All You Need Is a Second Look: Towards Arbitrary-Shaped Text Detection

Meng Cao, Can Zhang, Dongming Yang, *Member, IEEE*, and Yuexian Zou, *Senior Member, IEEE*

***Abstract*— Arbitrary-shaped text detection is a challenging task since curved texts in the wild are of the complex geometric layouts. Existing mainstream methods follow the instance segmentation pipeline to obtain the text regions. However, arbitrary-shaped texts are difficult to be depicted through one single segmentation network because of the varying scales. In this paper, we propose a two-stage segmentation-based detector, termed as NASK (Need A Second looK), for arbitrary-shaped text detection. Compared to the traditional single-stage segmentation network, our NASK conducts the detection in a coarse-to-fine manner with the first stage segmentation spotting the rectangle text proposals and the second one retrieving compact representations. Specifically, NASK is composed of a Text Instance Segmentation (TIS) network (1^{st} stage), a Geometry-aware Text RoI Alignment (GeoAlign) module, and a Fiducial pOint eXpression (FOX) module (2^{nd} stage). Firstly, TIS extracts the augmented features with a novel Group Spatial and Channel Attention (GSCA) module and conducts instance segmentation to obtain rectangle proposals. Then, GeoAlign converts these rectangles into the fixed size and encodes RoI-wise feature representations. Finally, FOX disintegrates the text instance into serval pivotal geometrical attributes to refine the detection results. Extensive experimental results on four public benchmarks including Total-Text, SCUT-CTW1500, ICDAR 2015 and ICDAR 2017 MLT verify that our NASK outperforms recent state-of-the-art methods.**

***Index Terms*— Arbitrary-shaped text detection, two-stage segmentation, self-attention, text geometric modeling.**

I. Introduction

SCENE text detection (STD) aims to accurately localize text regions given a natural scene image, and has attracted a surge of attention in the computer vision community due to its practical applications. However, despite the significant achievements in multi-oriented text detection, arbitrary-shaped text detection still remains challenging due to the complex geometric layouts.

Multi-oriented text instances take the rectangle or quadrilateral bounding-box to represent the detection results [1]–[3]. These simple representations, however, fall short when dealing with the more laborious arbitrary-shaped texts. Therefore, several segmentation-based methods [4]–[6] have been proposed to deal with such challenging yet universal scenario. Most of the current overwhelming majority of arbitrary-shaped text detection methods can roughly be classified into two categories: top-down, global modeling methods [6], [7] and bottom-up, local modeling methods [4], [5]. Typically, global modeling methods treat texts as a special type of object and directly take the regression-based methodology to obtain the detection results. For simplification, they often presuppose the distribution of the text instances (*e.g.* Bezier assumption for ABCNet [6] and Chebyshev polynomial approximation for TextRay [7]), and reconstruct texts with the regressed key points. Although these methods achieve faster inference speed, their predefined distribution hypotheses are usually empirical without universality, leading to the inferior performance. On the contrast, the bottom-up methods [4], [5] are more well-defined, *i.e.,* they use several crucial geometry attributes to rebuild the whole text instances. For example, the pioneering work TextSnake [5] represents text instances with a set of serially-connected disks and achieves competitive detection results. However, it suffers from two limitations.

Firstly, TextSnake applies the geometric attribute segmentation network on the input images directly, which makes it less resistant to noise [7]. In some cases where the text instances are extremely tiny, the geometric attribute prediction becomes more difficult because even the trivial segmentation deviation may lead to the ultimate failure. Therefore, a single segmentation network fails to process text instances that vary greatly in scales.

Secondly, the text geometric modeling in TextSnake is not optimal. As illustrated in Fig. 1(b), the control unit of each overlapping disk includes the text center line, disk radius, and the text line orientation. Implicitly, TextSnake regards the character direction to be perpendicular to the text direction. This assumption, however, is often too restrictive and may lead to failure detection especially when encountering distorted texts.

To address those above issues, we propose a two-stage segmentation-based network termed as NASK for accurate arbitrary-shaped text detection. NASK is short for **N**eed **A**

Manuscript received November 26, 2020; revised March 2, 2021; accepted March 17, 2021. Date of publication March 23, 2021; date of current version February 4, 2022. This work was supported in part by the PKU-HKUST Industry, Education and Research Institution (IER) Foundation under Grant HT-JD-CXY-201904 and in part by the Shenzhen Municipal Development and Reform Commission (Disciplinary Development Program for Data Science and Intelligent Computing). This article was recommended by Associate Editor Z. Wang. *(Corresponding author: Yuexian Zou.)*

Meng Cao, Can Zhang, and Dongming Yang are with the School of Electrical and Computer Engineering, Peking University, Shenzhen 518055, China, and also with the Shenzhen Graduate School, Peking University, Shenzhen 518055, China (e-mail: mengcao@pku.edu.cn; zhangcan@pku.edu.cn; yangdongming@pku.edu.cn).

Yuexian Zou is with the School of Electrical and Computer Engineering, Peking University, Shenzhen 518055, China, also with the Shenzhen Graduate School, Peking University, Shenzhen 518055, China, and also with the Peng Cheng Laboratory, Shenzhen 518066, China (e-mail: zouyx@pku.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCSVT.2021.3068133.

Digital Object Identifier 10.1109/TCSVT.2021.3068133

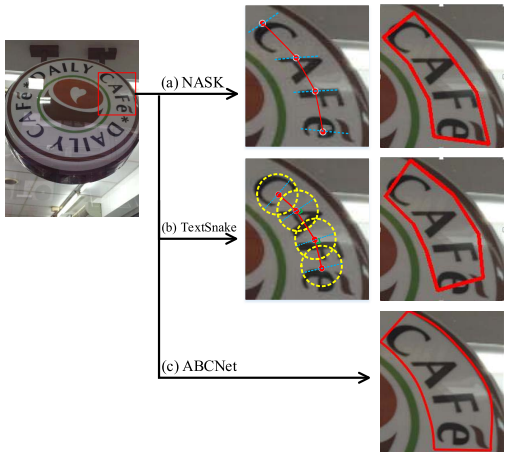


Fig. 1. Typical curved text instance representations. (a) Our proposed NASK; (b) TextSnake; (c) ABCNet. NASK achieves a more compact and accurate representation than the others. TextSnake suffers from boundary character cut off while ABCNet generates a loose result with more background.

Second loo**K**, indicating that our network is a two-stage approach with the coarse-to-fine detection. NASK consists of a Text Instance Segmentation network (TIS), a Geometry-aware Text RoI Alignment module (GeoAlign) and a Fiducial pOint eXpression module (FOX). The benefits of using the two cascaded segmentation networks are two-folds: 1) With the first stage segmentation to obtain the rectangle text proposals, the second stage FOX utilizes Region of Interest (RoI) features to predict the basic geometry attributes. Compared to applying FOX on the whole input images, it reduces the background interference greatly. 2) We utilize GeoAlign to transform varying-size text instances to the fixed-size feature maps before feeding them to the second stage network FOX. In this way, we save the FOX network from suffering from varying-size text input. Namely, text instances with varying shapes in input images are represented by fixed size feature maps, which eases the network training.

Specifically, the first stage segmentation network TIS is designed to localize rectangular text instances with the proposed Group Spatial and Channel Attention module (GSCA). Compared to the traditional Non-local neural network [8], GSCA takes one step further to model long-range dependencies more extensively by computing interactions between any two positions across both space and channels. Then the GeoAlign module transforms the varying-size rectangular RoIs into a fixed size. Compared to the traditional RoI Pooling [9] and RoI alignment [10], our GeoAlign adaptively selects the sampling points and avoids the background interference. Finally, FOX is a novel arbitrary-shaped text representation based on a set of fiducial points which are calculated with several geometry attributes. As shown in Fig. 1, in comparison to state-of-the-art methods [5], [6], our NASK achieves tighter and more flexible results, thus more suitable for the arbitrary-shaped text reconstruction.

In a nutshell, the main contributions of this work are as follows: (1) A novel attention module termed as GSCA is proposed to explore both the spatial-wise and channel-wise correlations in a more extensive way for more informative feature refinements. (2) We propose a more reasonable representation called Fiducial pOint eXpression module (FOX) tailored for arbitrary-shaped text instances. (3) We introduce a geometry-aware sampling method, a.k.a GeoAlign, for accurate RoI feature alignment. (4) Based on the novel two-stage segmentation architecture, our detector NASK achieves state-of-the-art performance on both curved and multi-oriented text detection benchmarks.

The preliminary work has been published in ICASSP 2020 [11] and we have extended it in the following significant aspects:

- We improve the previous Text RoI Pooling module [11] to the Geometry-aware RoI alignment module (GeoAlign). Experiment results show that GeoAlign leads to more accurate detection results.
- We revisit our Group Spatial and Channel Attention Module and redesign the Global Channel Attention branch with a squeeze-and-excitation module [12]. Compared to the previous version, it brings about better performance with negligible overheads.
- Besides the curved text datasets, more experiments are conducted on the multi-oriented scene text dataset, which demonstrates that our proposed NASK is a more general text detector with state-of-the-art performance.

II. Related Work

A. Scene Text Detection

Based on deep neural networks, scene text detection methods have progressed extensively. These methods are roughly classified into two categories.

1) Detection-Based Methods: Scene texts are detected using the adapted one-stage or two-stage frameworks which have been proved effective in general object detection tasks. TextBoxes [1] inherits the architecture of SSD [13] and makes some adaptive modifications to achieve both high accuracy and efficiency in a single network forward pass. TextBoxes++ [2] extends TextBoxes to handle multi-oriented texts and refines end-to-end text recognition combined with a text recognizer. RRPN [14] proposes a Rotation Region Proposal Network which generates inclined proposals with text orientation information to facilitate the multi-oriented text detection. EAST [15] is another one-stage detector which directly predicts text instances with arbitrary orientations and quadrilateral shapes in full images.

2) Segmentation-Based Methods: Segmentation-based methods draw inspiration from instance segmentation and conduct dense predictions in the pixel level [16]–[20]. Zhang *et al.* [16] detect multi-oriented scene text using Fully Convolutional Network (FCN) to predict the salient map of text regions in a holistic manner. Lyu *et al.* [18] propose to detect scene texts by localizing corner points of text bounding boxes and segmenting text regions in relative positions. PSENet [19] applies different scales of kernels for each text instance and generates the corresponding scale segmentation maps. Based on Mask R-CNN, SPCNet [20] proposes a supervised pyramid context network to suppress

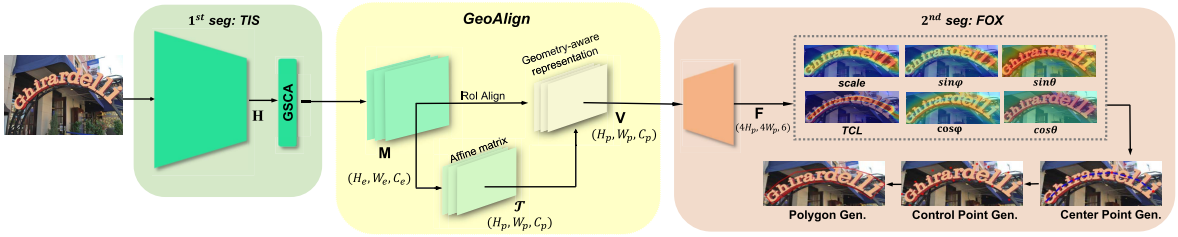


Fig. 2. The pipeline of NASK. $1^{st}seg$ and $2^{nd}seg$ mean the first and the second stage segmentation networks, respectively. In GeoAlign, RoI-wise affine transformations are predicted and embedded in feature map $\mathcal{T}$ to generate the geometry-aware representation feature map V.

false positives and achieves better performance when applied to rotated texts.

Though detection-based methods tend to have competitive performance on quadrilateral text detection, many of them fail to deal with curved texts limited by their baseline algorithms. In contrast, segmentation-based methods can naturally handle the more general arbitrary-shaped text case. However, they are more subject to background interference and more sensitive to the segmentation deviation [7]. To alleviate this dilemma, we design a two-stage segmentation network with the first stage to locate the rectangular text instances and the second stage to reconstruct a compact representation. Experimental results show that our two-stage architecture is more robust and efficient.

B. Arbitrary-Shaped Text Representation

Arbitrary-shaped text instances are of irregular layout and the conventional representations such as axis-aligned rectangles or quadrangles struggle with giving precise modeling. Several representations are proposed to fit texts of arbitrary shapes. TextSnake [5] describes text instances with a series of ordered, overlapping disks, each of which is sampled along the text center line and associated with specific radius and orientations. The final text shape is composed of circular circumscribed polygons. In this representation, the line between the tangent point and the corresponding circle center is perpendicular to the centerline, which may not be the most reasonable case. In NASK, we apply an additional character orientation prediction which models the text character direction. As illustrated in Fig. 1, with the added character orientation prediction, the proposed method has a more accurate and flexible representation, resulting in better detection results.

ABCNet [6] adopts a parameterized Bezier curve to fit arbitrarily-shaped texts. Specifically, it simplifies the detection problem to the control point regression problem, based on which the Bezier curve is generated. However, the Bezier curve assumption based on sparse control points is too restrictive and the Bernstein Polynomials [21] may not be the optimal solution. Besides, empirically, it tends to generate loose regions partially because of the sparse control points and fails to output compact detection results as shown in Fig. 1 (c). Compared to ABCNet, we do not presuppose the composition of the curve but represent it by a set of boundary points, which is more flexible and tighter.

C. Self-Attention

The self-attention mechanism has been proved to be effective in machine translation task [22]. Besides, it has also been widely used in other areas such as computer vision.

Non-local Operations [8], [23]–[26]: [8] proposes the non-local neural network based on the self-attention mechanism which computes the response at a position as a weighted sum of the features within all the same-channel positions. Due to its superior performance, it has been widely used in object detection and segmentation. Reference [27] applies the adapted non-local modules to increasing the resolution of feature maps in a coarse-to-fine manner, resulting in more accurate results. DANet [23] appends two types of attention modules, which model the semantic interdependencies in spatial and channel dimensions respectively. CCNet [24] captures the long-range dependency in a more efficient way, namely only considering the correlations among pixels on the criss-cross path.

Different from previous works, our GSCA extends the self-attention mechanism in both spatial and channel perspectives, and carefully designs the spatial and global channel attention branches to capture rich contextual relationships for better feature representations. The primary work DANet [23] also adopts a dual attention mechanism from both the spatial and channel perspectives. Our work, however, differs in the following two respects. Firstly, in the spatial attention module, GSCA takes a more radical approach that computes the correlations among all the elements in the feature map, not limited in the same-channel interrelationship. Secondly, in the channel attention branch, instead of computing the channel-wise correlations in DANet, we use a more simple yet efficient Squeeze-Excitation-like module [12] to achieve better performance while preserving the efficiency.

III. Approach

A. Overview

The overall pipeline of NASK is presented in Fig 2. It consists of three components: the first stage segmentation network TIS, a geometry-aware RoI transformation module GeoAlign and the second stage segmentation network FOX.

An input image is passed through the first stage segmentation network TIS. In order to efficiently aggregate the contextual information of the generated feature map **H**, we append a Group Spatial and Channel Attention Module after the fully convolutional network to obtain the more informative feature map **M**.

Given the refined feature map $\mathbf{M}$, we first threshold on pixels to obtain the binary classification map, *i.e.* text or non-text areas respectively. The *minAreaRect* method in OpenCV [28] is applied to group the predicted positive pixels into rectangle Connected Components. Then for the convenience of the next stage input, the cropped feature maps are required to be transformed into a fix size. Thus, we apply GeoAlign to conduct RoI-wise transformation. With GeoAlign, in addition to achieving the desired size normalization, we also obtain a more geometry-aware RoI feature representation, shown as the feature map $\mathbf{V}$ in Fig. 2.

Then, we feed the RoI features into a relatively simple segmentation network FOX with several convolution and up-sampling layers. The final output layer of FOX contains 6 channels, which represent the prediction of geometry attributes including text center line (TCL), character scale, character orientation and text orientation. Finally, text polygons are generated by applying *approxPolyDP* in OpenCV based on the detected fiducial points.

B. Group Spatial and Channel Attention Module

The conventional convolution is inherently a regional operation and limited to local receptive fields. The generated feature map with insufficient contextual information imposes a great adverse effect on the downstream tasks. To model comprehensive dependencies over local feature representations, we introduce a Group Spatial and Channel Attention Module, GSCA for short. GSCA captures contextual information in both spatial and channel aspects. As for the spatial relationship modeling, one may easily resort to the well-known Non-local Neural Network [8]. The Non-local operation, however, constrains the correlation modeling within the same channel, *i.e.,* it ignores the cross-channel element interactions. To alleviate this, we explicitly learn the correlations among all elements of the whole feature map (for both same-channel and cross-channel cases). Compared to Non-local Neural Network, GSCA exploits the spatial dependencies in a more radical way. In order to alleviate the huge computational overhead, we introduce the *channel grouping idea* to split all C channels into G groups and only the intra-group relationships (each group with $C' = C/G$ channels) are estimated. To capture the inter-group correlations, a Global Channel Attention branch is devised to generate the channel-wise attention and distribute information among every group.

As shown in Fig. 3, given the backbone generated feature map $\mathbf{H} \in \mathbb{R}^{H_e \times W_e \times C_e}$, for each position $\mathbf{u}$ in $\mathbf{H}$, we generate the intra-group affinity map $\mathbf{A} \in \mathbb{R}^{(H_e W_e C_e/G)\times(H_e W_e C_e/G)}$, which demonstrates the affinity relationships within the group. Specifically, the spatial-attended feature map $\mathbf{Y}'$ is generated as follows.

$$\mathbf{Y}' = concat\Big(f\big(g(\Theta(\mathbf{H})), g(\Phi(\mathbf{H}))\big)g(Q(\mathbf{H}))\Big), \tag{1}$$

where $\Theta(\mathbf{H})$, $\Phi(\mathbf{H})$, $Q(\mathbf{H})$ are learnable spatial transformations implemented as serially connected *convolution* and *reshape*. $f(\cdot,\cdot)$ is defined to be the matrix product for simplification. g is the grouping operation which divides the feature map into G groups along the channel dimension to relief the computation burden. *concat* denotes the channel-wise concatenation. Therefore, the output $\mathbf{Y}'$ shares the same shape as input $\mathbf{H}$.

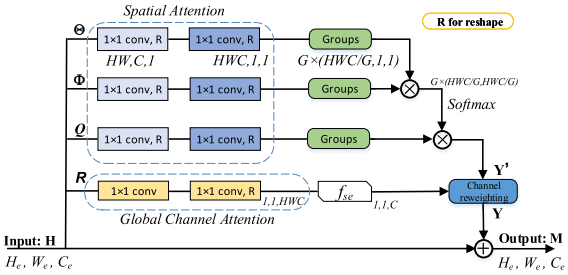
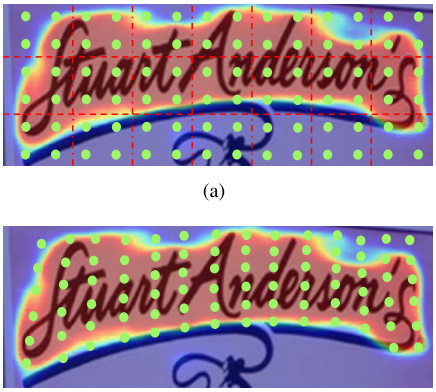


Fig. 3. The Group Spatial and Channel Attention module: Intra-group attention is learned by the serially connected spatial *convolution* and *reshape* denoted as Θ, Φ, Q while the global channel attention is captured by transformation $\mathbf{R}$ and f_{se}. "$\oplus$" denotes the element-wise sum while "$\otimes$" denotes matrix multiplication.



(a) (b)

Fig. 4. (a) RoI Align uniformly applies the pooling procedure with k^2 sample points in each bin, which brings about the background interference. (b) Our Geometry-aware RoI Alignment module adaptively selects the sample points within the text instances.

As for the Global Channel Attention branch, we capture the channel-wise weights λ with a squeeze-and-excitation module:

$$\boldsymbol{\lambda} = f_{se}(R(\mathbf{H})) = softmax\Big(W_2\big(\sigma(W_1\ H(\mathbf{H}))\big)\Big), \tag{2}$$

where $\boldsymbol{\lambda} \in \mathbb{R}^{1\times1\times C}$ is the chanel-wise attention weight and f_{se} is the excitation function. Specifically, σ denotes the ReLU function, $W_1 \in \mathbb{R}^{\kappa T\times T}$ and $W_2 \in \mathbb{R}^{T\times\kappa T}$ (κ is the expansion ratio) are the learnable parameters of two fully-connected layers. Thus, we apply the channel-wise reweighting as follows.

$$\mathbf{Y} = \lambda_i \mathbf{Y}'_{\mathbf{i}}, \tag{3}$$

where $i \in [1, C]$ is the channel index and C is the number of channels. λ_i and $\mathbf{Y}'_{\mathbf{i}}$ denote the i-th channel weight and i-th channel feature map respectively. Meanwhile, a short-cut path is used to preserve the local information and the final output $\mathbf{M}$ is the sum of $\mathbf{H}$ and $\mathbf{Y}$.

$$\mathbf{M} = \mathbf{H} + \mathbf{Y}. \tag{4}$$

C. Geometry-Aware RoI Alignment Module

To transform the varying-size RoIs into the fixed size, we have to apply a pooling-like module. In our previous

work [11], we simply apply the RoI Pooling module [9] to obtain the cropped RoI feature maps. Reference [10] has demonstrated that RoI Align is a better substitute for RoI Pooling. As shown in Fig. 4(a), RoI Align sets the sampling points in a uniform way, namely it averages k^2 points in each bin and then applies max-pooling. Due to the characteristic of curved texts, some sampling points are outside the text areas, which inevitably brings about the background interference. To address this problem, we take one step further to develop the Geometry-aware RoI Alignment module which adaptively samples points within the text areas. Before we specify our GeoAlign, let's revisit the RoI Align in detail.

For RoI Align, mathematically, given the RoI feature map $\mathbf{M} \in \mathbb{R}^{H_e \times W_e \times C_e}$ generated by the backbone network with GSCA, we have the following pooling feature map $\mathbf{V} \in \mathbb{R}^{H_p \times W_p \times C_p}$:

$$\mathbf{V}_{ij} = \frac{1}{k^2} \sum_{x=ki}^{k(i+1)-1} \sum_{y=kj}^{k(j+1)-1} \mathbf{M}(p(x, y)), \tag{5}$$

where $i \in [1, W_p]$, $j \in [1, H_p]$ denote the pixel index and k^2 is the number of sampling points within each bin. (x, y), $x \in [1, kW_p]$, $y \in [1, kH_p]$ is the horizontal and vertical sampling point index and $p(x, y)$ is the corresponding spatial position.

For GeoAlign, it adopts an additional affine transformation matrix $\mathcal{T}_{ij}$ to encode the geometry characteristics of the text information (e.g. rotation, translation, scale, and shear) and warps the uniformly sampling points to get the geometry-aware representation V as follows.

$$\mathbf{V}_{ij} = \frac{1}{k^2} \sum_{x=ki}^{k(i+1)-1} \sum_{y=kj}^{k(j+1)-1} \mathbf{M}(\mathcal{T}(p(x, y))), \tag{6}$$

where $\mathcal{T} \in \mathbb{R}^{kH_p \times kW_p \times 6}$ is the warping parameters for each sampling point. Specifically, the affine warping transformation process is as follows.

$$\mathcal{T}(p(x, y)) = \begin{bmatrix} \mathcal{T}_{x,y,1} & \mathcal{T}_{x,y,2} & \mathcal{T}_{x,y,3} \\ \mathcal{T}_{x,y,4} & \mathcal{T}_{x,y,5} & \mathcal{T}_{x,y,6} \end{bmatrix} \begin{pmatrix} p(x) \\ p(y) \\ 1 \end{pmatrix}, \tag{7}$$

where $\mathcal{T}_{x,y,i}$, $i \in [1, 6]$ represents 6-dimensional affine parameters for each sampling point position $p(x, y)$. $p(x)$ and $p(y)$ are the horizontal and vertical components of $p(x, y)$, respectively.

How to supervise the warping process? Namely, how to obtain the ground truth of the warped sampling points? Here, we take advantage of the boundary point annotations in the dataset and use the bilinear interpolation to obtain more dense sampling points. Finally, a simple L1 loss for the wrapped points is adopted and the details are presented in Eqn. 12.

D. Fiducial Point Expression Module

Building an appropriate representation for arbitrary-shaped texts plays an important role in accurate detection. We leverage on the fiducial points of the text instances to build an accurate and flexible representation. The detailed illustration of our FOX is depicted in Fig. 5. The geometrical attributes utilized to make up the text instances include the text center line (TCL), the character scale s, the character orientation ϕ and the text orientation θ.

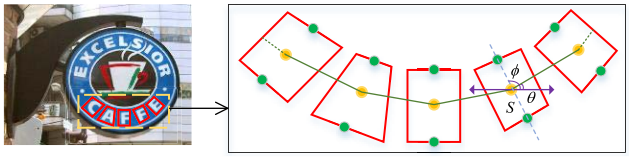


Fig. 5. The Illustration of the Fiducial Point Expression module. Center points are marked as yellow and fiducial points are marked as green.

Mathematically, a text instance can be viewed as an ordered sequence $S = \{S_1, \ldots, S_i, \ldots, S_n\}$, where n is the number of character segments. Each component S_i is a free-form quadrilateral. We construct the center point list $C = (c_{start}, c_1, \ldots, c_i, \ldots, c_n, c_{end})$, in which c_i is the center point (marked as yellow in Fig. 5) of S_i. Note that c_{start} is the midpoint of S_1's left edge and c_{end} is the midpoint of S_n's right edge. The center point list C is evenly sampled from the text center line (a side-shrunk version of text polygon annotations following [5]).

Following the above notations, we define $S_i = (c_i, s_i, \phi_i, \theta_i)$. The fiducial points (marked as green in Fig. 5) are defined as the midpoints of the top and bottom edges of each character quadrilateral. Thus, we compute the scale s_i as half the height of the character while the character orientation ϕ_i is the direction from the bottom-edge midpoint to the corresponding top-edge one. For the text orientation θ_i, it is defined as the horizontal angle between the current center c_i and the next one c_{i+1}.

Based on the dedicated designed fiducial point expression module, we set up a relatively simple segmentation network to generate the text polygon. The whole procedure can be divided into the following three steps.

Center Point Generation. Firstly, two up-sampling layers followed by one 1×1 convolution layer make up the full second stage segmentation network. Note that the final convolution layer is with 6 output channels to regress all the above geometrical attributes. Formally, the output is $\mathbf{F} = \{f_1, f_2, \ldots, f_6\}$ where f_1, f_2 denote the pixel-wise character scale s and the probability belonging to TCL respectively. After thresholding the feature map f_2, we obtain the text center line areas. Then center point list $C = (c_{start}, c_1, \ldots, c_i, \ldots, c_n, c_{end})$ are equidistantly sampled along the centerline.

Fiducial Point Generation. We use f_3 and f_4 to model the text orientation θ via its sine and cosine value. $sin\theta$ and $cos\theta$ are normalized to ensure their quadratic sum equals to 1:

$$\cos\theta = \frac{f_3}{\sqrt{f_3^2 + f_4^2}}$$
$$\sin\theta = \frac{f_4}{\sqrt{f_3^2 + f_4^2}}. \tag{8}$$

$sin\phi$ and $cos\phi$ are normalized with f_5 and f_6 in the same way.

For each c_i which has been obtained in the preceding step, according to the geometric relationship, two corresponding fiducial points in the bottom and top edges are computed as follows.

$$\begin{aligned} p_{2i-1} &= c_i + (s_i cos\phi_i, -s_i sin\phi_i) \\ p_{2i} &= c_i + (-s_i cos\phi_i, s_i sin\phi_i), \end{aligned} \tag{9}$$

where p_{2i-1} is the top-edge fiducial point for the center point c_i while p_{2i} is its bottom-edge counterpart. s_i and ϕ_i are the scale and the orientation for the i-th character respectively. Therefore, each text instance can be represented with $2n$ fiducial points.

Text Polygon Generation. Based on the obtained $2n$ fiducial points., we generate the text polygon for each instance via *approxPolyDP* in OpenCV [28] which approximates the polygon with given vertices.

E. Optimization

The overall loss function contains three terms corresponding to the three modules:

$$\mathcal{L} = \mathcal{L}_{\text{TIS}} + \alpha \mathcal{L}_{\text{Align}} + \beta \mathcal{L}_{\text{FOX}}, \tag{10}$$

where $\mathcal{L}_{\text{TIS}}$, $\mathcal{L}_{\text{Align}}$ and $\mathcal{L}_{\text{FOX}}$ are the loss for Text Instance Segmentation, the Geometry-aware RoI Alignment module and the Fiducial Point Expression module, respectively.

$\mathcal{L}_{\text{TIS}}$ is implemented as a cross-entropy loss with OHEM [29] adopted:

$$\mathcal{L}_{\text{TIS}} = \frac{1}{HWN} \sum_{i=1}^{H} \sum_{j=1}^{W} \sum_{n=1}^{N} -\log\left(p_n\left(\mathbf{M}_{i,j}\right)\right), \tag{11}$$

where H and W are the height and width of the output feature map $\mathbf{M}$ of the first stage segmentation network TIS. N represents the number of classification categories and here we set $N = 2$ for text and non-text areas respectively. $p_n(\mathbf{M}_{i,j})$ is the softmax score for the n-th class of the pixel $\mathbf{M}_{i,j}$.

To supervise the training for the geometry-aware alignment module, we apply the L1 loss for the sampling point warping.

$$\mathcal{L}_{\text{Align}} = \frac{1}{H_p W_p k^2} \left| \mathcal{T}(p(x, y)) - p^*(x, y) \right|, \tag{12}$$

where H_p and W_p denote the shape of the output pooling feature map. $x \in [1, kW_p]$ and $y \in [1, kH_p]$ are the horizontal and vertical index of the sampling points (k^2 points within each bin). $p^*(x, y)$ is the ground truth position, which is calculated by the interpolation of boundary points.

$\mathcal{L}_{\text{FOX}}$ represents the loss for all the regressed geometry attributes:

$$\begin{aligned} \mathcal{L}_{\text{FOX}} = \lambda_1 \mathcal{L}_{tcl} + \lambda_2 \mathcal{L}_s + \lambda_3 \mathcal{L}_{sin\theta} \\ + \lambda_4 \mathcal{L}_{cos\theta} + \lambda_5 \mathcal{L}_{sin\phi} + \lambda_6 \mathcal{L}_{cos\phi}, \end{aligned} \tag{13}$$

where $\mathcal{L}_{tcl}$ is the cross-entropy loss for TCL areas. $\mathcal{L}_s$, $\mathcal{L}_{sin\theta}$, $\mathcal{L}_{cos\theta}$, $\mathcal{L}_{sin\phi}$ and $\mathcal{L}_{cos\phi}$ are the Smoothed-L1 loss [9] computed as follows:

$$\begin{pmatrix} \mathcal{L}_s \\ \mathcal{L}_{sin\theta} \\ \mathcal{L}_{cos\theta} \\ \mathcal{L}_{sin\phi} \\ \mathcal{L}_{cos\phi} \end{pmatrix} = SmoothedL1 \begin{pmatrix} \frac{\widehat{s}-s}{s} \\ \widehat{sin\theta} - sin\theta \\ \widehat{cos\theta} - cos\theta \\ \widehat{sin\phi} - sin\phi \\ \widehat{cos\phi} - cos\phi \end{pmatrix}, \tag{14}$$

where $\widehat{s}$, $\widehat{sin\theta}$, $\widehat{cos\theta}$, $\widehat{sin\phi}$, $\widehat{cos\phi}$ are the predicted values while s, $sin\theta$, $cos\theta$, $sin\phi$, $cos\phi$ are the corresponding ground truth.

The hyper-parameters λ_1, λ_2, λ_3, λ_4, λ_5, λ_6, α, β are all set to 1 in our experiments.

IV. Experiments

A. Dataset and Evaluation Protocol

Total-Text [30] is a comprehensive scene text dataset for arbitrary-shaped texts. Except for the horizontal and multi-oriented texts, it contains a large amount of curved texts. All images are annotated with word-level polygons and transcriptions. The training and testing sets are with 1255 and 300 images respectively. We use the updated official Python scripts[1] to validate detection performance.

SCUT-CTW1500 [32] is another widely benchmarked scene text dataset proposed in 2017. It consists of 1000 training images and 500 testing images. Compared to Total-Text, it involves both English and Chinese texts. The text instances from this dataset are annotated with 14 boundary vertices. The evaluation script[2] is also provided by the official repository.

ICDAR 2015 (IC15) [39] is a commonly used dataset for multi-oriented text detection. It contains a total of 1500 pictures, 1000 of which are used for training and the remaining are for testing. The ground truth is annotated with word-level quadrangles. We also refer to the official online platform[3] for evaluation.

ICDAR 2017-MLT (IC17-MLT) [40] is a large scale multilingual text dataset, which includes 7200 training images, 1800 validation images and 9000 testing images from 9 languages. The annotation is denoted with 4 vertices of the quadrangle. The official platform[4] is utilized for evaluation.

B. Implementation Details

1) Network Structure: For TIS, we choose the ImageNet [41] pre-trained ResNet-50 [42] as our backbone network with the last two down-sampling operations removed. For Geometry-aware RoI Alignment, we predefine the shape of the output feature map to be 8×64. The second segmentation network, namely FOX, is relatively simple with two up-sampling layers followed by one 1×1 convolution with 6 output channels and the shape of the output feature map is 32×256.

[1] https://github.com/cs-chan/Total-Text-Dataset/tree/master/Evaluation_Protocol

[2] https://github.com/Yuliang-Liu/TIoU-metric/tree/master/curved-tiou

[3] https://rrc.cvc.uab.es/?ch=4

[4] https://rrc.cvc.uab.es/?ch=8

TABLE I
RESULTS ON TOTAL-TEXT, SCUT-CTW 1500, ICDAR 2015, ICDAR 2017 MLT DATASETS

Model	Total-Text				CTW 1500				ICDAR 2015				ICDAR 2017 MLT			
	R	*P*	*H*	*F*	*R*	*P*	*H*	*F*	*R*	*P*	*H*	*F*	*R*	*P*	*H*	*F*
Deconv [30]	40.0	33.0	36.0	-	-	-	-	-	-	-	-	-	-	-	-	-
TextField [31]	79.9	81.2	80.6	-	-	-	-	-	80.5	84.3	82.4	**6.0**	-	-	-	-
CTPN [3]	-	-	-	-	53.8	60.4	56.9	7.14	52.0	74.0	61.0	-	-	-	-	-
CTD [32]	-	-	-	-	69.8	77.4	73.4	13.3	-	-	-	-	-	-	-	-
SLPR [33]	-	-	-	-	70.1	80.1	74.8	-	83.6	85.5	84.5	-	-	-	-	-
SegLink [34]	23.8	30.3	26.7	-	40.0	42.3	40.8	10.7	76.8	73.1	75.0	-	-	-	-	-
EAST [15]	36.2	50.0	42.0	-	49.1	78.7	60.4	**21.2**	78.3	83.3	80.7	-	-	-	-	-
PSENet [35]	75.1	81.8	78.3	3.9	75.6	80.6	78.0	3.9	79.7	81.5	80.6	1.6	68.4	77.0	72.5	-
TextSnake [5]	74.5	82.7	78.4	-	**85.3**	67.9	75.6	-	80.4	84.9	82.6	1.1	-	-	-	-
LOMO [36]	75.7	**88.6**	81.6	-	69.6	**89.2**	78.4	-	83.5	**91.3**	87.2	-	60.6	78.8	68.5	-
ABCNet [6]	-	-	78.4	-	-	-	74.1	-	-	-	-	-	-	-	-	-
TextRay [7]	77.9	83.5	80.6	-	80.4	82.8	81.6	-	-	-	-	-	-	-	-	-
FOTS [37]	-	-	-	-	-	-	-	-	85.2	91.0	88.0	-	57.5	81.0	67.3	-
SPCNet [20]	82.8	83.0	82.9	-	-	-	-	-	85.8	88.7	87.2	-	66.9	73.4	70.0	-
PMTD [38]	-	-	-	-	-	-	-	-	87.4	**91.3**	89.3	-	72.8	85.2	78.5	-
NASK$_{conf}$ [11]	81.2	83.3	82.2	8.4	78.3	82.8	80.5	12.1	86.8	90.2	88.5	4.2	70.4	83.6	76.4	3.5
NASK	**83.2**	85.6	**84.4**	**8.4**	80.1	83.4	**81.7**	12.1	**89.2**	90.9	**90.0**	**4.2**	**73.6**	**86.4**	**79.5**	**3.5**

Note: *R*, *P*, *H*, *F* denote Recall, Precision, Hmean and FPS respectively. NASK$_{conf}$ is our previous conference version [11]. All data are given in percentile form.

2) Training Settings: We implement our method in PyTorch.[5] For all datasets, images are randomly cropped and resized into 512×512. The cropped image regions are rotated randomly in 4 directions with 0°, 90°, 180°, 270°. All experiments are conducted on four NVIDIA TitanX GPUs each with 12GB memory. The CPU configuration is Intel(R) Xeon(R) CPU E5-2680 v4 @ 2.40GHz. The Adam optimizer is adopted here. We design a warm-up training strategy with the first segmentation network pre-trained on Synthetic dataset [43] for 10 epochs with learning rate set to 2×10^{-4}. This strategy leads to a precise first-stage segmentation, which is a prerequisite for the subsequent text shape refinement. Then the whole model including TIS, GeoAlign and FOX is fine-tuned with the initial learning rate 10^{-4} and the learning rate decay factor is set to 0.9.

We evaluate the performance of $NASK$ on Total-Text and SCUT-CTW1500 after finetuning about 10 epochs. The number of sample points n in TCL is set to 8 and the group number G of GSCA is set to 4. Thresholds T_{tr}, T_{tcl} for regarding pixels to be text regions or TCL are set to (0.7,0.6) and (0.8,0.4) respectively for Total-Text and SCUT-CTW1500. Since ICDAR 2015 dataset only contains the multi-oriented text instances, we reduce the TCL sampling points to 4 for simplification. The GSCA group number remains 4. T_{tr}, T_{tcl} are set to (0.6, 0.5). ICDAR 2017-MLT is annotated in the quadrangle format and we also set TCL sampling points to 4. GSCA group number is 4. T_{tr}, T_{tcl} are set to (0.5, 0.5).

3) Inference Settings: During inference, given the input image, the longer size is resized to 512 while keeping the original aspect ratio. Then the input images are fed to the first and the second network in sequence, yielding the required geometry attribute values. We reconstruct the fiducial points using Eqn. 9. The final detection results are generated by applying *approxPolyDP* in OpenCV to obtain the compact results.

[5]https://pytorch.org/

C. Evaluation on Curved Text Benchmarks

As shown in Table I, on the Total-Text dataset, NASK achieves impressive performance compared with state-of-the-arts. Specifically, it achieves the highest *H-mean* value (84.4%) with *FPS* reaching 8.4. Compared with our conference version [11], our method achieves 2.2% performance gain in *H-mean* with no reduction in efficiency. Besides, the quantitative results on SCUT-CTW1500 dataset also show NASK achieves a competitive result comparable to state-of-the-arts. Although the $recall$ and $precision$ value of NASK is inferior, it obtains the optimal $H\text{-}mean$. Since there is a trade-off between $recall$ and $precision$, $H\text{-}mean$ is a more objective measurement for performance assessments. For qualitative evaluation, some detection results are shown in Fig. 6(a) and Fig. 6(b).

D. Evaluation on Multi-Oriented Text Benchmarks

NASK is a general text detector and can be applied to the multi-oriented text benchmark as well. We verify the superiority of our method on the oriented text by conducting experiments on ICDAR 2015. Quantitative and qualitative results are shown in Table I and Fig. 6(c), respectively. NASK achieves the $H\text{-}mean$ of 90.0%, which consistently outperforms the previous state-of-the-art methods. Moreover, our method also obtains the best $recall$ of 89.2% among all methods in Table I. For the cross-language dataset ICDAR2017-MLP, NASK also has the leading performance compared to the existing methods, *i.e.,* it achieves the state-of-the-art 79.5% $H\text{-}mean$. The visualizations of multilingual text detection are shown in Fig. 6(d).

E. Speed Analysis

Since the inference speed is closely related to hardware, we detail the settings for the reported FPS in Table I. On Tatal-Text, PSENet is tested with a single Titan Xp GPU according to the original paper, which is the same as our device.

(a) Total Text (b) SCUT-CTW 1500 (c) ICDAR 2015 (d) ICDAR 2017-MLT

Fig. 6. Qualitative detection results of Total Text, SCUT-CTW 1500, ICDAR 2015 and ICDAR 2017-MLT.

(a) Visualization of GSCA feature map

(b) Global Channel Attention Map

Fig. 7. (a) Column 1: images with a red cross mark (called *query pixel*) which is a selected position in $\boldsymbol{Q}$ shown in Fig. 3. Column 2 to 5: related feature heatmaps computed with GSCA. Specifically, we use the corresponding vectors in Φ and Θ to compute attention maps according to Eqn. 1. (b) Global Channel Attention Map displays the weight distribution along the channel.

TABLE II
ABLATION STUDIES ON SCUT-CTW 1500

Experiment	$1^{st}seg$	$2^{nd}seg$	*Attention*	*Pooling*	G	R	P	H	F
(a)	✓	✓	GSCA	GeoAlign	0	78.2	81.7	79.5	16.7
					1	80.9	83.8	82.3	1.2
					2	80.8	83.8	82.3	3.4
					4	80.1	83.4	81.7	12.1
					8	79.2	82.2	80.7	12.9
					12	78.7	81.8	80.2	13.7
					16	78.2	81.0	79.6	13.8
(b)	✗	✓	GSCA	GeoAlign	4	75.2	76.4	75.8	14.7
	✓	✗				73.1	72.4	72.7	18.8
	✓	✓				80.1	83.4	81.7	12.1
(c)	✓	✓	GSCA	GeoAlign	4	80.1	83.4	81.7	12.1
			$GSCA_{conf}$			78.9	82.5	80.7	12.1
			DANet		-	79.8	83.1	81.4	10.3
			CCNet		-	78.7	82.4	80.5	13.8
			None		-	78.2	81.1	79.5	16.7
(d)	✓	✓	GSCA	GeoAlign	4	80.1	83.4	81.7	13.1
				RoI Align		79.2	82.4	80.8	13.6
				RoI Pooling		78.5	81.7	80.1	14.3

Note: $1^{st}seg$ and $2^{nd}seg$ mean the first and the second stage segmentation network; *Attention* denotes different attention modules including GSCA, $GSCA_{conf}$, DANet, and CCNet; $GSCA_{conf}$ denotes the preceding version of GSCA in the conference paper [11]. *Pooling* denotes the adopted pooling methods including our proposed GeoAlign, RoI Align, and RoI Pooling. G denotes the group number of GSCA and $GSCA_{conf}$. All data are given in percentile form.

On CTW 1500, all the reported FPS values are obtained from [44] and all the experiments are conducted on a single Nvidia 1080 GPU. For a fair comparison, we also test our NASK on the same Nvidia 1080 GPU and NASK reaches 9.7 FPS, which is still competitive. On ICDAR 2015, all reported values (TextField, PSENet and TextSnake) are all tested on Titan Xp GPU, so it is comparable.

F. Ablation Studies

In this section, we conduct several ablation studies to provide more insights about our design intuition. All the ablation experiments are performed on the SCUT-CTW1500 dataset.

1) Ablation Study of GSCA:

Effectiveness of GSCA. We explore to reveal how GSCA helps. Specifically, we apply a set of comparative experiments with different G values. As shown in Table II (a), GSCAs with all G values lead to the performance gain in *H-mean* compared to the native model ($G = 0$). For instance, by setting G to 4, *H-mean* improves by 2.2%. In this case, it strikes a balance between performance and efficiency. Therefore, the group attention mechanism enhances the long-range relationship and boosts the detection accuracy with the affordable overhead.

We also present the visualization results of GSCA. In Fig. 7(a), we visualize the group-wise spatial attention map. Specifically, we randomly select one pixel in the input image and regard it as the *query pixel* in the feature map $\boldsymbol{Q}$ shown in Fig. 3. Then we use the corresponding vectors in Φ and Θ to compute attention maps according to Eqn. 1. The results indicate that GSCA is context-aware *i.e.,* most of the weights are focused on the pixels belonging to the same category with *query pixel*. Fig. 7(b) presents the channel-wise weight distribution computed by the Global Channel Attention branch, which helps the training process focus on the most discriminative channels.

Influence of the number of attention module groups *G*. There is a trade-off between the accuracy and the speed when setting the different group numbers of GSCA. Intuitively, less grouping leads to higher accuracy and lower speed, and vice versa. In Table II (a), as expected, the detection speed increases with the rise of the group number and reaches the limit at about 13.8 *FPS*. Notably, it is noticed that the quantitative performance is not much sensitive to G when $G \geq 4$. This may be explained by the fact that the Global Channel Attention

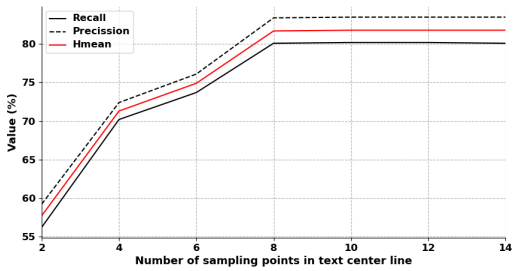


Fig. 8. Ablation studies of the number of sample points.

branch in Fig. 3 effectively captures the rich correlations among groups, thus alleviating the negative effects of the grouping operation.

GSCA vs. DANet vs. CCNet. To demonstrate the superiority of GSCA, we replaced GSCA with two widely used self-attention modules, DANet [23] and CCNet [24], respectively, while keeping the rest of the network unchanged. The comparison results list in Table II (c) show that the model equipped with GSCA outperforms that with CCNet, which may be due to the fact that CCNet only considers the spatial correlations. While DANet shares the similar *H-mean* with our GSCA, it has a lower *FPS* than GSCA (10.3 *vs.* 12.1). Therefore, in our task, GSCA is a more effective and efficient attention module compared with the state-of-the-arts. Besides, we also compare GSCA to our conference version GSCA_{conf} and the results show that GSCA outperforms GSCA_{conf} with no degrade in speed.

2) Ablation Study of Geometry-Aware RoI Alignment: **Influence of GeoAlign.** The proposed Geometry-aware RoI Alignment module effectively selects the sampling points and avoids the background interference. We conduct comparison experiments by replacing our Geometry-aware RoI Pooling with RoI Align [10] and RoI Pooling [9], respectively. The results list in Table II (d) demonstrate that the GeoAlign-equipped model achieves the best performance (1.6% *H-mean* gain compared to RoI Pooling) among the three variants with negligible overheads. Therefore, GeoAlign generates more informative features for the following geometric attribute prediction.

3) Ablation Study of Fiducial pOint eXpression Module: **Influence of the number of sample points *n*.** With our Fiducial Point Expression module, the curve text representation is decided by a set of $2n$ fiducial points, thus the number of text center line sample points is an important hyper-parameter. To explore this, we evaluate the performance under different values of n. The results shown in Fig. 8 witness a sustained increase when n changes from 2 to 8 and then the performance gradually converges. Therefore, we set n to 8 in our experiments.

4) Ablation Study of the Two-Stage Architecture Design: **Effectiveness of the two-stage segmentation.** To demonstrate the efficacy of our two-stage segmentation architecture, we conduct comparison experiments that only apply the first or the second segmentation network. (1) When only applying the first stage segmentation, it falls into a simple segmentation task. (2) For the experiment with only the second-stage segmentation network, we directly apply FOX on the input image and reconstruct the curved text instances.

The comparison results in Table II (b) show that the performance of the two-stage segmentation surpasses the single-stage segmentation by a large margin. The variant with only the first stage segmentation can not describe arbitrary-shaped text instances accurately, thus having inferior performance. For the other variant which drops the first stage of rectangle text instance segmentation, the geometric properties FOX refers to need to be predicted on the input image directly, which leads to the decrease of detection accuracy (75.8% *vs.* 81.7% in *H-mean*).

V. Conclusion

In this paper, we propose a novel two-stage segmentation-based text detector NASK to facilitate arbitrary-shaped text detection. We firstly leverage a text instance segmentation network TIS to obtain the rectangle proposals. To capture the long-range dependency, a self-attention based mechanism called Group Spatial and Channel Attention module (GSCA) is incorporated into TIS to augment the feature representation. Then Geometry-aware Text RoI Alignment (GeoAlign), a reformative alternative for RoI Align, is applied to warp the rectangle text proposals to the fixed size. Finally, we propose a Fiducial Point Expression module (FOX) which utilizes fiducial points to represent the arbitrary-shaped texts. Experiment results on both the multi-oriented and curved text datasets have demonstrated the effectiveness and efficiency of our proposed NASK.

Acknowledgment

Special acknowledgment are given to Aoto-PKUSZ Joint Lab for its support.

References

[1] M. Liao, B. Shi, X. Bai, X. Wang, and W. Liu, "TextBoxes: A fast text detector with a single deep neural network," 2016, *arXiv:1611.06779*. [Online]. Available: http://arxiv.org/abs/1611.06779

[2] M. Liao, B. Shi, and X. Bai, "TextBoxes++: A single-shot oriented scene text detector," *IEEE Trans. Image Process.*, vol. 27, no. 8, pp. 3676–3690, Aug. 2018.

[3] Z. Tian, W. Huang, T. He, P. He, and Y. Qiao, "Detecting text in natural image with connectionist text proposal network," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 56–72.

[4] W. Feng, W. He, F. Yin, X.-Y. Zhang, and C.-L. Liu, "TextDragon: An end-to-end framework for arbitrary shaped text spotting," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9076–9085.

[5] S. Long, J. Ruan, W. Zhang, X. He, W. Wu, and C. Yao, "TextSnake: A flexible representation for detecting text of arbitrary shapes," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 20–36.

[6] Y. Liu, H. Chen, C. Shen, T. He, L. Jin, and L. Wang, "ABCNet: Real-time scene text spotting with adaptive bezier-curve network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9809–9818.

[7] F. Wang, Y. Chen, F. Wu, and X. Li, "TextRay: Contour-based geometric modeling for arbitrary-shaped scene text detection," 2020, *arXiv:2008.04851*. [Online]. Available: http://arxiv.org/abs/2008.04851

[8] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7794–7803.

[9] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[10] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 2961–2969.

[11] M. Cao and Y. Zou, "All you need is a second look: Towards tighter arbitrary shape text detection," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 2228–2232.

[12] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7132–7141.

[13] W. Liu *et al.*, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, Oct. 2016, pp. 21–37.

[14] J. Ma *et al.*, "Arbitrary-oriented scene text detection via rotation proposals," *IEEE Trans. Multimedia*, vol. 20, no. 11, pp. 3111–3122, Nov. 2018.

[15] X. Zhou *et al.*, "EAST: An efficient and accurate scene text detector," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5551–5560.

[16] Z. Zhang, C. Zhang, W. Shen, C. Yao, W. Liu, and X. Bai, "Multi-oriented text detection with fully convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4159–4167.

[17] M. Cao, Y. Zou, D. Yang, and C. Liu, "GISCA: Gradient-inductive segmentation network with contextual attention for scene text detection," *IEEE Access*, vol. 7, pp. 62805–62816, 2019.

[18] P. Lyu, C. Yao, W. Wu, S. Yan, and X. Bai, "Multi-oriented scene text detection via corner localization and region segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7553–7563.

[19] W. Wang *et al.*, "Shape robust text detection with progressive scale expansion network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9336–9345.

[20] E. Xie, Y. Zang, S. Shao, G. Yu, C. Yao, and G. Li, "Scene text detection with supervised pyramid context network," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 9038–9045.

[21] G. G. Lorentz, *Bernstein Polynomials*. Providence, RI, USA: AMS, 2013.

[22] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[23] J. Fu *et al.*, "Dual attention network for scene segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3146–3154.

[24] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, and W. Liu, "CCNet: Criss-cross attention for semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 603–612.

[25] S. Hong, Y. Zou, W. Wang, and M. Cao, "Weakly labelled audio tagging via convolutional networks with spatial and channel-wise attention," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 296–300.

[26] W. Xie, J. Zhang, Z. Lu, M. Cao, and Y. Zhao, "Non-local nested residual attention network for stereo image super-resolution," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 2643–2647.

[27] Y. Yuan, L. Huang, J. Guo, C. Zhang, X. Chen, and J. Wang, "OCNet: Object context network for scene parsing," 2018, *arXiv:1809.00916*. [Online]. Available: http://arxiv.org/abs/1809.00916

[28] G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision With the OpenCV Library*. Newton, MA, USA: O'Reilly Media, 2008.

[29] A. Shrivastava, A. Gupta, and R. Girshick, "Training region-based object detectors with online hard example mining," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 761–769.

[30] C. K. Ch'ng and C. S. Chan, "Total-text: A comprehensive dataset for scene text detection and recognition," in *Proc. 14th IAPR Int. Conf. Document Anal. Recognit. (ICDAR)*, Nov. 2017, pp. 935–942.

[31] Y. Xu, Y. Wang, W. Zhou, Y. Wang, Z. Yang, and X. Bai, "TextField: Learning a deep direction field for irregular scene text detection," *IEEE Trans. Image Process.*, vol. 28, no. 11, pp. 5566–5579, Nov. 2019.

[32] L. Yuliang, J. Lianwen, Z. Shuaitao, and Z. Sheng, "Detecting curve text in the wild: New dataset and new solution," 2017, *arXiv:1712.02170*. [Online]. Available: http://arxiv.org/abs/1712.02170

[33] Y. Zhu and J. Du, "Sliding line point regression for shape robust scene text detection," in *Proc. 24th Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2018, pp. 3735–3740.

[34] B. Shi, X. Bai, and S. Belongie, "Detecting oriented text in natural images by linking segments," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2550–2558.

[35] X. Li, W. Wang, W. Hou, R.-Z. Liu, T. Lu, and J. Yang, "Shape robust text detection with progressive scale expansion network," 2018, *arXiv:1806.02559*. [Online]. Available: http://arxiv.org/abs/1806.02559

[36] C. Zhang *et al.*, "Look more than once: An accurate detector for text of arbitrary shapes," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 10552–10561.

[37] X. Liu, D. Liang, S. Yan, D. Chen, Y. Qiao, and J. Yan, "FOTS: Fast oriented text spotting with a unified network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5676–5685.

[38] J. Liu, X. Liu, J. Sheng, D. Liang, X. Li, and Q. Liu, "Pyramid mask text detector," 2019, *arXiv:1903.11800*. [Online]. Available: http://arxiv.org/abs/1903.11800

[39] D. Karatzas *et al.*, "ICDAR 2015 competition on robust reading," in *Proc. 13th Int. Conf. Document Anal. Recognit. (ICDAR)*, Aug. 2015, pp. 1156–1160.

[40] N. Nayef *et al.*, "ICDAR2017 robust reading challenge on multi-lingual scene text detection and script identification–RRC-MLT," in *Proc. 14th IAPR Int. Conf. Document Anal. Recognit. (ICDAR)*, vol. 1, Nov. 2017, pp. 1454–1459.

[41] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[42] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[43] A. Gupta, A. Vedaldi, and A. Zisserman, "Synthetic data for text localisation in natural images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2315–2324.

[44] Y. Liu, L. Jin, S. Zhang, C. Luo, and S. Zhang, "Curved scene text detection via transverse and longitudinal sequence connection," *Pattern Recognit.*, vol. 90, pp. 337–345, Jun. 2019.

Meng Cao received the B.E. degree from the Huazhong University of Science and Technology (HUST), in 2018. He is currently pursuing the M.Sc. degree with the School of Electronic and Computer Engineering (ECE), Peking University. His current research interests include computer vision and computer graphics.

Can Zhang received the B.Eng. degree in electronic engineering from Shanghai University, Shanghai, China, in 2017. He is currently pursuing the Ph.D. degree with the School of Electronic and Computer Engineering (ECE), Peking University, Shenzhen, China. His current research interests include computer vision, multimedia computing, and video understanding.

Dongming Yang (Member, IEEE) received the B.E. degree from Shan Xi University, in 2015, and the M.Sc. degree from the University of Chinese Academy of Sciences, in 2018. He is currently pursuing the Ph.D. degree in computer science and engineering from Peking University. His research interests include computer vision and pattern recognition. He is also an Intern with the Peng Cheng Laboratory.

Yuexian Zou (Senior Member, IEEE) received the M.Sc. degree from the University of Electronic Science and Technology of China, in 1991, and the Ph.D. degree from The University of Hong Kong, in 2000. She is currently a Full Professor with Peking University and the Director of the Advanced Data and Signal Processing Laboratory, Peking University Shenzhen Graduate School. Her main research interests include machine learning for signal processing and deep learning and its applications. She was a recipient of the Award Leading Figure for Science and Technology by Shenzhen Municipal Government, in 2009.