Adaptive Boundary Proposal Network for Arbitrary Shape Text Detection

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

[1]School of Computer and Communication Engineering, University of Science and Technology Beijing

[2]USTB-EEasyTech Joint Lab of Artificial Intelligence, [3]Tencent Technology (Shenzhen) Co. Ltd

[4]Institute of Artificial Intelligence, University of Science and Technology Beijing

zhangshixue111@163.com, {zhuxiaobin, chunyang, xuchengyin}@ustb.edu.cn, hongfawang@tencent.com

Abstract

Arbitrary shape text detection is a challenging task due to the high complexity and variety of scene texts. In this work, we propose a novel adaptive boundary proposal network for arbitrary shape text detection, which can learn to directly produce accurate boundary for arbitrary shape text without any post-processing. Our method mainly consists of a boundary proposal model and an innovative adaptive boundary deformation model. The boundary proposal model constructed by multi-layer dilated convolutions is adopted to produce prior information (including classification map, distance field, and direction field) and coarse boundary proposals. The adaptive boundary deformation model is an encoder-decoder network, in which the encoder mainly consists of a Graph Convolutional Network (GCN) and a Recurrent Neural Network (RNN). It aims to perform boundary deformation in an iterative way for obtaining text instance shape guided by prior information from the boundary proposal model. In this way, our method can directly and efficiently generate accurate text boundaries without complex post-processing. Extensive experiments on publicly available datasets demonstrate the state-of-the-art performance of our method. Code is available at the website: https://github.com/GXYM/TextBPN.

1. Introduction

Scene text detection has been widely applied in various applications, such as online education, product search, and video scene parsing. Benefiting from the rapid development of deep learning, text detection methods [29, 11, 42, 22] have achieved impressive performance on images in which text instances are regular shape or aspect ratio. Recently, arbitrary shape text detection has attracted ever-increasing interests for it can well adapt to real applications.

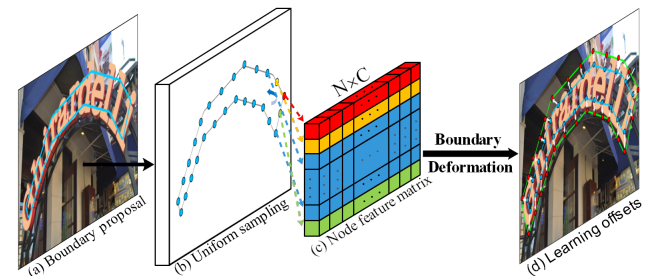Although arbitrary shape text detection methods [2, 12,

Figure 1. Illustration of the boundary proposal deformation process: (a) Boundary proposal; (b) Sampling on boundaries; (c) Extracting node feature matrix; (d) Learning offsets of sampling vertexes via adaptive boundary deformation model.

41] have achieved great improvements in recent years, there are still many issues to be addressed due to the challenging characteristic of scene texts, including varied shape, texture, scale, *etc*. Segmentation-based methods [37, 33, 12] have sparked a new wave of arbitrary shape text detection that locate text regions by a pixel-level prediction for enhancing the robustness to shape variations. However, there are still two main problems that remain to be explored.

One problem is that segmentation-based methods tend to fail in separating adjacent text instances in image. To solve this problem, existing methods [19, 34, 30, 40] usually shrink annotation boundaries as kernels (*e.g.*, text kernel [34], text center region [19]) to distinguish different text instances. For rebuilding a complete text, these methods usually need to merge the pixels in text regions to kernels by pre-defined expansion rules or auxiliary information (*e.g.*, similarity vector [34]). However, the merging process in [37, 33] are always performed by pixel-to-pixel, which is complex and inefficient. The other problem is that the final detected contours of texts in existing segmentation-based methods usually contain a lot of defects and noises. This because the performance of existing segmentation-based methods [4, 12, 33, 34] greatly relies on the accuracy of contour detection, neglecting adaptively adjusting detected contours. Different from generic object instances, text instances usually don't have closed contours and often contain

---
*Corresponding author.

a lot of background noisy pixels in coarse-grained boundary annotations. These will generate unpredictable results of pixels, especially those near boundaries, resulting in a lot of noises and defects in segmentation results.

To tackle the above-mentioned problems, we propose a novel adaptive boundary proposal network for arbitrary shape text detection, which can learn to directly produce accurate boundary for arbitrary shape text without any post-processing. Our adaptive boundary proposal network is mainly composed of a boundary proposal model and an adaptive boundary deformation model. The boundary proposal model is composed of multi-layer dilated convolutions, which will predict a classification map, a distance field, and a direction field based on shared convolutions. Inspired by RPN [25], we adopt the distance field and pixel classification map to generate coarse boundary proposals as shown in Fig. 1 (a). These coarse boundary proposals can roughly locate texts, and well separate adjacent texts because they are always slimmer than their boundary annotations in our method. To refine the coarse proposals, we adopt an innovative adaptive boundary deformation model to perform iterative boundary deformation for generating accurate text instance shape under the guidance of prior information (classification map, distance field and direction field). For fully excavating and exploiting topology and sequence context in each boundary proposal, the adaptive boundary deformation model adopt an encoder-decoder structure, in which the encoder mainly consists of a GCN and a RNN (B-LSTM). Notably, the proposed method is a unified end-to-end trainable framework with iterative optimization. Extensive experiments demonstrate that our method achieves state-of-the-art performance on several publicly available datasets.

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, which can directly generate accurate boundaries of arbitrary shape texts without any post-processing.

- We propose an adaptive boundary deformation model which can perform iterative boundary deformation for refining text boundary.

- Extensive experiments on public available datasets demonstrate the state-of-the-art performance of our method.

2. Related Work

Regression-Based Methods. Methods of this type rely on box-regression based object detection frameworks with word-level and line-level prior knowledge [22, 10, 13, 42, 40, 8]. Different from generic objects, texts are often presented in irregular shapes with various aspect ratios. To deal with this problem, RRPN [22] and Textboxes++ [10] localized text boxes by predicting the offsets from anchors. Different from these methods localizing text regions by implementing refinement on pre-defined anchors, EAST [42] and DDR [7] directly regressed the offsets from boundaries or vertexes to the current point for accurate and efficient text detection. Although regression-based methods have achieved good performance in quadrilateral text detection, they often can't well adapt to arbitrary shape text detection.

CC-Based Methods. The Connected Component (CC) based methods [39, 26, 5, 2, 41] usually detect individual text parts or characters firstly, followed by a link or group post-processing procedure for generating final texts. CRAFT [2] detected text regions by exploring affinities between characters. TextDragon [5] first detected the local region of the text, and then groups them by their geometric relations. Zhang *et al*. [41] used a graph convolution neural network (GCN) to learn and infer the linkage relationships of text components to group text components. Although CC-based methods have a more flexible representation that can adapt to irregular shape text well, the complex post-processing for grouping text components to complete texts is always time-consuming and unsatisfied.

Segmentation-Based Methods. Methods of this type [33, 37, 34, 12, 20] mainly draw inspiration from semantic segmentation methods and detect texts by estimating word bounding areas. To effectively distinguish adjacent text, PSENet [33] adopted a progressive scale algorithm to gradually expand the pre-defined kernels. In [34], Wang *et al*. proposed an efficient arbitrary shape text detector, named Pixel Aggregation Network (PAN), which is equipped with a low computational-cost segmentation head and learnable post-processing. DB [12] performed an adaptive binarization process in a segmentation network, which simplifies the post-processing and enhances the detection performance. However, the performances of these methods are strongly affected by the quality of segmentation accuracy.

Contour-based methods. In addition to the above methods, Contour-based methods also have attracted many researchers [21, 35, 32, 36]. Yao *et al*. [21] detected texts by predicting the corner of texts, and Lyu *et al*. [35] adopted a similar architecture as SSD [16] and rebuilt text with predicted corner points. Wang *et al*. [32] proposed an end-to-end approach toward arbitrary shape text spotting, which proposed a boundary point detection network to locate the text boundary in the text proposal region. Contour-Net [36] represented text region with a set of contour points, which adopted a Local Orthogonal Texture-aware Module (LOTM) to model the local texture information of proposal features in two orthogonal directions for generating contour points. However, compared with segmentation-based methods, the contour-based method has a big gap in detection performance and speed without recognition information.
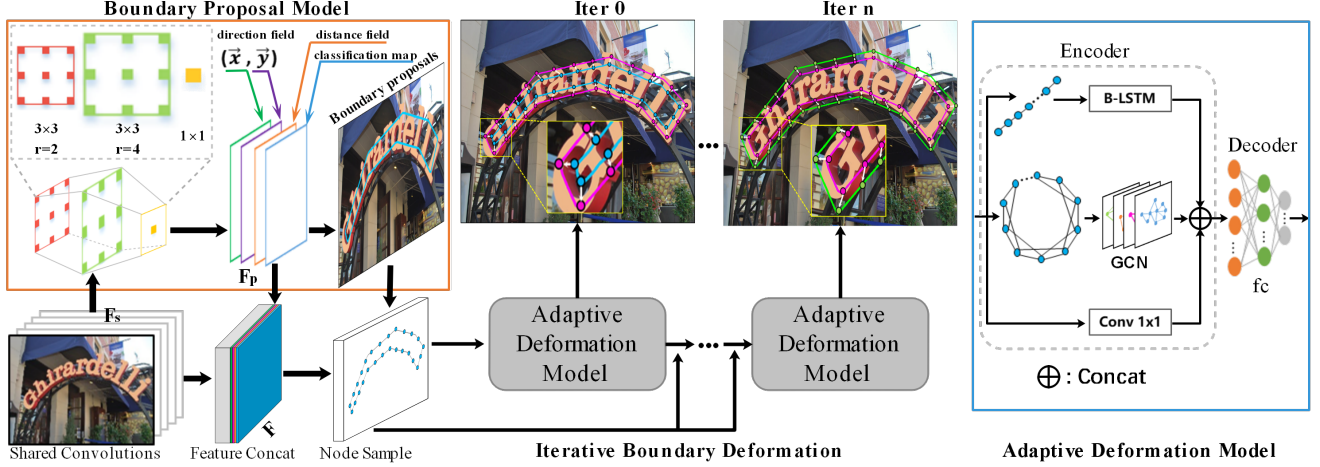
Figure 2. Framework of our method. Our network mainly consists of shared convolutions, boundary proposal model, and adaptive boundary deformation model, which is a unified end-to-end trainable framework with iterative optimization.

3. Proposed Method

3.1. Overview

The framework of our method is illustrated in Fig. 2. The ResNet-50 [6] is adopted to extract features. To preserve spatial resolution and take full advantage of multi-level information, we exploit a multi-level feature fusion strategy (similar to FPN [14]), as shown in Fig. 3. The boundary proposal model composed of multi-layer dilated convolutions uses the shared features for performing text pixels classification, generating the distance field and direction field [37]. Then, we use these information to produce coarse boundary proposals. Each boundary proposal consists of N points, representing a possible text instance. For refining the coarse proposals, an adaptive boundary deformation model is proposed to perform iterative boundary deformation for obtaining the more accurate text boundaries under the guidance of prior information (classification map, distance field, and direction field).

3.2. Adaptive Boundary Proposal Network

3.2.1 Boundary Proposal Generation

The boundary proposal model is composed of multi-layer dilated convolutions, include two 3×3 convolution layers with different dilation rates and one 1×1 convolution layers, as shown in Fig. 2. It will use the shared features extracted from the backbone network to generate classification map, distance field map, and direction field map.

Similar to other text detection methods [42, 19, 41], the classification map contains a classification confidence of each pixel (text/non-text). As in [37, 31], the direction field map (V) consists of a two-dimensional unit vector ($\vec{x}, \vec{y}$), which indicates the direction of each text pixel in boundary to its nearest pixel on boundary (pixel-to-boundary). For each pixel p inside a text instance T, we will find its near-
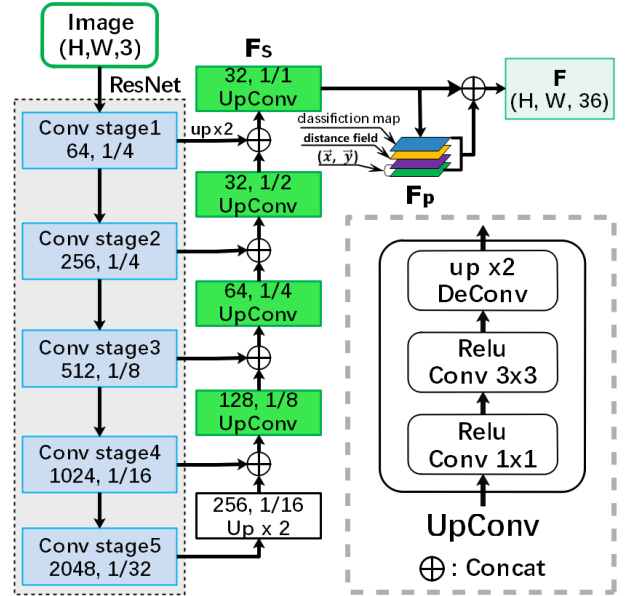

Figure 3. Architecture of shared convolutions, F_S denotes the shared features and F_p denotes the prior information (classification map, distance field, and direction field).

est pixel B_p on text boundary, as shown in Fig. 5. Then, a two-dimensional unit vector $\mathcal{V}_{gt}(p)$ that points away from the text pixel p to B_p can be formulated as

$$\mathcal{V}_{gt}(p) = \begin{cases} \overrightarrow{B_pp}/\left|\overrightarrow{B_pp}\right|, & p \in \mathbb{T} \\ (0,0), & p \notin \mathbb{T} \end{cases} \qquad (1)$$

where $\left|\overrightarrow{B_pp}\right|$ represents the distance between B_p and text pixel p, and $\mathbb{T}$ represent the total set of text instances in an image. For the non-text area ($p \notin \mathbb{T}$), we represent those pixels with $(0,0)$. The unit vector $\mathcal{V}_{gt}(p)$ not only directly encodes approximately relative location of p inside T and highlights the boundary between adjacent text instances [37], but also provides direction indication infor-
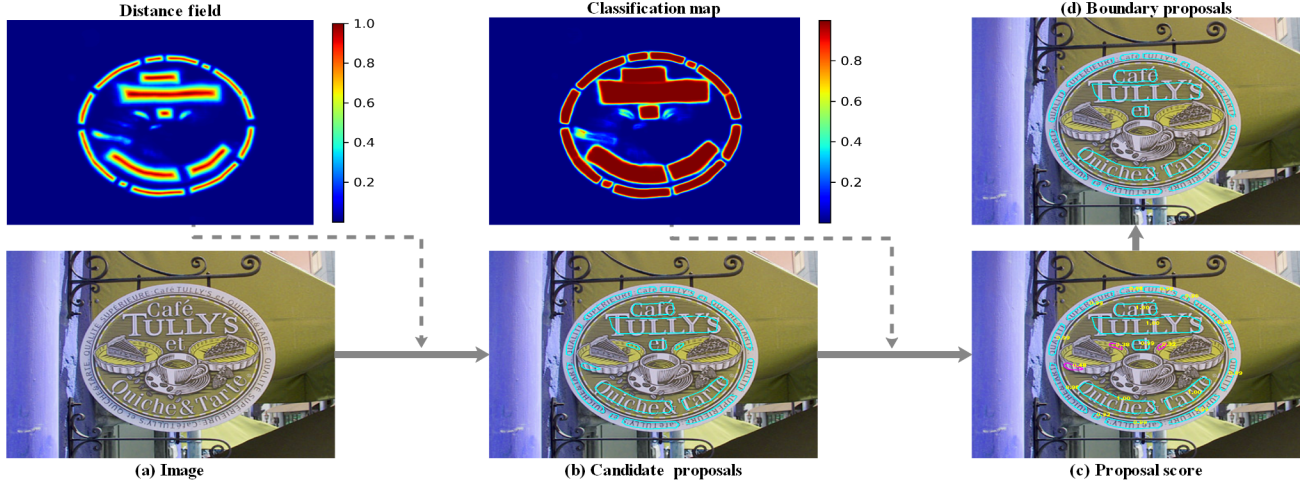
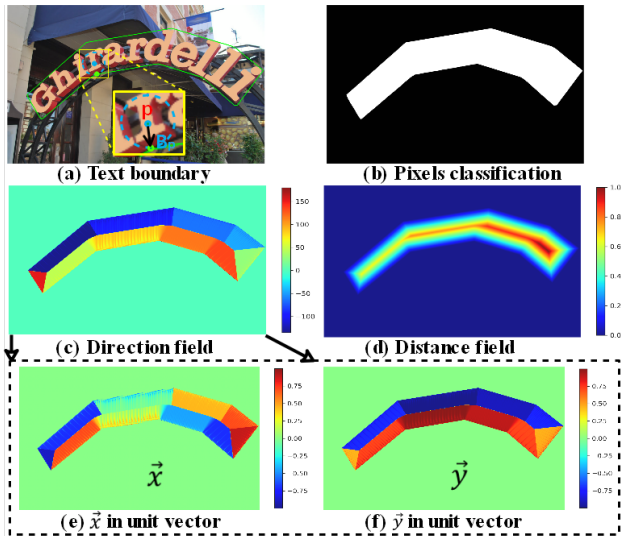Figure 4. Illustration of boundary proposal generation.

Figure 5. Illustration of ground truths for boundary proposal model, and the unit vector $(\vec{x}, \vec{y})$ in (e) and (f) is a vector representation of direction field in (c).

mation for boundary deformation.

For boundary deformation, the relative position distance information is as important as the direction information. In this work, the distance field map ($\mathcal{D}$) is a normalized distance map, and the normalized distance of the text pixel p to nearest pixel B_p on text boundary is defined as

$$\mathcal{D}_{gt}(p) = \begin{cases} \left| \overrightarrow{B_p p} \right| / L, & p \in \mathbb{T} \\ \\ 0, & p \notin \mathbb{T} \end{cases} \quad (2)$$

For the non-text area ($p \notin \mathbb{T}$), we represent the distance of those pixels with 0. L represents the scale of text instance T where the pixel p is located, and is defined as

$$L = max(D_{gt}(p)); \quad p \in T, \quad (3)$$

$\mathcal{D}_{gt}(p)$ not only directly encodes the relative distance of p inside T and further highlights the boundary between ad-

jacent text instances, but also provides a relative distance information for boundary deformation.

With the distance field map ($\mathcal{D}$), we can generate candidate boundary proposals by using a fixed threshold (th_d) to the predicted distance, as shown in Fig. 4 (b). However, these candidate boundary proposals inevitably contains false detections. Hence, we calculate the average confidence of each candidate boundary proposal according to the classification map for removing some of them with low confidence (th_s), as shown in Fig. 4 (c) and (d).

3.2.2 Adaptive Boundary Deformation

Inspired by interactive annotation of object segmentation methods [1, 15] and instance segmentation method [23], we perform arbitrary shape text detection by deforming a boundary proposal to a more accurate text boundary. Specifically, we take a boundary as input based on CNN features and predict per-vertex offsets pointing to the text boundary. In [1], the authors use a Recurrent Neural Network to model the sequence of 2D vertices of the polygon outlining an object. Afterwards, they propose a method [15] to treat the location of each control point as a continuous random variable, and learn to predict these variables via a Graph Neural Network that takes image evidence as input. Inspired by [1, 15], DeepSnake [23] performs object segmentation by deforming an initial contour to object boundary with circular convolution which consists of multi-layer 1-D convolutions with $1 \times N$ kernel size. However, these methods only consider individual topology context or sequence context, which isn't highly satisfactory for text detection. For the unique properties of texts, both topology context and sequence context are important in detection.

For each boundary proposal, we will uniformly sample N control points for facilitating batch processing. As shown in Fig. 2, the sampled control points form a closed poly-

gon in which topology context and sequence context are co-existing. To fully exploit the boundary topology and sequence context, we introduce an adaptive boundary deformation model combined with GCN and RNN, which can efficiently perform feature learning and iterative boundary deformation for refining coarse text boundaries.

Let $cp_i = [x_i, y_i]^T$ denote the location of the i-th control point, and $p = \{p_0, p_1, ..., p_{N-1}\}$ be the set of all control points. For a boundary proposal with N control points, we first construct feature vectors for each control point. The input feature f_i for a control point cp_i is a concatenation of 32-D shared features F_s obtained by CNN backbone and 4-D prior features F_p (*e.g.*, pixels classification, distance field and direction field). Therefore, the features of a control point are extracted from the corresponding location in F : $f_i = concat\{F_s(x_i, y_i), F_p(x_i, y_i)\}$. Here, $F_s(x_i, y_i)$ and $F_p(x_i, y_i)$ are computed by bilinear interpolation.

After obtaining the feature matrix X (size: $N \times C$) of boundary proposal, we adopt the adaptive deformation model based on an encoder-decoder architecture to perform efficient feature learning and iterative boundary deformation. The encoder model is combined with GCN and RNN for feature learning, which can fully exploit and fuse the boundary topology and sequence context. As shown in Fig. 2, the encoder model can be formulated as

$$X^{'} = RNN(X) \oplus GCN(X) \oplus Conv_{1 \times 1}(X) \quad (4)$$

where "$\oplus$" refers to the concatenation operation; RNN is consisted of one-layer B-LTSM with 128 hidden size; $Conv_{1 \times 1}$ consists of one-layer 1×1 convolution layers with 128 dimensions, which form a residual connection like RestNet [6]; GCN is consisted of four graph convolution layers activated by ReLU, and the graph convolution layer in our method is formulated as

$$\mathbf{X}_g = ReLU((\mathbf{X} \oplus \mathbf{GX})\mathbf{W}), \quad (5)$$
$$\mathbf{G} = \tilde{\mathbf{D}}^{-1/2} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-1/2}, \quad (6)$$

where $\mathbf{X} \in \Re^{N \times d_i}, \mathbf{X}_g \in \Re^{N \times d_o}$, d_i/d_o is the dimension of in/out features, and N is the number of control points; $\mathbf{G}$ is a symmetric normalized Laplacian of size $N \times N$; W is a layer-specific trainable weight matrix; $\tilde{A} = A + I_N$ 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 $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$. We form A by connecting each control point in p with its four neighbors.

The decoder in adaptive deformation model consists of three-layer 1×1 convolutions with ReLU, which will learn to predict offsets between control points and the target points. To obtain the more accurate text boundary, we perform iterative boundary deformation, as shown in Fig. 2.

3.3. Optimization

In this work, the total loss $\mathcal{L}$ can be formulated as

$$\mathcal{L} = \mathcal{L}_{Bp} + \frac{\lambda * \mathcal{L}_{Bd}}{1 + e^{(i-eps)/eps}}, \quad (7)$$

where $\mathcal{L}_{Bp}$ is a loss for the boundary proposal model, and $\mathcal{L}_{Bd}$ is a loss for the adaptive boundary deformation model; eps denotes the maximum epoch of training, and i denote the i-th epoch in train. In our experiments, λ is set to 0.1. In Eq. 7, $\mathcal{L}_{Bp}$ is computed as

$$\mathcal{L}_{Bp} = \mathcal{L}_{cls} + \alpha * \mathcal{L}_{\mathcal{D}} + \mathcal{L}_{\mathcal{V}}, \quad (8)$$

where $\mathcal{L}_{cls}$ is a cross-entropy classification loss for pixels classification, and $\mathcal{L}_{\mathcal{D}}$ is a L_2 regression loss for distance field. OHEM [27] is adopted for $\mathcal{L}_{cls}$ and $\mathcal{L}_{\mathcal{D}}$ in which the ratio between the negatives and positives is set to 3:1. To balance the losses in $\mathcal{L}_{Bp}$, the weights α is set to 3.0. Similar to [31], $\mathcal{L}_{\mathcal{V}}$ consists of L_2-norm distance and angle distance for direction field $\mathcal{V}$:

$$\mathcal{L}_{\mathcal{V}} = \sum_{p \in \Omega} w(p)\|\mathcal{V}_p - \hat{\mathcal{V}}_p\|_2 + \frac{1}{\mathbb{T}} \sum_{p \in \mathbb{T}} (1 - \cos(\mathcal{V}_p, \hat{\mathcal{V}}_p)), \quad (9)$$

where Ω represents image domain; the weight $(w(p) = 1/\sqrt{|GT_p|})$ at pixel p is proportional to the inverse square root of the size of ground truth segment GT_p containing p.

$\mathcal{L}_{Bd}$ is a point matching loss similar to [15]. In this work, the prediction and ground truth control point sets have equal sized and similar order (counter-clockwise), denoted as $\mathbf{p} = \{p_0, p_1, \cdots, p_{N-1}\}$, and $\mathbf{p}' = \{p'_0, p'_1, \cdots, p'_{N-1}\}$ (N is the number of points), respectively. Hence, the matching loss for $\mathbf{p}$ and $\mathbf{p}'$ is defined as

$$\mathcal{L}_{(\mathbf{p}, \mathbf{p}')} = \min_{j \in [0 \cdots, N-1]} \sum_{i=0}^{N-1} smooth_{L1}(p_i, p'_{(j+i)\%N}), \quad (10)$$

Because there is usually more than one text instance in an image, $\mathcal{L}_{Bd}$ is defined as

$$\mathcal{L}_{Bd} = \frac{1}{\mathbb{T}} \sum_{p \in \mathbb{T}} \mathcal{L}_{(\mathbf{p}, \mathbf{p}')}, \quad (11)$$

where $\mathbb{T}$ represents all the text instances in an image, $\mathbf{p}$ represents the control point set for text instance T ($T \in \mathbb{T}$).

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 with polygon and word-level annotations.
CTW-1500: It consists of $1,000$ training and 500 testing images, and curved text instances are annotated by polygons with 14 vertices.
MSRA-TD500: It consists of 500 training and 200 testing images, including English and Chinese texts which contain multi-lingual long texts with multi-orientations.
SynthText: It contains 800k synthetic images generated by blending natural images with artificial text which are all word-level annotated .
ICDAR2017-MLT: It consists of $7,200$ training images, $1,800$ validation images, and $9,000$ test images with multi-lingual (9 languages) texts annotated by quadrangle.

Table 1. Ablation experiments for deformation model on Total-Text and CTW-1500. The best score is highlighted in **bold**.

Methods	Total-Text				CTW-1500			
	Recall	Precision	F-measure	FPS	Recall	Precision	F-measure	FPS
FC	81.56	90.16	85.65	9.52	78.32	85.03	81.54	11.13
RNN	**83.31**	87.71	85.93	11.15	**81.26**	86.00	83.56	12.22
Circular convolution	82.80	89.73	86.13	9.33	80.35	84.88	82.55	10.89
Graph convolution	82.74	89.94	86.19	10.42	80.31	86.12	83.12	11.94
Adaptive deformation	83.30	**90.76**	**86.87**	10.56	80.57	**87.66**	**83.97**	12.08

4.2. Implementation Details

In our experiments, we first pre-train our network on SynthText by one epochs, in which images are randomly cropped and resized to 512×512. In pre-training, the Adam [9] optimizer is applied with a fixed learning rate of 0.001, and a mini-batch is set to 16. In fine-tuning, we randomly crop the text region, and resize them to 640×640 for training the model with the mini-batch 12. The Adam [9] optimizer is adopted, in which the initial learning rate is 0.001 and decreased to 0.9 of the original after each 50 epochs. The data augmentation includes: random rotation with an angle (sampled by Gaussian distribution in $(-60°, 60°)$), random cropping, and random flipping. In inference, we keep the aspect ratio of test images, then resize and pad them into the same size for testing. The code is implemented with PyTorch 1.7 and python 3. Training is performed on single GPU (RTX-3090), and testing is performed on single GPU (GeForce RTX-2080) with Intel Xeon Silver 4108 CPU @ 1.80GHz.

4.3. Ablation Study

In ablation experiments, we only train the model on corresponding real-world datasets for 660 epochs without pre-training, and the other training settings are identical with the fine-tuning process in Sec 4.2. In testing, the short side of an image is scaled to 640, and ensure that the long side doesn't exceed 1,024. The threshold th_d and th_s are set to 0.3 and 0.8, respectively.

Effectiveness of adaptive deformation model. To verify the effectiveness of the adaptive deformation model, we conduct ablation experiments on Total-Text and CTW-1500. Our deformation model consists of an encoder and a decoder. For fair comparison, we use a lightweight full connection network (FC) structure as decoder, and we adopt four types of encoder, *i.e.*, FC with $Conv_{1\times1}$, RNN, circular convolution, and graph convolution (GCN), for conducting comparative experiments. As listed in Tab. 1, our adaptive deformation model achieves the best performance compared with the other four methods on both Total-Text and CTW-1500, which achieves improvements by 0.94% in terms of F-measure on Total-Text compared with RNN, and by 0.85% in terms of F-measure on Total-Text compared with GCN. Moreover, our adaptive deformation model doesn't bring obvious more consume of detection time.
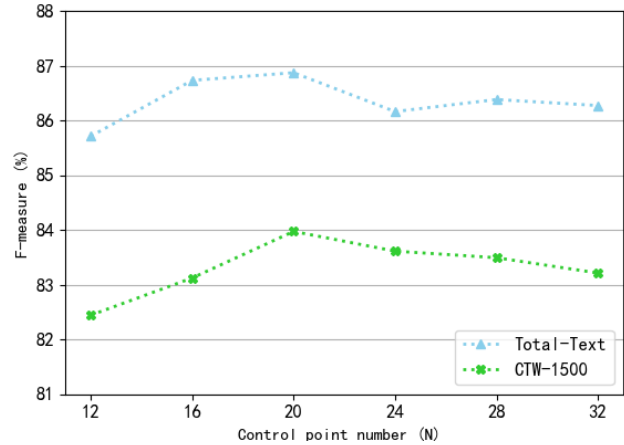

Figure 6. Experimental results of control point number (N).

Influence of control point number (N). We study the influence of the control point number (N) by setting the number of control point number from 12 to 32 with an interval of 4. The models are evaluated on Total-Text and CTW-1500. From Fig. 6, we can find that the F-measures drops when N is too large or too small. Especially, too few control points will make great degradation of performance. This is because the detection boundary often can't correctly cover the whole text when the control number is too small. Apparently, our model achieves the best detection performance on both two datasets while the number of control points is around 20. Therefore, in our experiments, the number of control points is fixed at 20.

Table 2. Experimental results of different iterations on CTW-1500.

	Iter. 1	Iter. 2	Iter. 3
F-measure	82.24	83.33	**83.97**
Speed (fps)	**13.68**	12.83	12.08

Influence of iteration number. To fully validate the influence of iteration number, we further compare models with different inference iterations. As listed in Tab. 2, with the increase of the number of iterations, the detection performance is gradually improved but the inference speed of our model is gradually dropped. When the number of iterations is from 2 to 3, the increase of detection performance is not very obvious. Considering the balance of speed and performance, the number of iterations is set to 3 by default in our experiments. As shown in Fig. 7, the detection boundaries become more accurate along with the increase of iterations.

| (a) boundary proposals | (b) iter 1 | (c) iter 2 | (d) iter 3 |

Figure 7. Visual results of different iterations. The blue contours are boundary proposals, and the green contours are detection boundaries after iterative deformation.

Table 3. Ablation study for classification map (cls), distance field (dis), and direction field (dir) on Total-Text.

cls	dis	dir	Recall	Precision	F-measure
✓	×	×	76.96	83.01	79.87
✓	✓	×	81.97	88.95	85.32
✓	✓	✓	**83.30**	**90.76**	**86.87**

Influence of prior information. We conduct ablation studies on Total-Text to verify the importance of each prior information (*e.g.*, classification map, distance field and direction field). As listed in Tab. 3, the detection performance is unsatisfactory when only use classification map. Once distance field and direction field are introduced, the performance is improved significantly, and F-measure is improved by 5.45% and 1.55%, respectively.

Table 4. Experimental results on Total-Text for different resolution FPN. "R", "P", and "F" represent Recall, Precision, and F-measure, respectively.

Methods	R	P	F	FPS
FPN-P1 (1/1)	**83.30**	**90.76**	**86.87**	10.56
FPN-P1 (1/2)	82.63	90.75	86.50	12.68
FPN-P2 (1/4)	82.99	89.51	86.13	**15.17**

Influence of different resolution FPN. We have conducted experiments without any pre-training to explore the influence of using the different resolution FPN-layer as the shared features. In Tab. 4, FPN-P2 (1/4) means that we use the FPN-P2 layer as the shared features (the resolution of FPN-P2 is $1/4$ of the original document image). From Tab. 4, we can see that FPN-P1 and FPN-P2 both achieve the state-of-the-art performance on Total-Text.

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

We compare our method with the previous state-of-the-art methods on Total-Text, CTW-1500 and MSRA-TD500. In testing, the short side of the images is scaled to 640, and

Table 5. Experimental results on Total-Text. 'Ext' denotes that the extra pre-training data is used. 'Syn'/ 'MLT' means SynthText or ICDAR2017-MLT dataset are used for pre-training, and 'MLT+' means there are other pre-training data besides MLT.

Methods	Ext	R	P	F	FPS
TextSnake [19]	Syn	74.5	82.7	78.4	-
ATTR [35]	-	76.2	80.9	78.5	10.0
MSR [38]	Syn	85.2	73.0	78.6	4.3
CSE [18]	MLT	79.7	81.4	80.2	0.42
TextDragon [5]	MLT+	75.7	85.6	80.3	-
TextField [37]	Syn	79.9	81.2	80.6	6.0
PSENet-1s [33]	MLT	77.96	84.02	80.87	3.9
SegLink++ [28]	Syn	80.9	82.1	81.5	-
LOMO [40]	Syn	79.3	87.6	83.3	4.4
CRAFT [2]	Syn	79.9	87.6	83.6	-
DB [12]	Syn	82.5	87.1	84.7	32.0
PAN [34]	Syn	81.0	89.3	85.0	**39.6**
TextPerception[24]	Syn	81.8	88.8	85.2	-
ContourNet [36]	-	83.9	86.9	85.4	3.8
DRRG [41]	MLT	84.93	86.54	85.73	-
Boundary [32]	Syn	85.0	88.9	87.0	-
Ours	-	83.30	**90.76**	86.87	10.56
Ours	Syn	84.65	90.27	87.37	10.28
Ours	MLT	**85.19**	90.67	**87.85**	10.69

the long side will be suppressed in 1,024. The threshold th_d for obtaining the candidate boundary proposals with distance filed is fixed to 0.3.

Total-Text. In testing, the threshold th_s is set to 0.825. The quantitative results are listed in Tab. 5. From Tab. 5, we can find that our method achieves 87.37% in terms of F-measure when pre-trained on SynthText, and 87.85% in the item of F-measure when pre-trained on MLT17. Obviously, our method significantly outperform other methods with a great margin. From the visible results in Fig. 8 (a) and Fig. 8 (b), we can observe that our method can precisely detect word-level irregular texts.

| (a) Total-Text | (b) Total-Text | (c) CTW-1500 | (d) CTW-1500 |

Figure 8. Visual experimental results. The blue contours are boundary proposals, and the green contours are final detection boundaries.

Table 6. Experimental results on CTW-1500.

Methods	Ext	R	P	F	FPS
TextSnake [19]	Syn	**85.3**	67.9	75.6	-
CSE [18]	MLT	76.1	78.7	77.4	0.38
LOMO[40]	Syn	76.5	85.7	80.8	4.4
ATRR[35]	Sy-	80.2	80.1	80.1	-
SegLink++ [28]	Syn	79.8	82.8	81.3	-
TextField [37]	Syn	79.8	83.0	81.4	6.0
MSR[38]	Syn	79.0	84.1	81.5	4.3
PSENet-1s [33]	MLT	79.7	84.8	82.2	3.9
DB [12]	Syn	80.2	86.9	83.4	22.0
CRAFT [2]	Syn	81.1	86.0	83.5	-
TextDragon [5]	MLT+	82.8	84.5	83.6	-
PAN [34]	Syn	81.2	86.4	83.7	**39.8**
ContourNet [36]	-	84.1	83.7	83.9	4.5
DRRG [41]	MLT	83.02	85.93	84.45	-
TextPerception[24]	Syn	81.9	87.5	84.6	-
Ours	-	80.57	87.66	83.97	12.08
Ours	Syn	81.45	**87.81**	84.51	12.15
Ours	MLT	83.60	86.45	**85.00**	12.21

Table 7. Experimental results on MSRA-TD500.

Methods	R	P	F	FPS
SegLink [26]	70.0	86.0	77.0	8.9
PixelLink [4]	73.2	83.0	77.8	-
TextSnake [19]	73.9	83.2	78.3	1.1
TextField [37]	75.9	87.4	81.3	5.2
MSR[38]	76.7	87.4	81.7	-
FTSN [3]	77.1	87.6	82.0	-
LSE[30]	81.7	84.2	82.9	-
CRAFT [2]	78.2	88.2	82.9	8.6
MCN [17]	79	88	83	-
ATRR[35]	82.1	85.2	83.6	-
PAN [34]	83.8	84.4	84.1	30.2
DB[12]	79.2	**91.5**	84.9	**32.0**
DRRG [41]	82.30	88.05	85.08	-
Ours (Syn)	80.68	85.40	82.97	12.68
Ours (MLT)	**84.54**	86.62	**85.57**	12.31

CTW-1500. In testing, the threshold th_s is set to 0.8. Representative visible results are shown in Fig. 8 (c) and (d), which indicate our method precisely detects boundaries of long curved text with line-level. The quantitative results are listed in Tab. 6. Compared with the previous sate-of-the-art methods [12, 34, 36], our approach achieves promising performance of both precision (87.81%) and F-measure (85.0%). Specifically, our method greatly outperforms TextSnake [19] and DB [12] on CTW-1500 in item of F-measure by 9.4% and 1.6%, respectively.

MSRA-TD500. In testing, the threshold th_s is set to 0.925. The quantitative comparisons with other methods on this dataset is listed in Tab. 7. From Tab. 7, we can conclude that our method successfully detects long text lines of arbitrary orientations and sizes. Notably, our method achieves 85.57% in terms of F-measure, which outperforms other sate-of-the-art methods, such as DB [12], DRRG [41], *etc*.

5. Conclusion

In this paper, we propose a novel adaptive boundary proposal network for arbitrary shape text detection, which adopt an boundary proposal model to generate coarse boundary proposals, and then adopt an adaptive boundary deformation model combined with GCN and RNN to perform iterative boundary deformation to obtain the more accurate text instance shape. Extensive experiments show that the proposed method can precisely detects the boundaries of arbitrary shape text in challenging datasets. In future study, we are interested in developing a real time method for text of arbitrary shapes based on current work.

References

[1] David Acuna, Huan Ling, Amlan Kar, and Sanja Fidler. Efficient interactive annotation of segmentation datasets with polygon-rnn++. In *CVPR*, pages 859–868, 2018.

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

[3] Yuchen Dai, Zheng Huang, Yuting Gao, Youxuan Xu, Kai Chen, Jie Guo, and Weidong Qiu. Fused text segmentation networks for multi-oriented scene text detection. In *ICPR*, pages 3604–3609, 2018.

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

[5] 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.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[7] Wenhao He, Xu-Yao Zhang, Fei Yin, and Cheng-Lin Liu. Deep direct regression for multi-oriented scene text detection. In *ICCV*, pages 745–753, 2017.

[8] Jie-Bo Hou, Xiaobin Zhu, Chang Liu, Kekai Sheng, LongHuang Wu, Hongfa Wang, and Xu-Cheng Yin. HAM: hidden anchor mechanism for scene text detection. *IEEE Trans. Image Process.*, 29:7904–7916, 2020.

[9] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[10] Minghui Liao, Baoguang Shi, and Xiang Bai. Textboxes++: A single-shot oriented scene text detector. *IEEE Trans.Image Processing*, 27(8):3676–3690, 2018.

[11] Minghui Liao, Baoguang Shi, Xiang Bai, Xinggang Wang, and Wenyu Liu. Textboxes: A fast text detector with a single deep neural network. In *AAAI*, pages 4161–4167, 2017.

[12] Minghui Liao, Zhaoyi Wan, Cong Yao, Kai Chen, and Xiang Bai. Real-time scene text detection with differentiable binarization. In *AAAI*, pages 11474–11481, 2020.

[13] 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.

[14] 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.

[15] Huan Ling, Jun Gao, Amlan Kar, Wenzheng Chen, and Sanja Fidler. Fast interactive object annotation with curve-gcn. In *CVPR*, pages 5257–5266, 2019.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] Pengyuan Lyu, Minghui Liao, Cong Yao, Wenhao Wu, and Xiang Bai. Mask TextSpotter: An end-to-end trainable neural network for spotting text with arbitrary shapes. In *ECCV*, pages 71–88, 2018.

[21] 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.

[22] 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.

[23] Sida Peng, Wen Jiang, Huaijin Pi, Xiuli Li, Hujun Bao, and Xiaowei Zhou. Deep snake for real-time instance segmentation. In *CVPR*, pages 8530–8539. IEEE, 2020.

[24] Liang Qiao, Sanli Tang, Zhanzhan Cheng, Yunlu Xu, Yi Niu, Shiliang Pu, and Fei Wu. Text perceptron: Towards end-to-end arbitrary-shaped text spotting. In *AAAI*, pages 11899–11907, 2020.

[25] 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.

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

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

[28] 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.

[29] 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.

[30] 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.

[31] Jianqiang Wan, Yang Liu, Donglai Wei, Xiang Bai, and Yongchao Xu. Super-bpd: Super boundary-to-pixel direction for fast image segmentation. In *CVPR*, pages 9250–9259. IEEE, 2020.

[32] Hao Wang, Pu Lu, Hui Zhang, Mingkun Yang, Xiang Bai, Yongchao Xu, Mengchao He, Yongpan Wang, and Wenyu Liu. All you need is boundary: Toward arbitrary-shaped text spotting. In *AAAI*, pages 12160–12167, 2020.

[33] 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.

[34] 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.

[35] 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.

[36] Yuxin Wang, Hongtao Xie, Zheng-Jun Zha, Mengting Xing, Zilong Fu, and Yongdong Zhang. Contournet: Taking a further step toward accurate arbitrary-shaped scene text detection. In *CVPR*, pages 11753–11762, 2020.

[37] 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.

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

[39] 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.

[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] Shi-Xue Zhang, Xiaobin Zhu, Jie-Bo Hou, Chang Liu, Chun Yang, Hongfa Wang, and Xu-Cheng Yin. Deep relational reasoning graph network for arbitrary shape text detection. In *CVPR*, pages 9699–9708, 2020.

[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.