An Accurate Segmentation-Based Scene Text Detector with Context Attention and Repulsive Text Border

Xi Liu, Gaojing Zhou, Rui Zhang, Xiaolin Wei
Meituan-Dianping Group, Beijing, China
{liuxi12, zhougaojing, zhangrui36, weixiaolin03}@meituan.com

Abstract

Scene text detection is one of the most challenging problems in computer vision and has attracted great interest. In general, scene text detection methods are divided into two categories: detection-based and segmentation-based methods. Recently, the segmentation-based methods are more and more popular due to their superior performances and the advantages of detecting arbitrary-shape texts. However, there still exist the following problems: (a) the misclassification of the unexpected texts, (b) the split of long text lines, (c) the failure of separating very close text instances. In this paper, we propose an accurate segmentation-based detector, which is equipped with context attention and repulsive text border. The context attention incorporates global channel attention, non-local self-attention and spatial attention to better exploit the global and local context, which can greatly increase the discriminative ability for pixels. Due to the enhancement of pixel-level features, false positives and the misdetections of long texts are reduced. Besides, for the purpose of solving very close text instance, a repulsive pixel link, which focuses on the relationships between pixels at the border, is proposed. Experiments on several standard benchmarks, including MSRA-TD500, ICDAR2015, ICDAR2017-MLT and CTW1500, validate the superiority of the proposed method.

1. Introduction

Scene text detection, which refers to precisely localizing all the instances of texts in a scene image, has been widely studied. It is a critical step in many text-related real-world applications, such as photo translation [1], autonomous driving, image retrieval [14] and augmented reality. It is quite challenging due to the large variations of color, size, aspect ratio, font, orientation, lighting conditions and background in scene texts [54].

With the development of deep learning, great progress has been made in the computer vision tasks such as object detection and segmentation [9, 10, 13, 21, 25, 42, 44, 45]. Scene text detection, which can be seen as a type of object detection applied to text, has also witnessed great success [11, 22, 23, 26, 27, 28, 30, 32, 33, 43, 59, 61]. In general, scene text detection methods can be divided into two categories: detection-based and segmentation-based methods. The detection-based methods adapt the general object detection framework to detect the text or text parts by directly regressing rectangles or quadrangles with certain orientations. However, these frameworks cannot detect the text instances with arbitrary shapes and often fail to detect small texts. The segmentation-based methods use pixel-wise segmentation to segment text areas and extract text instances by post-processing the segmented areas. They have gained more interest due to their advantages of detecting arbitrary-shape texts and the superior performances compared with detection-based methods. However, there still exist several problems. The first one is the misclassification of the unexpected texts or text-like patterns. The second one is the split of the long text line into several text instances. The third one is the failure of separating very close text instances. Some examples are shown in Fig. 1 (a)(b)(c).

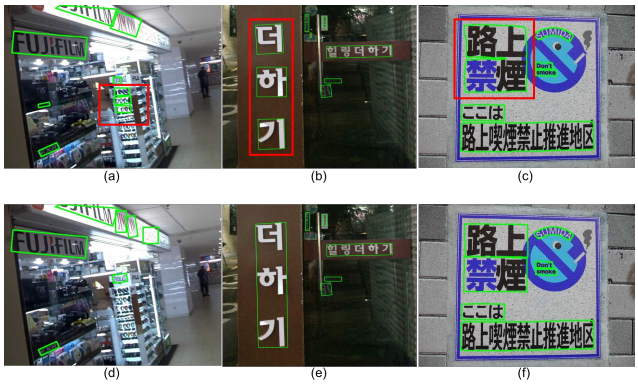

Figure 1. Different types of problems in scene text detection and the results of our method. Note that the error detections are marked with Red boxes. (a) is the misclassification of the unexpected texts, (b) is the split of long text lines, (c) is the failure of separating very close text instances. (d), (e), (f) are the results of our method, which successfully solves the problems.

To address these problems, in this paper, we propose an accurate segmentation-based text detector. Two modules:

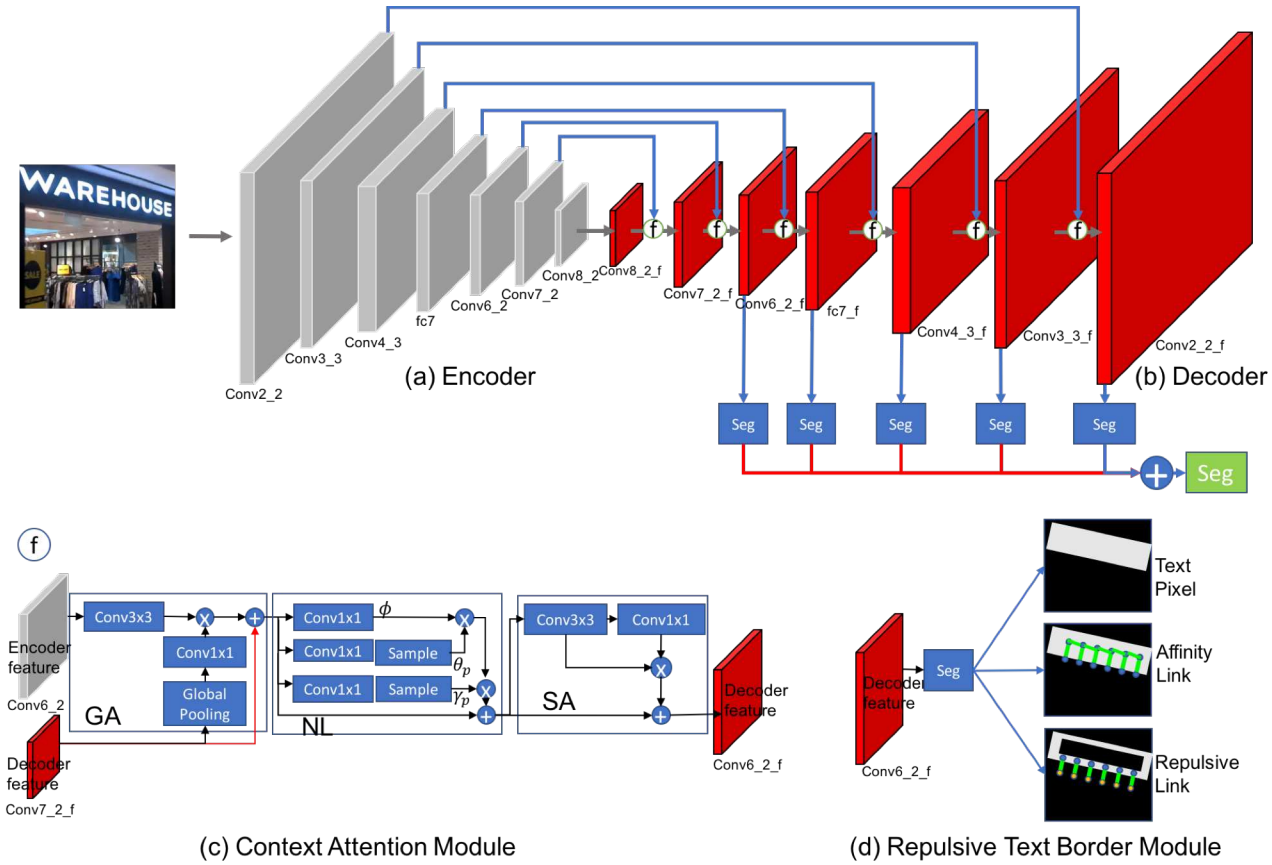

Figure 2. Architecture of the proposed method. The network consists of (a) Encoder, (b) Decoder, (c) Context Attention Module, GA is global attention, NL is non-local self-attention, and SA is spatial attention, (d) Repulsive Text Border Module. The red arrow line means upsample.

context attention module and repulsive text border module are specifically introduced. First, context plays a critical role in segmentation since it is very helpful for reducing local ambiguities for pixel classification. We design an effective attention mechanism to better exploit the context information by sequentially applying global attention, non-local self-attention and spatial attention. The global attention uses global average-pooled high-level decoder features to compute channel-wise attention to the low-level encoder features, which increases the discriminative ability of low-level features. Non-local attention mechanism is proved to be effective for capturing long range dependencies. For the long text lines detection, long range contextual information is necessary to avoid the split of long text line into several text instances. We use a simple yet effective non-local module introduced in the work of [60] as our non-local attention module. It embeds a pyramid sampling module into non-local blocks to largely reduce the computation. The spatial attention utilizes the local inter-spatial relationship of features and focuses on 'where' is text, which further solve the false positives. It applies a convolution layer with one channel to generate a spatial - attention map and enhances the input features by broadcasting the attention map. As shown in Fig. 1(d)(e), our method can successfully solve the false positive and the split of long text. Second, text border is key to separating very close text lines. In PixelLink [3], it learns two kinds of pixel-wise predictions: text/non-text prediction and link prediction. The pixel link is important for separating text instance since texts are detected by linking pixels within the same text instance. However, the pixel link generally pays attention to the link between neighbor pixels that belong to the same text instance. Note that the link between pixels located at the text border requires more attention. Therefore, we introduce an extra repulsive pixel link that explicitly represents the relationship between two pixels at the text border. Predicted positive pixels are then joined together by predicted positive pixel links and negative repulsive links. Fig. 1(f) is the result of our method which shows that the very close text instances can be separated.

To validate the effectiveness of our proposed scene text detector, we conduct extensive experiments on four standard benchmarks and achieve an F-measure of 86.1%

on MSRA-TD500, 87.5% on ICDAR2015, 75.3% on ICDAR2017-MLT, 82.0% on CTW1500. The experimental results show that our method outperforms most of the state-of-art methods. The contributions of this paper can be summarized as follows:

(1) We propose an effective attention mechanism to better exploit the context information, which can effectively reduce the false positives and avoid the split of long text line into several text instances.

(2) To further solve the very close text instance, we propose to learn an extra repulsive pixel link that explicitly represents the relationship between pixels located at text border.

(3) The proposed method achieves state-of-the-art performance on several benchmark datasets of scene text including long straight, horizontal, multi-oriented and curved text.

2. Related Work

Scene text detection has been extensively studied in the last decades. State-of-the-art text detection algorithms are deep neural network based methods. Most of the deep learning based text detection methods can roughly be divided into two branches: detection-based and segmentation-based approaches.

Detection-based methods treat text as a specific object and take advantage of the development in general object detection. Zhong et al. [57] proposed a text detection framework based on Faster-RCNN. They designed an inception-RPN which used multi-scale convolution filters to produce text region proposals. Ma et al. [34] added rotation to both anchors and RoIPooling in Faster R-CNN, to deal with the orientation of scene text. Gupta et al. [6] borrowed the YOLO [41] framework and employed a fully-convolutional regression network to perform text detection and bounding box regression at all locations and multiple scales of an image. TextBoxes [22] modified anchors and kernels of SSD to detect large aspect-ratio scene text. TextBoxes++ [20] extended TextBoxes by regressing quadrilaterals instead of horizontal bounding boxes to handle arbitrary-oriented text. Shi et al. [43] employed SSD framework and learned the locally detectable text elements, namely segments and links. RRD [23] also relied on SSD framework and introduced rotation-sensitive feature for detection branch and rotation-invariant feature for classification branch to learn better regression of long oriented text. These methods always need complex anchor setting and fail to detect texts with arbitrary shapes.

Segmentation-based methods are mostly inspired by fully convolutional networks (FCN) [31]. Zhang et al. [56] first presented a framework which used FCN to produce a coarse saliency map for text. Yao et al. [53] casted the detection task as a segmentation problem by predicting three kinds of score maps: text/non-text, character classes, and character linking orientations. PixelLink [3] performed pixel-wise text/non-text and link prediction, then added some post-processing on the linked positive pixels to obtain the final text boxes. PSENet [18] used FCN to predict text instances with multiple scales, then designed a progressive scale expansion algorithm to reconstruct the whole text instance. More recently, several works such as Mask Text Spotter [32] and SPCNet [50] borrowed the state-of-art instance segmentation approach Mask R-CNN to detect text instances and achieved impressive performance. The biggest advantage of these methods is the ability to extract arbitrary-shape texts. However, their performances are greatly affected by the segmentation results.

Compared with previous works, our method incorporates context attention and repulsive text border to improve text detection performance. Relying on the context information, the misclassification of the unexpected texts or text-like patterns and the split of long text lines are greatly reduced, which are common issues for most of segmentation-based methods. Moreover, the proposed repulsive pixel link that explicitly represents the relationship between two pixels at the text border are verified to be effective for separating the very close text instances.

3. Approach

In this section, we describe our proposed method in detail. Firstly, we present the general framework of our method. Secondly, we elaborate the context attention and repulsive text border modules. Finally, the training and inferring details are presented.

3.1. Overall Architecture

The network architecture of our approach is illustrated in Fig. 2. It is based on a fully convolutional network with encoder-decoder structure. In the encoder part, VGG-16 is used as backbone and the last two layers fc6 and fc7 are converted from fully-connected layers into convolutional layers. Besides, three extra layers are added after fc7 layer in the same manner as SSD [25]. In the decoder part, the output feature maps are generated by fusing low-level decoder features with high-level encoder features. The fusing process is implemented by introducing a context attention module. As shown in Fig. 2(c), the context attention module uses global attention, non-local self-attention and spatial attention to effectively model the local and global context, which will be detailed in Section 3.2. For each output feature map of the decoder (conv2_2_f, conv3_3_f, conv4_3_f, fc7_f, conv6_2_f), three sibling 1x1 convolution and softmax layers are attached to generate three score maps for text pixel, affinity pixel link and repulsive pixel link (see Fig. 2(d)). Since every pixel has 8 neighbors, the output score maps have 2, 16 and 16 channels, respectively. The details of learning pixel links are presented in Section 3.3. Finally, the score maps of each output feature map are resized and added together to obtain

three segmentation masks: text pixel mask, affinity link and repulsive link masks. Based on the segmentation results, we join the positive pixels with positive pixel links and negative repulsive links together, and obtain the detection results by extracting the bounding boxes of the connected components.

3.2. Context Attention

Context plays a critical role in segmentation since it is helpful for reducing local ambiguities for pixel classification. In our context attention, there are three sub-modules: global attention, non-local self-attention and spatial attention. Given the low-level encoder feature map $F_{low} \in \mathbb{R}^{C\times H\times W}$ and the high-level decoder feature map $F_{high} \in \mathbb{R}^{C'\times H'\times W'}$ as input, the context attention module sequentially goes through 1D channel attention, non-local self-attention and 2D spatial attention to generate the output feature map $F_{CA} \in \mathbb{R}^{C'\times H\times W}$, as illustrated in Fig.2(c). The overall process can be summarized as:

$$F_{GA} = GA(F_{low}, F_{high}), \quad (1)$$

$$F_{NL} = NL(F_{GA}), \quad (2)$$

$$F_{CA} = F_{SA} = SA(F_{NL}), \quad (3)$$

where $GA(\cdot)$ is global attention, $NL(\cdot)$ is non-local self-attention, and $SA(\cdot)$ is spatial attention.

Global Attention Module. High-level features always contain rich text category information, which can be a good guidance for low-level features to select text localization details.

We perform global average pooling on the high-level decoder features $F_{high} \in \mathbb{R}^{C'\times H'\times W'}$ and a 1×1 convolution over the pooled features to generate the global attention map. The low-level encoder features $F_{low} \in \mathbb{R}^{C\times H\times W}$ are then multiplied by the attention map. Note that the channel number of the attention map and the low-level features may be different. A 3×3 convolution is added to the low-level features. Finally, the high-level features are upsampled and added with the weighted low-level features to get the output features $F_{GA} \in \mathbb{R}^{C'\times H\times W}$. In short, the output feature is computed as:

$$F_{GA} = GA(F_{low}, F_{high}) = GAttMap \odot Conv_{3\times3}(F_{low}) + UP(F_{high}), \quad (4)$$

$$GAttMap = Conv_{1\times1}\left(AvgPool(F_{high})\right), \quad (5)$$

where $\odot$ represents element-wise multiplication, $UP(\cdot)$ is upsample operation.

Non-local Self-Attention Module. Non-local attention is potent to capture the long range dependencies that are crucial for pixel classification. Especially for the long text lines, long range contextual information is necessary to avoid the split of long text line into several text instances.

Considering the large computation of non-local operation, we use a simple yet effective non-local module introduced in the work of [60]. Given the output feature $F_{GA} \in \mathbb{R}^{C'\times H\times W}$ of the global attention module as input, three 1x1 convolutions are first used to transform the input to different embeddings: $\phi \in \mathbb{R}^{\hat{C}\times H\times W}$, $\theta \in \mathbb{R}^{\hat{C}\times H\times W}$ and $\gamma \in \mathbb{R}^{\hat{C}\times H\times W}$. Spatial pyramid pooling is then applied after θ and γ to get sampled θ_p and γ_p.

The ϕ, θ_p and γ_p are flattened to $\phi \in \mathbb{R}^{\hat{C}\times N}$, $\theta_p \in \mathbb{R}^{\hat{C}\times S}$, $\gamma_p \in \mathbb{R}^{\hat{C}\times S}$. A normalized similarity matrix is calculated as:

$$\overrightarrow{V_p} = f(\phi^T \times \theta_p), \quad (6)$$

where the normalizing function f can take the form from softmax, rescaling, and none. The attention output is acquired by

$$O_p = \overrightarrow{V_p} \times {\gamma_p}^T, \quad (7)$$

and the final output $F_{NL} \in \mathbb{R}^{C'\times H\times W}$ is given by

$$F_{NL} = Reshape(W_o(O_p^T) + F_{GA}), \quad (8)$$

where W_o is a 1x1 convolution operation to recover the channel dimension from $\hat{C}$ to C'.

Spatial Attention Module. The spatial attention utilizes the local inter-spatial relationship of features and focuses on 'where' is text, which further solve the false positives.

Given the output feature F_{NL} of the non-local self-attention module as input, we perform a 3×3 convolution and then a 1×1 convolution with one channel to generate a text saliency map. A sigmoid function is further applied to obtain the spatial attention map $SAttMap \in \mathbb{R}^{H\times W}$. The attention output O_s is calculated as:

$$O_s = Broadcast(SAttMap) \odot Conv_{3\times3}(F_{NL}), \quad (9)$$

$$SAttMap = Sigmoid(Conv_{1\times1}(Conv_{3\times3}(F_{NL}))), \quad (10)$$

where $SAttMap$ is broadcast to the same C' channel as F_{NL}, $\odot$ represents element-wise multiplication. The output $F_{SA} \in \mathbb{R}^{C'\times H\times W}$ of the spatial attention, also the final output $F_{CA} \in \mathbb{R}^{C'\times H\times W}$ of the context attention, is given by

$$F_{CA} = F_{SA} = F_{NL} + O_s. \quad (11)$$

3.3. Repulsive Text Border

Text border is critical for scene text detection since the border is actually the splitting mark for different text instances. Especially for the very close text instances and the curved texts, which often appear in scene text, more accurate text border is required. Inspired by the work of PixelLink [3], which learns 8-neighbor links for a pixel and uses the links to determine the text border, we also use 8-neighbor link to learn the text border. We introduce two kinds of 8-neighbor links: affinity and repulsive pixel links for each pixel.

As shown in Fig. 3(a), for a given pixel and one of its neighbors, if they lie within the same text instance, the affinity pixel link between them is labeled as positive, and otherwise negative. We only focus on the positive pixels and the loss for affinity pixel links is calculated by:

$$L_{alink} = \frac{L_{alink\_pos}}{sum(alink\_pos)} + \frac{L_{alink\_neg}}{sum(alink\_neg)}, \quad (12)$$

where L_{alink_pos} and L_{alink_neg} are the cross-entropy losses

on the positive and negative affinity links, respectively; $sum(alink\_pos)$ and $sum(alink\_neg)$ are the number of the positive and negative affinity links, respectively.

The affinity pixel links generally pay attention to the link between neighbor pixels that belong to the same text instance. However, the links between pixels located at the text border require more attention. As illustrated in Fig. 3(b), we shrink the annotated text box G with the offset D to G_d and consider the gap between G and G_d as the text border (gray area in Fig. 3(b)). The offset D is computed from the perimeter L and area A of the box G:

$$D = \frac{A(1-r^2)}{L}, \tag{13}$$

where r is the shrink ratio, set to 0.4 empirically. We only focus on the positive pixels in the text border and ignore the other positive pixels. For a pixel in the text border and one of its neighbors, if they lie within different text instances or the neighbor pixel is non-text, the repulsive pixel link between them is labeled as positive, and otherwise negative. Similarly, we also use class-balanced cross-entropy loss as the loss for repulsive pixel links:

$$L_{rlink} = \frac{L_{rlink\_pos}}{sum(W_{rlink\_pos})} + \frac{L_{rlink\_neg}}{sum(W_{rlink\_pos})}, \tag{14}$$

where L_{rlink_pos} and L_{rlink_neg} are the cross-entropy losses on the positive and negative repulsive links, respectively; $sum(W_{rlink_pos})$and $sum(W_{rlink\_neg})$ are the sum of the weighted positive and negative repulsive links, respectively. For the positive repulsive links in which the two neighbor pixels lie in two text instances, they are assigned larger weight (2.0) while for other repulsive links, their weight is set to 1.0.

3.4. Training and Inference

The objective function of learning pixels and links is defined as follows:

$$L_{seg} = \lambda L_{pixel} + L_{alink} + L_{rlink}, \tag{15}$$

where L_{pixel} is the loss on pixel classification task, L_{alink} and L_{rlink} are the link losses. λ is the weight of pixel loss and set to 2.0.

Considering the extreme imbalance of text and non-text pixels, we use online hard example mining (OHEM) to select negative pixels and adopt the weighted cross-entropy loss to supervise pixel classification:

$$L_{pixel} = \frac{1}{(1+r)S} W L_{pixel\_CE}, \tag{16}$$

where L_{pixel_CE} is the cross-entroy loss on text/non-text prediction, r is the negative-positive ratio and is set to 3. S is the total number of the positive pixels. W is pixel weight matrix. For the negative pixels, their weights are set to 1.0, and for each positive pixel i, its weight is calculated as:

$$w_i = \frac{S}{N \cdot S_i}, \tag{17}$$

where N is the number of text instances, S_i is the number of pixels of the text instance that the positive pixel lies in.

Given predictions on pixels, affinity links and repulsive links, three different thresholds are applied to them. The pixel above the pixel threshold is regarded as positive. The link between two neighbor pixels is regarded as positive if the affinity link score is above the affinity link threshold and the repulsive link score is below the repulsive link threshold. Positive pixels are then grouped together using positive links, resulting in a collection of text instances.

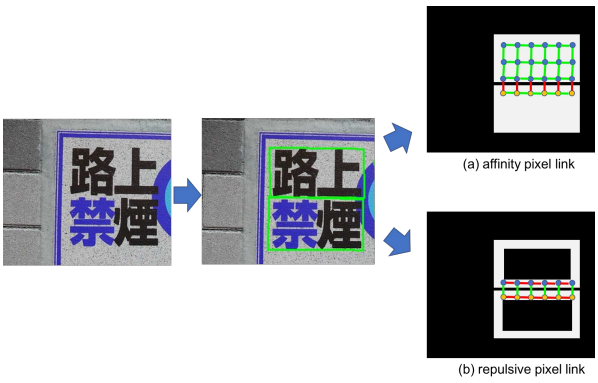

Figure 3. An illustration of affinity and repulsive pixel link. Green lines in (a) and (b) denote positive affinity and repulsive pixel links, respectively; red lines in (a) and (b) denote negative affinity and repulsive pixel links, respectively.

4. Experiments

We evaluate our method on four public datasets: MSRA-TD500 [52], ICDAR2015[15], ICDAR2017-MLT [37] and CTW1500 [29], and compare it with several state-of-art methods.

4.1. Datasets

SynthText [6] is a synthetically generated dataset containing 800 thousand images and about 8 million word instances. It is created by blending natural images with texts of random sizes and fonts. We only use the dataset for pre-training our network.

MSRA-TD500 [52] includes 300 training images and 200 test images collected from natural scenes. It is a dataset with multilingual, arbitrary-oriented and long text lines.

ICDAR2015 [15] is the most commonly used benchmark for detecting scene text in arbitrary directions. It contains 1000 training images and 500 testing images. The images are collected by Google Glass without taking care of positioning, image quality, and viewpoint. Therefore, text in these images is of various scales, orientations, contrast, blurring, and viewpoint, making it challenging for detection. Annotations are provided as word quadrilaterals.

ICDAR2017-MLT [37] is a large-scale multilingual text dataset, which includes 7200 training images, 1800 validation images and 9000 test images. The dataset

consists of scene text images which come from 9 languages. Image annotations are labeled as word-level quadrangles.

CTW1500 [29] is a recent challenging dataset for curve text detection. It has 1000 training images and 500 testing images with over 10 thousand text annotations. Text instances are annotated by 14 vertices of polygons.

4.2. Implementation Details

We pre-train our network on SynthText and then finetune it on the real datasets. The models are optimized by SGD with momentum = 0.9. For training, images are resized to 512*512 after random cropping. Batch size is set to 12 owing to the GPU memory limitation and the learning rate is fixed to 1e-4 and set to 1e-5 for the last several epochs. VGG16 is used as the backbone of our network. Thresholds on pixel and links are crucial for detecting performance. We find the thresholds for each dataset via a grid search with 0.05 step on a hold-out validation set. The whole algorithm is implemented in Tensorflow 1.8.0 and pure Python.

4.3. Ablation Study

To verify the effectiveness of our design, we conduct all experiments of ablation studies on the ICDAR2015 dataset (an oriented text dataset) and CTW1500 dataset (a curved text dataset). The scale of test image for ICDAR2015 and CTW1500 is 1280x768.

Baseline. We implement the method with no context attention and only affinity pixel link as our baseline method.

Context Attention. We implement the model with context attention and only affinity pixel link. Considering that there are three modules in context attention, we implement three models: GA, GA+NL, GA+NL+SA. From Tab. 1, the GA achieves 2.2% improvement on ICDAR2015 and 1.5% improvement on CTW1500 than baseline; the GA+NL achieves 0.5% improvement on ICDAR2015 and 1.4% improvement on CTW1500 than GA; the GA+NL+SA achieves 0.9% improvement on ICDAR2015 and 1.1% improvement on CTW1500 than GA+NL. The results demonstrate that the attention modules used in context attention are all useful. Overall, the model with context attention makes 3.6% improvement on ICDAR2015 and 4.0% improvement on CTW1500.

The effectiveness of repulsive link. To investigate the effectiveness of repulsive link, we implement the model (GA+NL+SA+RL) with context attention and the affinity and repulsive link. From Tab. 1, the model with repulsive link achieves 0.4% improvement on ICDAR2015 and 0.7% improvement on CTW1500, in comparison to the model without repulsive link (GA+NL+SA).

Method	ICDAR2015			CTW1500		
	P	R	F	P	R	F
Baseline	85.1	82.0	83.5	81.1	73.9	77.3
GA	87.5	83.9	85.7	82.8	75.1	78.8
GA+NL	88.0	84.5	86.2	83.9	76.8	80.2
GA+NL+ SA	89.7	84.6	87.1	85.3	77.7	81.3
GA+NL+ SA+RL	90.0	85.1	87.5	85.8	78.6	82.0

Table 1. Ablation experiments of validating the effectiveness of different modules on ICDAR2015 and CTW1500 dataset. "GA" means global attention, "GA+NL" means global attention + non-local self-attention, "GA+NL+SA" means context attention, "GA+NL+SA+RL" means context attention + repulsive link.

4.4. Results on Scene Text Benchmarks

Long straight text detection. We evaluate the performance of our method on MSRA-TD500, which contains multi-lingual, arbitrary-oriented and long text lines. Images are resized to 768x768 for testing. Thresholds of text pixel, affinity pixel link and repulsive pixel link are set to (0.9, 0.85, 0.8). As shown in Tab. 2, our method achieves F-measure of 86.1%, which is better than all the other methods. The results also demonstrate the advantages of our method for dealing with long text lines. Some of the detection results are visualized in Fig. 4(a).

Oriented text detection. We evaluate our method on the ICDAR 2015 to test its ability of detecting oriented text. Thresholds of text pixel, affinity pixel link and repulsive link are set to (0.85,0.85,0.8). We use a single scale of 1280x768 for test images and achieve 90.0, 85.1 and 87.5 in precision, recall and F-measure, respectively. As shown in Tab. 3, except for the end-to-end method FOTS which combines text detection and recognition, our method outperforms the state-of-art methods. Also note that the very high precision (90.0%) is obtained, which verifies that our method can suppress false positives effectively. Some of the detection results are visualized in Fig. 4(b).

Multilingual text detection. To verify the generalization ability of our method on multilingual scene text detection, we evaluate our method on ICDAR2017-MLT. We use a single scale of 1536x1536 for test images. The 7200 training images are used for training and the 1800 validation images are used for selecting the models and thresholds. Thresholds of text pixel, affinity link and repulsive link are set to (0.9,0.45,0.8). We achieve an F-measure of 75.3%, which is comparable to the best reported result in literature. Some of the detection results are visualized in Fig. 4(c).

Curved text detection. We evaluate the ability of our model to detect curved text on CTW1500 dataset. Our method can be flexibly applied to curved text without special modifications. The only modification lies in the interface of reading text polygons with 14 vertices. We use a single scale of 1280x768 for test images. Thresholds of text pixel, affinity link and repulsive link are set to

(0.75,0.8,0.8). As shown in Tab. 5, our method achieves the state-of-the-art results and outperforms some existing methods such as TextSnake [30] and LOMO [55]. Some of the detection results are visualized in Fig. 4(d).

Method	Precision	Recall	F-measure
RRPN [34]	82.0	68.0	74.0
SegLink [43]	86.0	70.0	77.0
PixelLink [3]	83.0	73.2	77.8
Lyu et al. [33]	87.6	76.2	81.5
MCN [27]	88.0	79.0	83.0
PAN [48]	84.4	83.8	84.1
OURS	88.8	83.5	86.1

Table 2. Quantitative results of different methods on MSRA-TD500 (**long straight text**) dataset. Our method achieves the best performance over all the other methods, showing the advantages of dealing with long text lines.

Method	Precision	Recall	F-measure
SegLink[43]	73.1	76.8	75.0
RRPN[34]	84.0	77.0	80.0
EAST[59]	83.3	78.3	80.7
TextBoxes++ [20]	87.2	76.7	81.7
TextSnake [30]	84.9	80.4	82.6
PixelLink [3]	85.5	82.0	83.7
PSENet-1s [18]	86.9	84.5	85.7
Mask Textspotter [32]	91.6	81.0	86.0
LOMO [55]	91.3	83.5	87.2
SPCNet [50]	88.7	85.8	87.2
FOTS [26]	-	-	88.0
OURS	90.0	85.1	87.5

Table 3. Quantitative results of different methods on ICDAR 2015 (**oriented text**) dataset. Except for the end-to-end method FOTS, our method outperforms all the other methods.

Method	Precision	Recall	F-measure
E2E-MLT [38]	64.6	53.8	58.7
He et al. [12]	76.7	57.9	66.0
Lyu et al. [33]	83.8	56.6	66.8
FOTS [26]	81.0	57.5	67.3
Border [51]	77.7	62.1	69.0
AF-RPN [58]	75.0	66.0	70.0
PSENet-1s [18]	77.0	68.4	72.5
LOMO MS [55]	80.2	67.2	73.1
SPCNet [50]	80.6	68.6	74.1
OURS	83.7	68.4	75.3

Table 4. Quantitative results of different methods on ICDAR2017-MLT (**multilingual text**) dataset. MS means multi-scale testing.

Method	Precision	Recall	F-measure
SegLink [43]	42.3	40.0	40.8
EAST [59]	78.7	49.1	60.4
CTD [29]	74.3	65.2	69.5
CTD+TLOC [29]	77.4	69.8	73.4
TextSnake [30]	67.9	85.3	75.6
LOMO MS [55]	85.7	76.5	80.8
PSENet-1s [18]	84.8	79.7	82.2
OURS	85.8	78.6	82.0

Table 5. Quantitative results of different methods on CTW1500 (**curved text**) dataset.

5. Conclusion and Future Work

In this paper, we propose an accurate segmentation-based scene text detector with context attention and repulsive text border. We design an effective attention mechanism to better exploit the context information by sequentially applying global attention, non-local self-attention and spatial attention. The context is helpful for reducing local ambiguities for pixel classification, which can greatly reduce false positives and the misdetections of

(a) (b) (c) (d)

Figure 4. Examples of detection results. From left to right: (a) MSRA-TD500, long straight text, (b) ICDAR2015, oriented text, (c) ICDAR2017-MLT, multilingual text, (d) CTW1500, curved text.

long text lines. To further solve the very close text instance, we propose to learn an extra repulsive pixel link that explicitly represents the relationship between pixels located at text border. The robustness and effectiveness of our approach are verified on several public benchmarks including long, curved, oriented and multilingual text cases. In the future, we would like to further focus on the text border and develop a two-stream segmentation network to simultaneously learn text pixels and text boundaries.

References

[1] A. Bissacco, M. Cummins, Y. Netzer, and H. Neven. Photoocr: Reading text in uncontrolled conditions. In ICCV, 2013.

[2] J. L. Cao, Y. W. Pang, and X. L. Li. Triply Supervised Decoder Networks for Joint Detection and Segmentation. In CVPR, 2019.

[3] D. Deng, H. Liu, X. Li, and D. Cai. Pixellink: Detecting scene text via instance segmentation. In AAAI, 2018.

[4] M. En, Rong Li, J. Li, B. Liu. Feature Pyramid Based Scene Text Detector. In ICDAR, 2017.

[5] R. Girshick. Fast R-CNN. In ICCV, 2015.

[6] A. Gupta, A. Vedaldi, and A. Zisserman. Synthetic data for text localisation in natural images. In CVPR, 2016.

[7] K. He, G. Gkioxari, P. Dollar, and R. Girshick. Mask r-cnn. In ICCV, 2017.

[8] T. He, W. Huang, Y. Qiao and J. Yao. Accurate text localization in natural image with cascaded convolutional text network. arXiv, 2016.

[9] K. He, X. Zhang, S. Ren, J. Sun. Deep residual learning for image recognition. In CVPR, 2016.

[10] K. He, X. Zhang, S. Ren, J. Sun. Identity mappings in deep residual networks. In ECCV, 2016.

[11] W. He, X. Zhang, F. Yin, and C. Liu. Deep direct regression for multi-oriented scene text detection. In ICCV, 2017.

[12] W. He, X.-Y. Zhang, F. Yin, and C.-L. Liu. Multi-oriented and multi-lingual scene text detection with direct regression. IEEE Transactions on Image Processing, 27(11):5406–5419, 2018.

[13] G. Huang, Z. Liu, K.Q. Weinberger, L. van der Maaten. Densely connected convolutional networks. In CVPR, 2017.

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

[15] D. Karatzas, L. Gomez-Bigorda, A. Nicolaou, S. K. Ghosh, A. D. Bagdanov, M. Iwamura, J. Matas, L. Neumann, V. R. Chandrasekhar, S. Lu, F. Shafait, S. Uchida, and E. Valveny. ICDAR 2015 competition on robust reading. In ICDAR, 2015.

[16] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L. G. i Bigorda, S. R. Mestre, J. Mas, D. F. Mota, J. Almazan, and L. de las Heras. ICDAR 2013 robust reading competition. In ICDAR, 2013.

[17] Z. Li, C. Peng, G. Yu, X. Zhang, Y. Deng, J. Sun. DetNet: A backbone network for object detection. arXiv:1804.06215, (2018).

[18] X. Li,W. H. Wang, W. B. Hou, R. Z. Liu, T. Lu, and J. Yang. Shape robust text detection with progressive scale expansion network. In CVPR, 2019.

[19] H. Li, P. Xiong, J. An, and L. Wang. Pyramid attention network for semantic segmentation. In BMVC, 2018.

[20] M. Liao, B. Shi, and X. Bai. Textboxes++: A single-shot oriented scene text detector. IEEE Transactions on Image Processing, vol. 27, no. 8, 2018.

[21] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, Serge Belongie. Feature Pyramid Networks for Object Detection. arXiv preprint. arXiv: 1612.03144, 2017.

[22] M. Liao, B. Shi, X. Bai, X. Wang, and W. Liu. Textboxes: A fast text detector with a single deep neural network. In AAAI, 2017.

[23] M. H. Liao, Z. Zhu, B. G. Shi, G. S. Xia, X. Bai. Rotation-sensitive Regression for Oriented Scene Text Detection. In CVPR, 2018.

[24] C. Lin, J. Lu, G. Wang, and J. Zhou. Graininess-aware deep feature learning for pedestrian detection. In ECCV, 2018.

[25] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In ECCV, 2016.

[26] X.Liu, D.Liang, S.Yan, D.Chen, Y.Qiao, and J.Yan. Fots: Fast oriented text spotting with a unified network. In CVPR, 2018.

[27] Z. C. Liu, G. S. Lin, S. Yang, J. S. Feng, W. S. L, W. L. Goh. Learning Markov Clustering Networks for Scene Text Detection. In CVPR, 2018.

[28] Y. Liu and L. Jin. Deep matching prior network: Toward tighter multi-oriented text detection. In CVPR, 2017.

[29] Y. L. Liu, L. W. Jin, S. T. Zhang, and S. Zhang. Detecting curve text in the wild: New dataset and new solution. arXiv preprint arXiv:1712.02170, 2017.

[30] S. B. Long, J. Q. Ruan, W. J. Zhang, X. He, W. H. Wu, C. Yao. TextSnake: A Flexible Representation for Detecting Text of Arbitrary Shapes. In ECCV, 2018.

[31] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In CVPR, 2015.

[32] P. Y. Lyu, M. H. Liao, C. Yao, W. H. Wu, X. Bai. Mask TextSpotter: An End-to-End Trainable Neural Network for Spotting Text with Arbitrary Shapes. In ECCV, 2018.

[33] P. Y. Lyu, C. Yao, W. H. Wu, X. Bai. Multi-oriented scene text detection via corner localization and region segmentation. In CVPR, 2018.

[34] J. Ma, W. Shao, H. Ye, L. Wang, H. Wang, Y. Zheng, and X. Xue. Arbitrary-oriented scene text detection via rotation proposals. IEEE Transactions on Multimedia, 20(11):3111–3122, 2018.

[35] J. Mao, T. Xiao, Y. Jiang, and Z. Cao. What can help pedestrian detection? In CVPR, 2017.

[36] S. Mohanty, T. Dutta, and H. P. Gupta. Robust Scene Text Detection with Deep Feature Pyramid Network and CNN based NMS Model. In ICPR, 2018.

[37] N. Nayef, F. Yin, I. Bizid, H. Choi, Y. Feng, D. Karatzas, Z. Luo, U. Pal, C. Rigaud, J. Chazalon, et al. Icdar2017 robust reading challenge on multi-lingual scene text detection and script identification-rrc-mlt. In ICDAR, 2017.

[38] Y. Patel, M. Busta, and J. Matas. E2e-mlt-an unconstrained end-to-end method for multi-language scene text. arXiv preprint arXiv:1801.09919, 2018.

[39] V.-Q. Pham, S. Ito, and T. Kozakaya. Biseg: Simultaneous instance segmentation and semantic segmentation with fully convolutional networks. In BMVC, 2017.

[40] P. O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollar. Learning to refine object segments. In ECCV, 2016.

[41] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In CVPR, 2016.

[42] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In NIPS, 2015.

[43] B. Shi, X. Bai, and S. Belongie. Detecting Oriented Text in Natural Images by Linking Segments. In CVPR, 2017.

[44] K. Simonyan, K., Zisserman, A. Vedaldi. Very deep convolutional networks for large-scale image recognition. arXiv, 2014.

[45] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S.Reed, D. Anguelov, D. Erhan, et al. Going deeper with convolutions. In CVPR, 2015.

[46] J. Tang, Z. B. Yang, Y. P. Wang, Q. Zheng, Y. C. Xu, X. Bai. Detecting Dense and Arbitrary-shaped Scene Text by Instance-aware Component Grouping. Pattern Recognition, 2019.

[47] Zhi Tian, Weilin Huang, Tong He, Pan He, and Yu Qiao. Detecting text in natural image with connectionist text proposal network. In ECCV, 2016.

[48] W. h. Wang, E. Xie, X. G. Song, Y. H. Zang, W. J. Wang, T. Lu, G. Yu, and C. H. Shen. Efficient and Accurate Arbitrary-Shaped Text Detection with Pixel Aggregation Network. In ICCV, 2019.

[49] S. Woo, J. Park, J. Lee. CBAM: Convolutional Block Attention Module. In ECCV, 2018.

[50] E. Xie, Y. H. Zang, S. Shao, G. Yu, C. Yao, and G. Y. Li. Scene text detection with supervised pyramid context network. In AAAI, 2019.

[51] C. Xue, S. Lu, and F. Zhan. Accurate scene text detection through border semantics awareness and bootstrapping. In ECCV, 2018.

[52] C. Yao, X. Bai, W. Y. Liu, Y. Ma, and Z. W. Tu. Detecting texts of arbitrary orientations in natural images. In CVPR, 2012.

[53] C. Yao, X. Bai, N. Sang, X. Zhou, S. Zhou and Z. Cao. Scene text detection via holistic, multi-channel prediction. arXiv preprint arXiv:1606.09002, 2016.

[54] Q. Ye and D. Doermann. Text detection and recognition in imagery: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 37, no. 7, pp. 1480–1500, 2015.

[55] C. Zhang, B. Liang, Z. Huang, M. En, and et al. Look More Than Once: An Accurate Detector for Text of Arbitrary Shapes. In CVPR, 2019.

[56] Z. Zhang, C. Zhang, W. Shen, C. Yao, W. Liu and X. Bai. Multi-oriented text detection with fully convolutional networks. In CVPR, 2016.

[57] Z. Zhong, S. Huang. Deeptext: A new approach for text proposal generation and text detection in natural images. In ICASSP, 2017.

[58] Z. Zhong, L. Sun, and Q. Huo. An anchor-free region proposal network for faster r-cnn based text detection approaches. arXiv preprint arXiv:1804.09003, 2018.

[59] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, and J. Liang. East: An efficient and accurate scene text detector. In CVPR, 2017.

[60] Z. Zhu, M. D. Xu, S. Bai, T. T. Huang, and X. Bai. Asymmetric non-local neural networks for semantic segmentation. In ICCV, 2019.

[61] Y. Wu and P. Natarajan. Self-organized text detection with minimal post-processing via border learning. In ICCV, 2017.