irrelevant information. In addition, we present the visualization of the node similarity matrix in Figure 6 for better understanding the graph generation process.

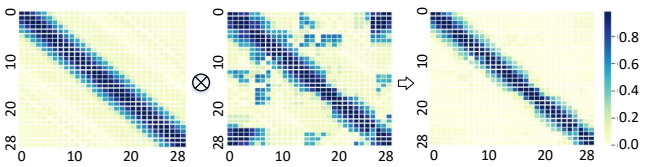

Figure 6: Visualization of a node similarity matrix, which is calculated according to Eq. (3).

Model	LM	IIIT5k	SVT	IC15	CUTE
VR	FastText	93.9	90.5	82.8	87.2
S-GTR	FastText	94.8	92.9	84.0	90.6
VR	BERT	94.3	92.0	83.8	90.8
S-GTR	BERT	95.8	94.6	85.0	92.5

Table 5: Results of S-GTR with different language models. FastText and BERT are two pretrained language models.

	CRNN	ASTER	ABINet	S-GTR
ACC	59.2%	57.4%	68.4%	72.2%
1-NED	0.68	0.69	0.79	0.82

Table 6: Results of different methods on MLT-17. "NED" is short for Normalized Edit Distance.

Compatibility of GTR to Different LMs To further investigate the compatibility of GTR to LMs, we apply GTR in a basic VR model with two different LMs, $i.e.$, FastText (Bojanowski et al. 2017) and BERT (Devlin et al. 2019). As shown in Table 5, GTR contributes to consistent gains on both FastText and BERT Language Model. In addition, we also find that using a better LM together with GTR can further improve text recognition performance.

Chinese Scene Text Recognition Like English text recognition, the Chinese scene text recognition task offers an alternative way to assess the capability of STR models. The Chinese STR task is more challenging as it requires model to handle a larger vocabulary and more complex data associations. In addition to the recognition accuracy, we also report the Normalized Edit Distance (NED) of different methods following the ICDAR-2019 ReCTS (Zhang et al. 2019). As shown in Table 6, S-GTR outperforms other methods significantly on the multi-language dataset MLT-2017 (Nayef et al. 2017). It demonstrates that GTR is still very effective for textual reasoning of Chinese text materials.

Conclusion

In this paper, we propose the idea of performing textual reasoning based on visual semantics from a basic visual recognition (VR) model for the Scene Text Recognition (STR) task. We implement it as a graph-based textual reasoning (GTR) module, which can act as an easy-to-plug-in module in existing representative methods. It is shown to be very effective in improving STR performance while being complementary to the common practice, $i.e.$, language model-based linguistic context reasoning. Experimental results on six challenging STR benchmarks demonstrate that GTR can be plugged in different types of state-of-the-art STR models and improve their recognition performance further. GTR also shows good compatibility with different language models. Based on a simple segmentation-based VR model, a simple S-GTR baseline sets state-of-the-art on both English and Chinese text materials. We hope this work can provide a new perceptive to study textual reasoning in the STR task and inspire more follow-up work in the future, such as efficient design for spatial context-based reasoning as well as the way of effective fusion of different reasoning results.

Acknowledgements

This work was supported in part by National Natural Science Foundation of China under Grants (No.62076186, No.61822113), and in part by Science and Technology Major Project of Hubei Province (Next-Generation AI Technologies) under Grant (No.2019AEA170). The numerical calculations in this paper have been done on the supercomputing system in the Supercomputing Center of Wuhan University.

References

Baek, J.; Kim, G.; Lee, J.; Park, S.; Han, D.; Yun, S.; Oh, S. J.; and Lee, H. 2019. What is wrong with scene text recognition model comparisons? dataset and model analysis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4715–4723.

Baek, J.; Matsui, Y.; and Aizawa, K. 2021. What If We Only Use Real Datasets for Scene Text Recognition? Toward Scene Text Recognition With Fewer Labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3113–3122.

Bojanowski, P.; Grave, E.; Joulin, A.; and Mikolov, T. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5: 135–146.

Chen, Y.; Rohrbach, M.; Yan, Z.; Shuicheng, Y.; Feng, J.; and Kalantidis, Y. 2019. Graph-based global reasoning networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 433–442.

Cheng, Z.; Bai, F.; Xu, Y.; Zheng, G.; Pu, S.; and Zhou, S. 2017. Focusing attention: Towards accurate text recognition in natural images. In *Proceedings of the IEEE International Conference on Computer Vision*, 5076–5084. Venice, Italy: IEEE.

Cheng, Z.; Xu, Y.; Bai, F.; Niu, Y.; Pu, S.; and Zhou, S. 2018. Aon: Towards arbitrarily-oriented text recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5571–5579. Salt Lake City, UT, USA: IEEE.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT*.

Fang, S.; Xie, H.; Wang, Y.; Mao, Z.; and Zhang, Y. 2021. Read Like Humans: Autonomous, Bidirectional and Iterative Language Modeling for Scene Text Recognition. *arXiv preprint arXiv:2103.06495*.

Gao, Y.; Chen, Y.; Wang, J.; Lei, Z.; Zhang, X.; and Lu, H. 2018. Recurrent Calibration Network for Irregular Text Recognition. *CoRR*, abs/1812.07145: arXiv–1812.

Graves, A.; Fernández, S.; Gomez, F.; and Schmidhuber, J. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the International Conference on Machine Learning*, ICML '06, 369–376. New York, NY, USA: Association for Computing Machinery.

Gupta, A.; Vedaldi, A.; and Zisserman, A. 2016. Synthetic data for text localisation in natural images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2315–2324. Las Vegas, NV, USA: IEEE.

Hu, W.; Cai, X.; Hou, J.; Yi, S.; and Lin, Z. 2020. Gtc: Guided training of ctc towards efficient and accurate scene text recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 07, 11005–11012. New York, NY, USA: AAAI.

Jaderberg, M.; Simonyan, K.; Vedaldi, A.; and Zisserman, A. 2014a. Deep structured output learning for unconstrained text recognition. *arXiv preprint arXiv:1412.5903*.

Jaderberg, M.; Simonyan, K.; Vedaldi, A.; and Zisserman, A. 2014b. Synthetic Data and Artificial Neural Networks for Natural Scene Text Recognition. *ArXiv*, abs/1406.2227: 1–10.

Jaderberg, M.; Simonyan, K.; Vedaldi, A.; and Zisserman, A. 2016. Reading text in the wild with convolutional neural networks. *International Journal of Computer Vision*, 116(1): 1–20.

Jaderberg, M.; Simonyan, K.; Zisserman, A.; and Kavukcuoglu, K. 2015. Spatial transformer networks. In *Proceedings of the Annual Conference on Neural Information Processing Systems*, 2017–2025. Montreal, Quebec, Canada: MIT Press.

Karatzas, D.; Gomez-Bigorda, L.; Nicolaou, A.; Ghosh, S.; Bagdanov, A.; Iwamura, M.; Matas, J.; Neumann, L.; Chandrasekhar, V. R.; Lu, S.; et al. 2015. ICDAR 2015 competition on robust reading. In *Proceedings of the International Conference on Document Analysis and Recognition*, 1156–1160. Nancy, France: IEEE.

Karatzas, D.; Shafait, F.; Uchida, S.; Iwamura, M.; i Bigorda, L. G.; Mestre, S. R.; Mas, J.; Mota, D. F.; Almazan, J. A.; and De Las Heras, L. P. 2013. ICDAR 2013 robust reading competition. In *Proceedings of the International Conference on Document Analysis and Recognition*, 1484–1493. Washington, DC, USA: IEEE.

Kipf, T.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. *ArXiv*, abs/1609.02907: 1–14.

Lee, J.; Lee, I.; and Kang, J. 2019. Self-Attention Graph Pooling. *ArXiv*, abs/1904.08082.

Li, H.; Wang, P.; Shen, C.; and Zhang, G. 2019. Show, attend and read: A simple and strong baseline for irregular text recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 8610–8617. Honolulu, Hawaii,USA: AAAI.

Liao, M.; Zhang, J.; Wan, Z.; Xie, F.; Liang, J.; Lyu, P.; Yao, C.; and Bai, X. 2019. Scene text recognition from two-dimensional perspective. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 8714–8721. Honolulu, Hawaii,USA: AAAI.

Litman, R.; Anschel, O.; Tsiper, S.; Litman, R.; Mazor, S.; and Manmatha, R. 2020. Scatter: selective context attentional scene text recognizer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11962–11972.

Mishra, A.; Alahari, K.; and Jawahar, C. 2012. Scene text recognition using higher order language priors. In *British Machine Vision Conference*, 1–11. Surrey, UK: BMVA.

Nayef, N.; Yin, F.; Bizid, I.; Choi, H.; Feng, Y.; Karatzas, D.; Luo, Z.; Pal, U.; Rigaud, C.; Chazalon, J.; et al. 2017. Icdar2017 robust reading challenge on multi-lingual scene text detection and script identification-rrc-mlt. In *Proceedings of the International Conference on Document Analysis and Recognition*, volume 1, 1454–1459. IEEE.

Phan, T. Q.; Shivakumara, P.; Tian, S.; and Tan, C. L. 2013. Recognizing text with perspective distortion in natural scenes. In *Proceedings of the IEEE International Conference on Computer Vision*, 569–576. Sydney,Australia: IEEE.

Qiao, Z.; Zhou, Y.; Yang, D.; Zhou, Y.; and Wang, W. 2020. Seed: Semantics enhanced encoder-decoder framework for scene text recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 13528–13537.

Risnumawan, A.; Shivakumara, P.; Chan, C. S.; and Tan, C. L. 2014. A robust arbitrary text detection system for natural scene images. *Expert Systems with Applications*, 41(18): 8027–8048.

Shi, B.; Bai, X.; and Yao, C. 2016. 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.

Shi, B.; Yang, M.; Wang, X.; Lyu, P.; Yao, C.; and Bai, X. 2018. Aster: An attentional scene text recognizer with flexible rectification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(9): 2035–2048.

Tarvainen, A.; and Valpola, H. 2017. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30.

Wan, Z.; He, M.; Chen, H.; Bai, X.; and Yao, C. 2020a. Textscanner: Reading characters in order for robust scene text recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 12120–12127. New York, NY, USA: AAAI.

Wan, Z.; Zhang, J.; Zhang, L.; Luo, J.; and Yao, C. 2020b. On vocabulary reliance in scene text recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11425–11434.

Wang, K.; Babenko, B.; and Belongie, S. 2011. End-to-end scene text recognition. In *Proceedings of the International Conference on Computer Vision*, 1457–1464. Barcelona,Spain: IEEE.

Wang, T.; Zhu, Y.; Jin, L.; Luo, C.; Chen, X.; Wu, Y.; Wang, Q.; and Cai, M. 2020. Decoupled Attention Network for Text Recognition. *ArXiv*, abs/1912.10205.

Wang, Z.; Zheng, L.; Li, Y.; and Wang, S. 2019. Linkage based face clustering via graph convolution network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1117–1125. Long Beach, CA, USA: IEEE.

Wojna, Z.; Gorban, A. N.; Lee, D.-S.; Murphy, K.; Yu, Q.; Li, Y.; and Ibarz, J. 2017. Attention-based extraction of structured information from street view imagery. In *Proceedings of the International Conference on Document Analysis and Recognition*, volume 1, 844–850. IEEE.

Xu, Y.; Zhang, Q.; Zhang, J.; and Tao, D. 2021. ViTAE: Vision Transformer Advanced by Exploring Intrinsic Inductive Bias. In *Thirty-Fifth Conference on Neural Information Processing Systems*.

Yan, R.; Peng, L.; Xiao, S.; and Yao, G. 2021. Primitive Representation Learning for Scene Text Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 284–293.

Yang, X.; He, D.; Zhou, Z.; Kifer, D.; and Giles, C. L. 2017. Learning to Read Irregular Text with Attention Mechanisms. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 3. Melbourne, Australia: ijcai.org.

Ye, J.; Zhang, J.; Liu, J.; Du, B.; and Tao, D. 2021. I3CL: Intra-and Inter-Instance Collaborative Learning for Arbitrary-shaped Scene Text Detection. *arXiv preprint arXiv:2108.01343*.

Yu, D.; Li, X.; Zhang, C.; Liu, T.; Han, J.; Liu, J.; and Ding, E. 2020. Towards accurate scene text recognition with semantic reasoning networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 12113–12122. Seattle, WA, USA: IEEE.

Yue, X.; Kuang, Z.; Lin, C.; Sun, H.; and Zhang, W. 2020. RobustScanner: Dynamically Enhancing Positional Clues for Robust Text Recognition. In *Proceedings of the European Conference on Computer Vision*, 135–151. Glasgow,UK: Springer.

Zhang, J.; and Tao, D. 2020. Empowering things with intelligence: a survey of the progress, challenges, and opportunities in artificial intelligence of things. *IEEE Internet of Things Journal*, 8(10): 7789–7817.

Zhang, R.; Zhou, Y.; Jiang, Q.; Song, Q.; Li, N.; Zhou, K.; Wang, L.; Wang, D.; Liao, M.; Yang, M.; et al. 2019. Icdar 2019 robust reading challenge on reading chinese text on signboard. In *2019 international conference on document analysis and recognition (ICDAR)*, 1577–1581. IEEE.

Zhang, S.-X.; Zhu, X.; Hou, J.-B.; Liu, C.; Yang, C.; Wang, H.; and Yin, X.-C. 2020. Deep relational reasoning graph network for arbitrary shape text detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 9699–9708. Seattle, WA, USA: IEEE.