Deep Relational Reasoning Graph Network for Arbitrary Shape Text Detection

Shi-Xue Zhang[1], Xiaobin Zhu[1], Jie-Bo Hou[1], Chang Liu[1]
Chun Yang[1], Hongfa Wang[4], Xu-Cheng Yin[1,2,3*]

[1]School of Computer and Communication Engineering, University of Science and Technology Beijing
[2]Institute of Artificial Intelligence, University of Science and Technology Beijing
[3]USTB-EEasyTech Joint Lab of Artificial Intelligence, [4]Tencent Technology (Shenzhen) Co. Ltd
zhangshixue111@163.com, {zhuxiaobin, chunyang, xuchengyin}@ustb.edu.cn
houjiebo@gmail.com, lasercat@gmx.us, hongfawang@tencent.com

Abstract

*Arbitrary shape text detection is a challenging task due to the high variety and complexity of scenes texts. In this paper, we propose a novel unified relational reasoning graph network for arbitrary shape text detection. In our method, an innovative local graph bridges a text proposal model via Convolutional Neural Network (CNN) and a deep relational reasoning network via Graph Convolutional Network (GCN), making our network end-to-end trainable. To be concrete, every text instance will be divided into a series of small rectangular components, and the geometry attributes (*e.g.*, height, width, and orientation) of the small components will be estimated by our text proposal model. Given the geometry attributes, the local graph construction model can roughly establish linkages between different text components. For further reasoning and deducing the likelihood of linkages between the component and its neighbors, we adopt a graph-based network to perform deep relational reasoning on local graphs. Experiments on public available datasets demonstrate the state-of-the-art performance of our method. Code is available at* `https://github.com/GXYM/DRRG`.

1. Introduction

Scene text detection has been widely applied in various applications, such as online education, product search, instant translation, and video scene parsing [39, 26]. With the prosperity of deep learning, text detection algorithms [27, 42, 21, 19] have achieved impressive performance in controlled environments where text instances have regular shapes or aspect ratios. However, because of the limited text representation forms, pioneer works tend to fail in detecting texts with arbitrary shapes. Recently, some methods, *e.g.*, TextSnake [17] and CRAFT [1], try to solve this problem with the Connected Component (CC) strategy. However, these methods haven't fully explored the abundant relations between text components, which can benefit the aggregation of text components for final text instance.

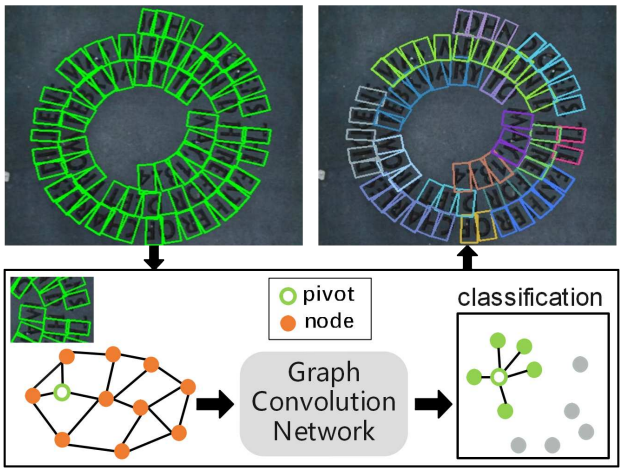

Figure 1. Illustration of relational reasoning: Generating local graphs based on the geometry attributes of text components; inferring linkage likelihood via GCN; finally grouping node classification results into text.

In the CC-based method, one essential task is to excavate the rational relations between separated character/component regions for linking them into holistic text instances. The existing methods usually use pre-defined rule, link map or embedding map to group the detected components into text instance. Generally speaking, grouping the text components with learned link relationship or embedding relationship is more robust than using pre-defined rules, especially in the cases of long and curve texts. From our key observations and experiments, deep relational reasoning for mining the stable relations between these component regions can greatly enhance the performance of ar-

*Corresponding author.

bitrary shape text detection. The link or embedding based methods [21, 28] usually uses CNNs to deduce the linkage of separate components, but the separated components are actually non-Euclidean data and CNNs are not good at processing non-Euclidean data. Therefore, the simple link map or embedding map is inadequate for learning stable relations between two non-adjacent components. The non-Euclidean data can be represented with graph, so we can transform the separate text components into graphs. As shown in Fig. 1, we regard one text component as a node. Hence, we can select a node as a pivot and connect it with surrounding nodes into a local graph, as described in Sec. 3.3. The context information contained in local graphs (edges among the nodes) is informative for estimating the linkage likelihood between pivot and other nodes. It's a consensus that graph network has innate advantage for deducing relationships between nodes on the graph. Recently, the GCN based methods have achieved remarkable performance in clustering face [33] and global reasoning for various tasks [2]. Highly motivated by the works in [33, 2], we apply a graph convolution network to perform deep reasoning on local graphs to deduce deep linkage likelihood between components and corresponding neighbors for arbitrary shape text detection.

In this paper, we propose a novel unified deep relational reasoning graph network for arbitrary shape text detection. According to CTPN [27] and TextSnake [17], we divide every text instance into text components, and propose a text proposal network to estimate the geometry attributes of these text components. To group the generated components, we adopt a graph-based network to perform deep relational reasoning and inferring the linkage relationship using the geometry attributes of components and neighbors. In addition, a local graph is designed to bridge the text proposal network and relational reasoning network, making our network end-to-end trainable. Finally, we group the detected text components into holistic text instances according to the relational results.

In summary, the main contributions of this paper are three-fold:

- We propose a novel unified end-to-end trainable framework for arbitrary shape text detection, in which a novel local graph bridges a CNN based text proposal network and a GCN based relational reasoning network.
- To the best of our knowledge, our work presents one of the very first attempts to perform deep relational reasoning via graph convolutional network for arbitrary shape text detection.
- The proposed method achieves the state-of-the-art performance both on polygon datasets and quadrilateral datasets.

2. Related Work

Regression-Based Methods. Methods of this type rely on a box-regression based object detection frameworks with word-level and line-level prior knowledge [19, 10, 11, 42]. Different with generic objects, texts are often presented in irregular shapes with various aspect ratios. To deal with this problem, RRD [11] adjusts anchor ratios of SSD [13] for accommodating the aspect ratio variations in irregular shapes. Textboxes++ [10] modifies convolutional kernels and anchor boxes to effectively capture various text shapes. EAST [42] directly inferences pixel-level quadrangles of word candidates without anchor mechanism and proposal detection. Although regression-based methods have achieved good performance in quadrilateral text detection, they often can't well adapt to arbitrary shape text detection.

Segmentation-Based Methods. Methods of this type [3, 30, 28, 34, 17] mainly draw inspiration from semantic segmentation methods and detect texts by estimating word bounding areas. In PixelLink [3], linkage relationships between a pixel and its neighboring pixels are predicted for grouping pixels belonging to same instance. To effectively distinguish adjacent text instances, PSENet [30] adopts a progressive scale algorithm to gradually expand the predefined kernels. Tian *et al*. [28] considered each text instance as a cluster and perform pixel clustering through an embedding map. TextField [34] adopts a deep direction field to link neighbor pixels and generate candidate text parts. However, the performances of these methods are strongly affected by the quality of segmentation accuracy.

CC-Based Methods. The CC-based methods usually detect individual text parts or characters firstly, followed by a link or group post-processing procedure for generating final texts. CC-based methods [24, 38, 41, 37] had been widely used in traditional scene text detection methods before the popularity of deep learning. In the era of deep learning, CC-based methods have also been extensively studied [27, 21, 25, 1, 4]. CTPN [27] uses a modified framework of Faster R-CNN [20] to extract horizontal text components with a fixed-size width for easily connecting dense text components and generating horizontal text lines. SegLink [21] decomposes every scene text into two detectable elements, namely segment and link, where the link indicates that a pair of adjacent segments belong to the same word. CRAFT [1] detects the text area by exploring each character and affinity between characters. TextDragon [4] first detects the local area of the text, and then groups these bounding boxes according to their geometric relations.

Relational Reasoning. CC-based methods are usually robust for long or non-quadrilateral text, but the performance of these methods are strongly depends on the robustness of grouping or linkage results. Text pixels are clustered by learning the linkage relationship between a pixel and its neighboring pixels in [3] . In [28], embedding features are

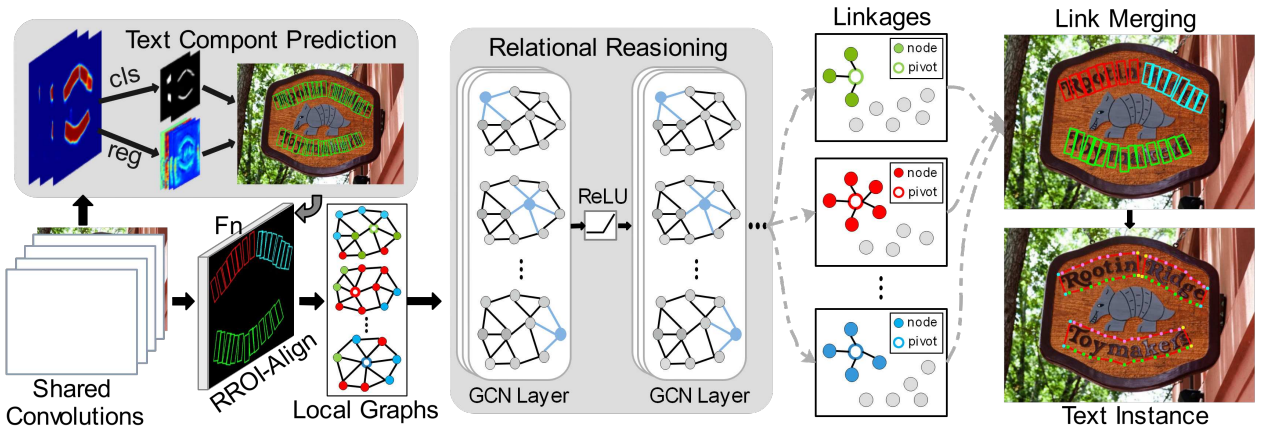

Figure 2. Overview of our overall architecture. Our network mainly consists of five components, *i.e.*, shared convolutions, text component prediction, local graphs, relational reasioning, and link merging.

used to provide instance information and to generate the text area. CRAFT [1] predicts character region maps and affinity maps by weakly-supervised learning. The region map is used to localize characters, and the affinity map is used to group characters into a single instance. These methods are based on the CNNs, which cannot directly capture relations between distant component regions for the limitation of local convolutional operators. Recently, Wang *et al*. [33] proposed a spectral-based GCN to solve the problem of clustering faces, where the designed GCN can rationally link different face instances belonging to the same person in complex situations.

3. Proposed Method

3.1. Overview

The framework of our method is illustrated in Fig. 2. The text component proposal network and the deep relational reasoning graph network share convolutional features, and the shared convolutions use the VGG-16 [23] with FPN [12] as backbone, as shown in Fig. 3. The text proposal network uses the shared features to estimate geometric attributes of text components. After obtaining the geometry attributes, the local graph can roughly establish linkages between different text components. Based on local graphs, the relational reasoning network will further infer the deep likelihood of linkages between the component and its neighbors. Finally, text components will be aggregated into holistic text instance according to the reasoning results.

3.2. Text Component Prediction

In our work, each text instance is constructed by a series of ordered rectangular components, as shown in Fig. 4 (a). And each text component D is associated with a group of geometry attributes, *i.e.*, $D = (x, y, h, w, \cos\theta, \sin\theta)$, in which x and y are the axis of text box; h and w are the height and the width of the component; $\cos\theta$ and $\sin\theta$ indicate the

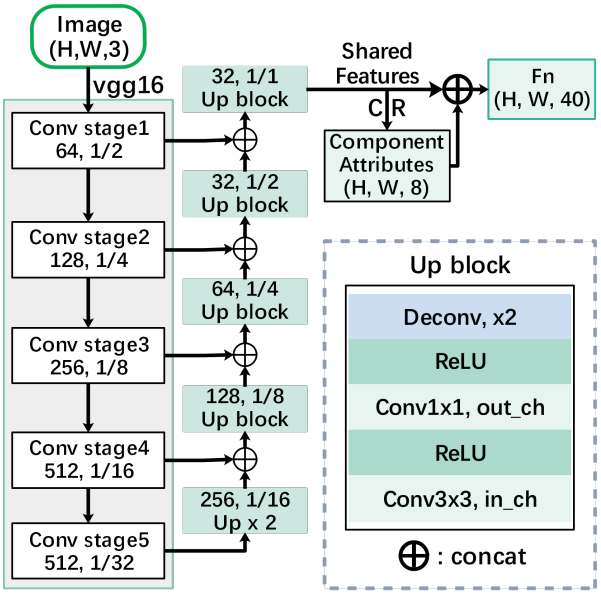

Figure 3. Architecture of shared convolutions, where CR represents classification and regression operation in text component prediction, and details are listed in Eq. 3.

orientation of the text component. The h is the sum of $h1$ and $h2$, as shown in Fig. 4 (c). The w is obtained by a linear transformation on h, which is computed as

$$w_i = \begin{cases} w_{min}, & h_i <= 2 \cdot w_{min}, \\ h_i/2, & 2 \cdot w_{min} < h_i < 2 \cdot w_{max}, \\ w_{max}, & h_i >= 2 \cdot w_{max}, \end{cases} \tag{1}$$

where h_i denotes the height of the i-th text component. In experiments, we empirically set $w_{min} = 8$ and $w_{max} = 24$.

In order to define the orientation of text components and extract the text center region (TCR) easily, we use the method in [17] to calculate the head and tail of text region, as shown with the black arrow in Fig. 4 (a). Text region is divided into a series of ordered quadrilateral regions along the long side (indicated by yellow lines

as shown in Fig. 4 (a)), so we can obtain two groups of points $P1 = \{tp_0, tp_1, ..., tp_i, ..., tp_n\}$ and $P2 = \{bp_0, bp_1, ..., bp_i, ..., bp_n\}$. The line marked with red points is the top line and green points is the bottom. In our approach, we need to clearly define the top and bottom of each text instance, according to the following criterion:

$$p = \sum_{i=0}^{n} sin(v_i), v_i \in V, \tag{2}$$

where V ($V = \{tp_0 - bp_0, ..., tp_i - bp_i, ..., tp_n - bp_n\}$) is a group of vertexes (tp_i is the center of the top line and bp_i is the center of the bottom line). If $p >= 0$, $P1$ is top and $P2$ is bottom, else $P1$ is bottom and $P2$ is top. The angle of vector v_i indicates the orientation θ of text component.

TCR is obtained by shrinking text region (TR), as shown in Fig. 4 (b). First, we compute the text center line, Then, we shrink the two ends of center line by $0.5w$ end pixels, making it easy for the network to separate adjacent text instances and reduce the computation cost of NMS. Finally, we expand the center line area by $0.3h$. After extracting shared features, two convolution layers are applied to predict the attributes of the text component as

$$CR = conv_{1\times1}(conv_{3\times3}(F_{share})), \tag{3}$$

where $CR \in \Re^{h\times w\times 8}$, with 4 channels for the classification logits of TR/TCR, and 4 channels for the regression logits of $h_1, h_2, \cos\theta$, and $\sin\theta$. The final predictions are obtained by softmaxing TR/TCR and regularizing $\cos\theta$ and $\sin\theta$ for squaring sum equals 1 [17]. Final detection results are produced by threshold and locality-aware NMS on the positive samples.

Detection Loss. The text component prediction loss is consisted of two losses, and computed as

$$L_{det} = L_{cls} + L_{reg}, \tag{4}$$

where L_{reg} is a smooth $L1$ [20] regression loss and L_{cls} is a cross-entropy classification loss. The classification loss is computed as

$$L_{cls} = L_{tr} + \lambda_1 L_{tcrp} + \lambda_2 L_{tcrn}, \tag{5}$$

where L_{tr} represents the loss for TR; L_{tcrp} only calculates pixels inside TR and L_{tcrn} just calculates the pixels outside TR. L_{tcrn} is used to suppress noise of the background in TCR. In this way, the obtained TCR can benefit post-processing steps. The OHEM [22] is adopted for TR loss, in which the ratio between the negatives and positives is set to 3:1. In our experiments, the weights λ_1 and λ_2 are empirically set to 1.0 and 0.5, respectively.

Because the attributes of height and orientation are absent for non-TCR region, we only calculate regression loss

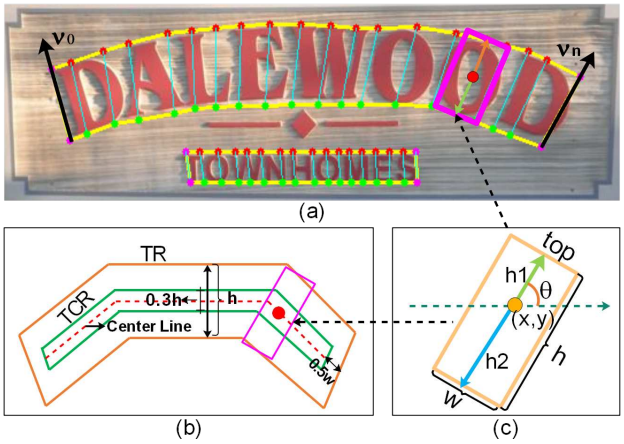

Figure 4. Illustration of the proposal of text component: (a) Generating text component; (b) Extracting text center region; (c) Calculating geometry attributes.

for TCR region as followings:

$$L_{reg} = L_h + \beta(L_{sin} + L_{cos}), \tag{6}$$

$$L_{sin} = smooth_{L1}(\hat{\sin}\,\theta - \sin\theta), \tag{7}$$

$$L_{cos} = smooth_{L1}(\hat{\cos}\,\theta - \cos\theta), \tag{8}$$

$$L_h = \frac{1}{\Omega}\sum_{i\in\Omega}(\log(h+1)\sum_{k=0}^{2} smooth_{L1}(\frac{\hat{h_{ki}}}{h_{ki}} - 1)), \tag{9}$$

where h_{ki}, $\sin\theta$ and $\cos\theta$ are ground-truth values, and $\hat{h_{ki}}$, $\hat{\sin}\,\theta$ and $\hat{\cos}\,\theta$ are the corresponding predicted values; the Ω denotes the set of positive elements in TCR; the h is the height of text component in ground truth. The weight $\log(h+1)$ is beneficial for the height regression of large scale text component. The hyper-parameter β is set to 1.0 in our work.

3.3. Local Graph Generation

We estimate the linkage likelihood between two nodes (text components) based on their context information in a local graph. It is inefficient to construct a whole graph for each image because text component usually only has the possibility of connection with its neighbors. Therefore, we construct multiple local graphs for every image. These local graphs generally contain a limited number of nodes, which will make it easy making relational reasoning high efficient.

We modify IPS [33] to generate local graph, where pivot's neighbors up to h-hop are used as nodes. In our work, we just use 2-hop as nodes for local graph. For clear explanation, V_p is used to represent the nodes in local graph G_p and p represents the pivot. The 1-hop neighbors of p consist of 8 nearest neighbors, and the 2-hop neighbors of p consist of 4 nearest neighbors. The high-order neighbors provide auxiliary information of the local structure of the context between a pivot and its neighbor [33]. Here, we

only consider the Euclidean similarity E_s between nodes for performing KNN operation, and E_s computed as

$$E_s = 1 - D(p, v_i)/max(H_m, W_m), v_i \in V_p, \quad (10)$$

where $D(p, v_i)$ is an L2 distance between p and v_i, H_m is image height, and W_m is image width. To avoid gradient accumulation of easy samples caused by many identical graphs in training, the pivot p should satisfy the following criterion:

$$G_{iou} = \frac{G_p \cap G_q}{G_p \cup G_q} < \xi, \ p, q \in T, \quad (11)$$

where G_p and G_q are two local graphs; the pivot p and q are in the same text instance T; $G_p \cap G_q$ is the intersection of 1-hop neighbors of G_p and G_q; $G_p \cup G_q$ is the union of 1-hop neighbors of G_p and G_q. In our experiments, ξ is set to 0.75 . This strategy not only leads to considerable acceleration, but also reduces the number of easy samples, yet keep the balance of hard and easy samples.

3.4. Deep Relational Reasoning

The text components in every image will be divided into multiple local graphs by local graph generation, which consists of the pivot and its 2-hop neighbors. The rough linkage information contained in the local graph (edges among the nodes) is valuable for estimating the linkage likelihood between the pivot and its neighbors. For further reasoning and deducing the likelihood of linkage between the pivot and its neighbors, We adopt a specific graph-based neural network [33, 8] to excavate the linkage relationships between the pivot and its neighbors based on local graph. The graph is usually expressed as $g(X, A)$, and the graph convolutional network usually takes the feature matrix X and the adjacency matrix A as the input of the network. Therefore, we need to extract the feature matrix X and compute matrix A for the local graph.

Node Feature Extraction. The node features consist of two parts features, namely, RROI features and geometric features. In order to obtain the RROI features, we use the RRoI-Align layer which integrates the advantages of RoI-Align [6] and RRoI [19] to extract the feature block of the input text component. To ensure the convergence ability of our model, we use the ground truth to generate the text component in training. Text components within the same text instance have similar geometric features. However, the RROI features will lose some geometric attributes, such as location information. Therefore, we should take these geometric attributes into consideration during node feature generation, as shown in Fig. 5. For one text component, we feed it with the feature maps F_n to RRoI-Align layer, and then a $1 \times 3 \times 4 \times C_r$ feature block is obtained, where F_n is illustrated in Fig. 3. Afterwards, it will be reshaped to $1 \times 12 \cdot C_r$, namely F_r. The geometric attributes of text

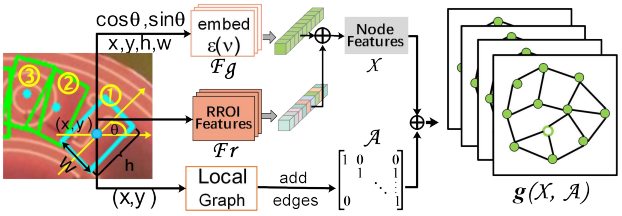

Figure 5. Illustration of $g(X, A)$ generation. F_r represents the geometry features; X is node feature matrix; A is an adjacency matrix; $g(X, A)$ represents the mathematical expression of local graph. The details of embedding operation in Eq. 12 and Eq. 13.

component are embedded into high dimensional spaces according to the technique in [29, 5]. The embedding is performed by applying *sine* and *cosine* functions of varying wavelengths to a scalar z as

$$\varepsilon_{2i}(z) = \cos(\frac{z}{1000^{2i/C_\varepsilon}}), i \in (0, C_\varepsilon/2 - 1), \quad (12)$$

$$\varepsilon_{2i+1}(z) = \sin(\frac{z}{1000^{2i/C_\varepsilon}}), i \in (0, C_\varepsilon/2 - 1), \quad (13)$$

where the dimension of the embedding vector $\varepsilon(z)$ is C_ε. As a result, each text component is embedded into a vector F_g with $6 \cdot C_\varepsilon$ dimension. Finally, F_r and F_g will be concatenated together as node features.

Node Feature Normalization. We normalize the features of node by subtracting x_p. It encodes the pivot p information into the features of a local graph and makes the relation reasoning network easily learn the linkage relationships between the pivot and its neighbors.

$$\mathbf{F}_p = [..., x_q - x_p, ...]^T, q \in V_p, \quad (14)$$

where x_p is the feature of the pivot p; the V_p denotes the node set on local graph and their features are $\{x_q | q \in V_p\}$.

Adjacency Matrix Generation. We use an adjacency matrix $A_p \in \Re^{N*N}$ to represent the topological structure of local graph. For a node $n_i \in V_p$, we filter out the top u nearest neighbors $U(n_i)$. For the node $n_j \in U(n_i)$, we will set $A_p(n_i, n_j) = 1$. The hyper-parameter u is empirically set to 3 in our work.

Graph Convolutions. After obtaining the feature matrix X and the adjacency matrix A, we use a graph-based relational reasoning network to estimate the linkage relationships of the pivot and its neighbors based on the established graph. We modify the structure in [33, 8], and the graph convolution layer in our method can be formulated as

$$\mathbf{Y}^{(l)} = \sigma((\mathbf{X}^{(l)} \oplus \mathbf{G}\mathbf{X}^{(l)})\mathbf{W}^l), \quad (15)$$

$$\mathbf{G} = \tilde{\mathbf{D}}^{-1/2}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-1/2}, \quad (16)$$

where $\mathbf{X}^{(l)} \in \Re^{N \times d_i}$, $\mathbf{Y}^{(l)} \in \Re^{N \times d_o}$, d_i/d_o is the dimension of input / output node features and N is the number of nodes; $\mathbf{G}$ is a symmetric normalized laplacian of size

$N \times N$; the operator $\oplus$ represents matrix concatenation; $W^{(l)}$ is a layer-specific trainable weight matrix; $\sigma(\cdot)$ denotes a non-linear activation function; $\tilde{A} = A + I_N$ is an adjacency matrix of the local graph with added self-connections; I_N is the identity matrix and $\tilde{\mathbf{D}}$ is a diagonal matrix with $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$. Our relational reasoning model is the stack of one Batch Normalization layer and four graph convolution layers activated by the ReLU function. We adopt softmax cross-entropy loss as the objective function for optimization. Similar to [33], we only back-propagate the gradient for nodes on the 1-hop neighbors in training, because we just care about the linkage between a pivot and its 1-hop neighbors. For testing, we also only consider the classification on 1-hop nodes.

3.5. Inference

Given the text components, we group text components into text instances according to the reasoning results. We first apply thresholding to TR and TCR respectively, and then NMS is applied to reduce redundancy. To infer the likelihood of linkages between the pivot and its neighbors, we loop over all text components, constructing a local graph with each component as the pivot. Consequently, we obtain a set of edges weighted by the linkage likelihood. Finally, we use Breath First Search (BFS) to cluster and merge the linkages.

After we get the clustered text components, we sort the components for boundary generation. The text instance T can be represented as $T = \{D_0, ...D_i, ..., D_n\}$. The Min-Path algorithm is applied to search the shortest path through all text component centers, and then we sort T by searching results. For boundary generation, we just need to link the mid-point of the ordered top and bottom in ordered text components sequentially, as shown in Fig. 2.

4. Experiments

4.1. Datasets

Total-Text: It consists of $1,255$ training and 300 testing complex images, including horizontal, multi-oriented, and curved text instances with polygon and word-level annotations.

CTW-1500: It consists of $1,000$ training and 500 testing images. Every image has curved text instances, which are all annotated by polygons with 14 vertices.

MSRA-TD500: It consists of 500 training and 200 testing images, including English and Chinese scripts. This dataset is dedicated for detecting multi-lingual long texts of arbitrary orientations.

ICDAR2015: It consists of $1,000$ training images and 500 testing images, including many multi-orientated and very small-scale text instances. The ground truth is annotated with word-level quadrangle.

ICDAR2017: It consists of $7,200$ training images, $1,800$ validation images and $9,000$ test images with texts in 9 languages for multi-lingual scene text detection. The text instances are also annotated by quadrangle.

4.2. Implementation Details

The backbone of our network is the pre-trained VGG16 [23] on ImageNet [9]. The training procedure mainly includes two steps: pre-training our network on SynthText dataset with two epochs, and fine-tuning on specific benchmark dataset with 600 epochs. In the pre-training stage, we randomly crop text regions, which will be resized to 512. The batch size is set to 12. Adam optimizer is applied to train our model with a learning rate 10^{-4}. In fine-tuning, for multi-scale training, we randomly crop the text region, and resize them to 640×640 (batch is 8), 800×800 (batch is 4), and 960×960 (batch is 4), respectively. In fine-tuning, SGD optimizer is applied to train our model. The initial learning rate is 0.01 and multiplied by 0.8 after each 100 epochs. Also, the basic data augmentation techniques like rotations, crops, color variations, and partial flipping are applied. The hyper-parameters related to local graph are fixed during training and testing. Experiments are performed on single GPU (RTX-2080Ti), and PyTorch 1.2.0.

Datasets	Methods	R	P	H
Total-Text	baseline	80.06	85.45	82.67
	baseline+**gcn**	83.11	85.94	84.50
CTW1500	baseline	80.57	83.06	81.80
	baseline+ **gcn**	81.45	83.75	82.58
TD500	baseline	78.52	83.24	80.81
	baseline+ **gcn**	82.30	88.05	85.08

Table 1. Ablation study for relational reasoning network. "R", "P" and "H" represent recall, precision and Hmean, respectively. For "baseline", we adopt the intersection of TR and TCL, instead of the relationship learned by GCN, to group text patches.

4.3. Ablation Study

To verify the effectiveness of the relational reasoning network, We conduct ablation experiments on Total-Text, CTW1500 and MSRA-TD500. Tab. 1 shows the experimental results on three datasets. For reducing the influence of data on the experimental results, we adopt the SynthText to pre-train model, and then we fine-tune it on Total-Text and CTW1500. Because MSRA-TD500 consists of English and Chinese, we use ICDAR2017-MLT to pre-train our network for MSRA-TD500. The longer sides of the images within Total-Text, CTW1500 and MSRA-TD500 are restricted to $1,280$, $1,024$ and 640, respectively, meanwhile keeping the aspect ratio. As shown in Tab. 1, the relational reasoning network achieves improvements by 1.83%, 0.78% and 4.27% in Hmean on Total-Text, CTW1500 and

Methods	Total-Text			CTW-1500			MSRA-TD500		
	Recall	Precision	Hmean	Recall	Precision	Hmean	Recall	Precision	Hmean
SegLink [21]	-	-	-	-	-	-	70.0	86.0	77.0
MCN [15]	-	-	-	-	-	-	79	88	83
TextSnake [17]	74.5	82.7	78.4	**85.3**	67.9	75.6	73.9	83.2	78.3
LSE† [28]	-	-	-	77.8	82.7	80.1	81.7	84.2	82.9
ATTR† [32]	76.2	80.9	78.5	-	-	-	82.1	85.2	83.6
MSR† [36]	73.0	85.2	78.6	79.0	84.1	81.5	76.7	87.4	81.7
CSE [16]	79.7	81.4	80.2	76.1	78.7	77.4	-	-	-
TextDragon [4]	75.7	85.6	80.3	82.8	84.5	83.6	-	-	-
TextField [34]	79.9	81.2	80.6	79.8	83.0	81.4	75.9	87.4	81.3
PSENet-1s† [30]	77.96	84.02	80.87	79.7	84.8	82.2	-	-	-
ICG [25]	80.9	82.1	81.5	79.8	82.8	81.3	-	-	-
LOMO*† [40]	79.3	87.6	83.3	76.5	85.7	80.8	-	-	-
CRAFT [1]	79.9	87.6	83.6	81.1	86.0	83.5	78.2	**88.2**	82.9
PAN † [31]	81.0	**89.3**	85.0	81.2	**86.4**	83.7	**83.8**	84.4	84.1
Ours	**84.93**	86.54	**85.73**	83.02	85.93	**84.45**	82.30	88.05	**85.08**

Table 2. Experimental results on Total-Text, CTW-1500 and MSRA-TD500. The symbol * means the multi-scale test is performed. The symbol † indicates the backbone network is not VGG16. The best score is highlighted in **bold**.

Figure 6. The representative samples with irregular labels on CTW-1500. Up row: the results of our method. Bottom row: the ground truth of CTW-1500.

MSRA-TD500, respectively. Remarkably, the recall of our method with relational reasoning network has improved significantly in all datasets (3.05% on Total-Text, 0.88% on CTW1500, and 3.78% on MSRA-TD500). Our method coherently improves the detection performance on MSRA-TD500 abundant with long texts (recall 3.78%, precision 4.81%, Hmean 4.27%). The performance of our method on CTW1500 is not remarkable, because its annotations are sometimes confusing. The CTW1500 has no "DO NOT CARE", so some small texts and Non-English texts are not annotated, as shown in Fig. 6 ①. Moreover, the text line annotations are confusing, as shown in Fig. 6 ② and ③.

4.4. Comparison with the state-of-the-arts

Polygon-Type Datasets. Here, ICDAR2017-MLT is used to pre-train our model, and fine-tuning is only conducted on CTW1500 and Total-Text, separately. All experiments are performed with a single image resolution.

Total-Text. This dataset mainly contains curved and multi-oriented texts, annotated in word-level. In testing, we resize the shortest side to 512 if it is less than 512, and keep the longest side is not larger than 1,280. Some visible results are listed in Fig. 7 (a) (b). From Fig. 7, we can observe that our method precisely detects word-level irregular texts, and it is can accurately separate close text instances of arbitrary shapes. The quantitative results are shown in Tab. 2. The proposed method achieves 85.73% Hmean, significantly outperforming other methods.

CTW1500. This dataset mainly contains curved and multi-oriented texts, annotated in line-level. In testing, we resize the shortest side to 512 if it is less than 512, and keep the longest side is not larger than 1,024. Some visible results are shown in Fig. 7 (c) and Fig. 6. It indicates that the proposed method correctly detects the boundaries of arbitrary shape text precisely. The quantitative results are listed in Tab. 2. Compared with the other stsate-of-the-art methods, our approach achieves promising in recall (83.02%) and Hmean (84.45%). Specifically, our method greatly outperforms TextSnake on CTW1500 and Total-Text, improves Hmean by 8.85% and 6.6% respectively.

Quadrilateral-Type Datasets. For fairly comparison, we adopt IC17 to pre-train our model, then fine-tune it on IC15 and TD500, separately. However, these datasets are evaluated with rectangular boxes, hence we need to convert the detection results into rectangular boxes. Therefore, we shrink the text instance by 0.05, and take the smallest circumscribed rectangle for evaluation.

MSRA-TD500. This dataset contains lots of long texts and text scales vary significantly. In testing, we resize the

(a) Total-Text (b) Total-Text (c) CTW1500 (d) MSRA-TD500

Figure 7. Experimental results of our method. Up row: each column shows the results of GCN clustering on different datasets. Bottom row: each column shows the corresponding results of boundary generation.

Methods	Recall	Precision	Hmean
SegLink [21]	76.8	73.1	75.0
MCN [15]	72	80	76
EAST* [42]	78.3	83.3	80.7
TextField [34]	80.05	84.3	82.4
TextSnake [17]	84.9	80.4	82.6
Textboxes++*[10]	78.5	87.8	82.9
PixelLink [3]	82.0	85.5	83.7
FOTS† [14]	82.04	88.84	85.31
PSENet-1s† [30]	84.5	86.92	85.69
LSE† [28]	**85.0**	88.3	86.6
ATRR† [32]	83.3	**90.4**	86.8
CRAFT [1]	84.3	89.8	**86.9**
Ours	84.69	88.53	86.56

Table 3. Experimental results on ICDAR2015.

shortest side to 512 if it's less than 512, and keep the longest side isn't larger than 640. Fig. 7 (d) are some representative results. The proposed method successfully detects long text lines of arbitrary orientations and sizes. The quantitative comparisons with other methods on this dataset is listed in Tab. 2. Notably, our method achieves 85.08% on Hmean, significantly outperforms other methods.

ICDARs (IC15, IC17). Considering IC15 contains many low resolution and many small text instances. The instance balance [3] is applied to assist training. The IC17 contains multilingual scene text and the annotations are given in word-level. In inference, we adjust the size of test images appropriately. For IC15, we resize the shortest side to 960 if it is less than 960, and keep the longest side is not larger than 1,960. For IC17, we resize the shortest side to 512 if it is less than 512, and keep the longest side is not larger than 2,048. The quantitative results are listed in Tab. 4 and Tab. 3. Apparently, our method achieves 86.56% Hmean on IC15 and 67.31% Hmean on IC15. The proposed method achieves competitive results against the state-of-the-art methods.

Methods	Recall	Precision	Hmean
SARI FDU RRPN[19]	55.50	71.17	62.37
He et al. [7]	57.9	76.7	66.0
Border† [35]	60.6	73.9	66.6
Lyu et al. [18]	55.6	**83.8**	66.8
FOTS† [14]	57.51	80.95	67.25
LOMO† [40]	60.6	78.8	**68.5**
Ours	**61.04**	74.99	67.31

Table 4. Experimental results on ICDAR17 MLT.

5. Conclusion

In this paper, we propose a novel CC-based method for arbitrary shape scene text detection. The proposed method adopts a spectral-based graph convolution network learn linkage relationship between the text components, and use this information to guide post-processing to connect components to text instances correctly. Experiments on five benchmarks show that the proposed method not only has good performance for arbitrary shape text detection, but also good for oriented and multilingual text. In the future, we are interested in developing an end-to-end text reading system for text of arbitrary shapes with graph network.

Acknowledgements. This work was supported by National Key R&D Program of China (No.2019YFB1405990), Beijing Natural Science Foundation (No.4194084), China Postdoctoral Science Foundation (No.2018M641199) and Fundamental Research Funds for the Central Universities (No. FRF-TP-18-060A1).

References

[1] Youngmin Baek, Bado Lee, Dongyoon Han, Sangdoo Yun, and Hwalsuk Lee. Character region awareness for text detection. In *CVPR*, pages 9365–9374, 2019.

[2] Yunpeng Chen, Marcus Rohrbach, Zhicheng Yan, Shuicheng Yan, Jiashi Feng, and Yannis Kalantidis. Graph-based global reasoning networks. In *CVPR*, pages 433–442, 2019.

[3] Dan Deng, Haifeng Liu, Xuelong Li, and Deng Cai. PixelLink: Detecting scene text via instance segmentation. In *AAAI*, pages 6773–6780, 2018.

[4] 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 *ICCV*, pages 9075–9084, 2019.

[5] Jiayuan Gu, Han Hu, Liwei Wang, Yichen Wei, and Jifeng Dai. Learning region features for object detection. In *ECCV*, pages 392–406, 2018.

[6] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. In *ICCV*, pages 2980–2988, 2017.

[7] Wenhao He, Xu-Yao Zhang, Fei Yin, and Cheng-Lin Liu. Multi-oriented and multi-lingual scene text detection with direct regression. *IEEE Trans. Image Processing*, 27(11):5406–5419, 2018.

[8] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

[9] Alex Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. *NIPS*, pages 1106–1114, 2012.

[10] 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.

[11] Minghui Liao, Zhen Zhu, Baoguang Shi, Gui-Song Xia, and Xiang Bai. Rotation-sensitive regression for oriented scene text detection. In *CVPR*, pages 5909–5918, 2018.

[12] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. In *CVPR*, pages 936–944, 2017.

[13] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: Single shot multibox detector. In *ECCV*, pages 21–37, 2016.

[14] Xuebo Liu, Ding Liang, Shi Yan, Dagui Chen, Yu Qiao, and Junjie Yan. FOTS: Fast oriented text spotting with a unified network. In *CVPR*, pages 5676–5685, 2018.

[15] Zichuan Liu, Guosheng Lin, S.Yang, Jiashi Feng, Weisi Lin, and Wang Ling Goh. Learning markov clustering networks for scene text detection. In *CVPR*, pages 6936–6944, 2018.

[16] Zichuan Liu, Guosheng Lin, Sheng Yang, Fayao Liu, Weisi Lin, and Wang Ling Goh. Towards robust curve text detection with conditional spatial expansion. In *CVPR*, pages 7269–7278, 2019.

[17] Shangbang Long, Jiaqiang Ruan, Wenjie Zhang, Xin He, Wenhao Wu, and Cong Yao. Textsnake: A flexible representation for detecting text of arbitrary shapes. In *ECCV*, pages 19–35, 2018.

[18] Pengyuan Lyu, Cong Yao, Wenhao Wu, Shuicheng Yan, and Xiang Bai. Multi-oriented scene text detection via corner localization and region segmentation. In *CVPR*, pages 7553–7563, 2018.

[19] Jianqi Ma, Weiyuan Shao, Hao Ye, Li Wang, Hong Wang, Yingbin Zheng, and Xiangyang Xue. Arbitrary-oriented scene text detection via rotation proposals. *IEEE Trans. Multimedia*, 20(11):3111–3122, 2018.

[20] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(6):1137–1149, 2017.

[21] Baoguang Shi, Xiang Bai, and Serge J. Belongie. Detecting oriented text in natural images by linking segments. In *CVPR*, pages 3482–3490, 2017.

[22] Abhinav Shrivastava, Abhinav Gupta, and Ross B. Girshick. Training region-based object detectors with online hard example mining. In *CVPR*, pages 761–769, 2016.

[23] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[24] Lei Sun, Qiang Huo, Wei Jia, and Kai Chen. A robust approach for text detection from natural scene images. *Pattern Recognition*, 48(9):2906–2920, 2015.

[25] Jun Tang, Zhibo Yang, Yongpan Wang, Qi Zheng, Yongchao Xu, and Xiang Bai. Seglink++: Detecting dense and arbitrary-shaped scene text by instance-aware component grouping. *Pattern Recognition*, 96, 2019.

[26] Shu Tian, Xu-Cheng Yin, Ya Su, and Hong-Wei Hao. A unified framework for tracking based text detection and recognition from web videos. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(3):542–554, 2018.

[27] Zhi Tian, Weilin Huang, Tong He, Pan He, and Yu Qiao. Detecting text in natural image with connectionist text proposal network. In *ECCV*, pages 56–72, 2016.

[28] Zhuotao Tian, Michelle Shu, Pengyuan Lyu, Ruiyu Li, Chao Zhou, Xiaoyong Shen, and Jiaya Jia. Learning shape-aware embedding for scene text detection. In *CVPR*, pages 4234–4243, 2019.

[29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, pages 5998–6008, 2017.

[30] Wenhai Wang, Enze Xie, Xiang Li, Wenbo Hou, Tong Lu, Gang Yu, and Shuai Shao. Shape robust text detection with progressive scale expansion network. In *CVPR*, pages 9336–9345, 2019.

[31] Wenhai Wang, Enze Xie, Xiaoge Song, Yuhang Zang, Wenjia Wang, Tong Lu, Gang Yu, and Chunhua Shen. Efficient and accurate arbitrary-shaped text detection with pixel aggregation network. In *ICCV*, pages 8439–8448, 2019.

[32] Xiaobing Wang, Yingying Jiang, Zhenbo Luo, Cheng-Lin Liu, Hyunsoo Choi, and Sungjin Kim. Arbitrary shape scene text detection with adaptive text region representation. In *CVPR*, pages 6449–6458, 2019.

[33] Zhongdao Wang, Liang Zheng, Yali Li, and Shengjin Wang. Linkage based face clustering via graph convolution network. In *CVPR*, pages 1117–1125, 2019.

[34] Yongchao Xu, Yukang Wang, Wei Zhou, Yongpan Wang, Zhibo Yang, and Xiang Bai. Textfield: Learning a deep direction field for irregular scene text detection. *IEEE Trans. Image Processing*, 28(11):5566–5579, 2019.

[35] Chuhui Xue, Shijian Lu, and Fangneng Zhan. Accurate scene text detection through border semantics awareness and bootstrapping. In *ECCV*, pages 370–387, 2018.

[36] Chuhui Xue, Shijian Lu, and Wei Zhang. MSR: multi-scale shape regression for scene text detection. In *IJCAI*, pages 989–995, 2019.

[37] Xu-Cheng Yin, Wei-Yi Pei, Jun Zhang, and Hong-Wei Hao. Multi-orientation scene text detection with adaptive clustering. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(9):1930–1937, 2015.

[38] Xu-Cheng Yin, Xuwang Yin, Kaizhu Huang, and Hong-Wei Hao. Robust text detection in natural scene images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(5):970–983, 2014.

[39] Xu-Cheng Yin, Ze-Yu Zuo, Shu Tian, and Cheng-Lin Liu. Text detection, tracking and recognition in video: A comprehensive survey. *IEEE Trans. Image Processing*, 25(6):2752–2773, 2016.

[40] Chengquan Zhang, Borong Liang, Zuming Huang, Mengyi En, Junyu Han, Errui Ding, and Xinghao Ding. Look more than once: An accurate detector for text of arbitrary shapes. In *CVPR*, pages 10552–10561, 2019.

[41] Zheng Zhang, Wei Shen, Cong Yao, and Xiang Bai. Symmetry-based text line detection in natural scenes. In *CVPR*, pages 2558–2567, 2015.

[42] Xinyu Zhou, C.Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, and Jiajun Liang. EAST: An efficient and accurate scene text detector. In *CVPR*, pages 2642–2651, 2017.