

A Single-Shot Arbitrarily-Shaped Text Detector based on Context Attended Multi-Task Learning

Pengfei Wang*
School of Artificial Intelligence,
Xidian University
pfwang@stu.xidian.edu.cn

Chengquan Zhang*
Department of Computer Vision
Technology (VIS), Baidu Inc.
zhangchengquan@baidu.com

Fei Qi*[†]
School of Artificial Intelligence,
Xidian University
fred.qi@ieee.org

Zuming Huang
Department of Computer Vision
Technology (VIS), Baidu Inc.
huangzuming@baidu.com

Mengyi En
Department of Computer Vision
Technology (VIS), Baidu Inc.
enmengyi@baidu.com

Junyu Han
Department of Computer Vision
Technology (VIS), Baidu Inc.
hanjunyu@baidu.com

Jingtuo Liu
Department of Computer Vision
Technology (VIS), Baidu Inc.
liujingtuo@baidu.com

Errui Ding
Department of Computer Vision
Technology (VIS), Baidu Inc.
dingerrui@baidu.com

Guangming Shi
School of Artificial Intelligence,
Xidian University
gmshi@xidian.edu.cn

ABSTRACT

Detecting scene text of arbitrary shapes has been a challenging task over the past years. In this paper, we propose a novel segmentation-based text detector, namely SAST, which employs a context attended multi-task learning framework based on a Fully Convolutional Network (FCN) to learn various geometric properties for the reconstruction of polygonal representation of text regions. Taking sequential characteristics of text into consideration, a Context Attention Block is introduced to capture long-range dependencies of pixel information to obtain a more reliable segmentation. In post-processing, a Point-to-Quad assignment method is proposed to cluster pixels into text instances by integrating both high-level object knowledge and low-level pixel information in a single shot. Moreover, the polygonal representation of arbitrarily-shaped text can be extracted with the proposed geometric properties much more effectively. Experiments on several benchmarks, including ICDAR2015, ICDAR2017-MLT, SCUT-CTW1500, and Total-Text, demonstrate that SAST achieves better or comparable performance in terms of accuracy. Furthermore, the proposed algorithm runs at 27.63 FPS on SCUT-CTW1500 with a Hmean of 81.0% on a single NVIDIA Titan Xp graphics card, surpassing most of the existing segmentation-based methods.

KEYWORDS

FCN; Arbitrarily-shaped Text Detection; Real-time Segmentation

*Equal contribution. This work is done when Pengfei Wang is an intern at Baidu Inc.

[†]Corresponding author.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

MM '19, October 21–25, 2019, Nice, France

© 2019 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-6889-6/19/10...\$15.00

<https://doi.org/10.1145/3343031.3350988>

ACM Reference Format:

Pengfei Wang, Chengquan Zhang, Fei Qi, Zuming Huang, Mengyi En, Junyu Han, Jingtuo Liu, Errui Ding, and Guangming Shi. 2019. A Single-Shot Arbitrarily-Shaped Text Detector based on Context Attended Multi-Task Learning. In *Proceedings of the 27th ACM International Conference on Multimedia (MM '19)*, October 21–25, 2019, Nice, France. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/3343031.3350988>

1 INTRODUCTION

Recently, scene text reading has attracted extensive attention in both academia and industry for its numerous applications, such as scene understanding, image and video retrieval, and robot navigation. As the prerequisite in textual information extraction and understanding, text detection is of great importance. Thanks to the surge of deep neural networks, various convolutional neural network (CNN) based methods have been proposed to detect scene text, continuously refreshing the performance records on standard benchmarks [1, 15, 28, 42]. However, text detection in the wild is still a challenging task due to the significant variations in size, aspect ratios, orientations, languages, arbitrary shapes, and even the complex background. In this paper, we seek an effective and efficient detector for text of arbitrary shapes.

To detect arbitrarily-shaped text, especially those in curved form, some segmentation-based approaches [23, 35, 37, 44] formulated text detection as a semantic segmentation problem. They employ a fully convolutional network (FCN) [27] to predict text regions, and apply several post-processing steps such as connected component analysis to extract final geometric representation of scene text. Due to the lack of global context information, there are two common challenges for segmentation-based text detectors, as demonstrated in Fig. 1, including: (1) Lying close to each other, text instances are difficult to be separated via semantic segmentation; (2) Long text instances tend to be fragmented easily, especially when character spacing is far or the background is complex, such as the effect of strong illumination. In addition, most segmentation-based detectors have to output large-resolution prediction to precisely describe text

contours, thus suffer from time-consuming and redundant post-processing steps.

Some instance segmentation methods [4, 7, 45] attempt to embed high-level object knowledge or non-local information into the network to alleviate the similar problems described above. Among them, Mask-RCNN [7], a proposal-based segmentation method that cascades detection task (i.e., RPN [29]) and segmentation task by RoIAlign [7], has achieved better performance than those proposal-free methods by a large margin. Recently, some similar ideas [14, 24, 39] have been introduced to settle the problem of detecting text of arbitrary shapes. However, they are all facing a common challenge that it takes much more time when the number of valid text proposals increases, due to the large number of overlapping computations in segmentation, especially in the case that valid proposals are dense. In contrast, our approach is based on a single-shot view and efficient multi-task mechanism.

Inspired by recent works [16, 22, 33] in general semantic instance segmentation, we aim to design a segmentation-based Single-shot Arbitrarily-Shaped Text detector (SAST), which integrates both the high-level object knowledge and low-level pixel information in a single shot and detects scene text of arbitrary shapes with high accuracy and efficiency. Employing a FCN [27] model, various geometric properties of text regions, including text center line (TCL), text border offset (TBO), text center offset (TCO), and text vertex offset (TVO), are designed to learn simultaneously under a multi-task learning formulation. In addition to skip connections, a Context Attention Block (CAB) is introduced into the architecture to aggregate contextual information for feature augmentation. To address the problems illustrated in Fig. 1, we propose a point-to-quad method for text instance segmentation, which assigns labels to pixels by combining high-level object knowledge from TVO and TCO maps. After clustering TCL map into text instances, more precise polygonal representations of arbitrarily-shaped text are then reconstructed based on TBO maps.

Experiments on public datasets demonstrate that the proposed method achieves better or comparable performance in terms of both accuracy and efficiency. The contribution of this paper are three-fold:

- We propose a single-shot text detector based on multi-task learning for text of arbitrary shapes including multi-oriented, multilingual, and curved scene text, which is efficient enough for some real-time applications.
- The Context Attention Block aggregates the contextual information to augment the feature representation without too much extra calculation cost.
- The point-to-quad assignment is robust and effective to separate text instance and alleviate the problem of fragments, which is better than connected component analysis.

2 RELATED WORK

In this section, we will review some representative segmentation-based and detection-based text detectors, as well as some recent progress in general semantic segmentation. A comprehensive review of recent scene text detectors can be found in [40, 49].

Segmentation-based Text Detectors. The greatest benefit of segmentation-based methods is the ability to detect both straight

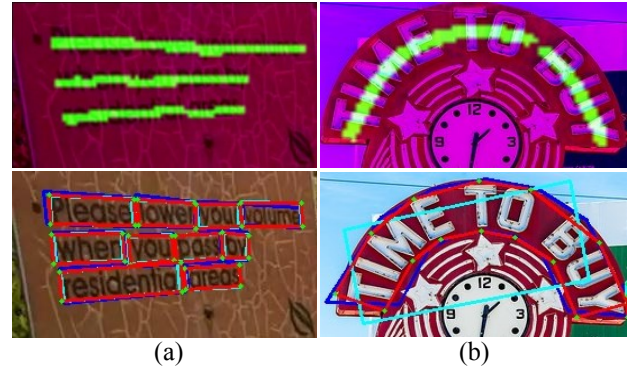


Figure 1: Two common challenges for segmentation-based text detectors: (a) Adjacent text instances are difficult to be separated; (b) The long text may break into fragments. The first row is the response for text regions from segmentation branch. In the second row, cyan contours are the detection results from EAST [47], while red contours are from SAST.

text and curved text in a unified manner. With FCN [27], the segmentation-based text detectors first classify text at pixel level, then followed by several post-processing to extract final geometric representation of scene text, so the performance of this kind of detectors is strongly affected by the robustness of segmentation results. In PixelLink [2], positive pixels are joined into text instances by predicted positive links, and the bounding boxes are extracted from segmentation result directly. TextSnake [23] proposes a novel presentation for arbitrarily-shaped text, and treats a text instance as a sequence of overlapping disks lying at text center line to describe the geometric properties of text instances of irregular shapes. PSENet [35] shrinks the original text instance into various scales, and gradually expands the kernels to the text instances of complete shapes through a progressive scale expansion algorithm. The main challenge of FCN-based methods is separating text instances, which lie close to each other. The runtime of approaches above highly depends on the employed post-processing step, which often involves several pipelines and tends to be rather slow.

Detection-based Text Detectors. Scene text is regarded as a special type of object, several methods [10, 17, 18, 26, 43, 47] are based on Faster R-CNN [29], SSD [20] and DenseBox [12], which generates text bounding boxes by regressing coordinates of boxes directly. TextBoxes [17] and RRD [18] adopt SSD as a base detector and adjust the anchor ratios and convolution kernel size to handle variation of aspect ratios of text instances. He et al. [10] and EAST [47] perform direct regression to determine vertex coordinates of quadrilateral text boundaries in a per-pixel manner without using anchors and proposals, and conduct the Non-Max Suppression (NMS) to get the final detection results. RRPN [26] generates inclined proposals with text orientation angle information and propose Rotation Region-of-Interest (RRoI) pooling layer to detect arbitrary-oriented text. Limited by the receptive field of CNNs and the relatively simple representations like rectangle bounding box or quadrangle adopted to describe text, detection-based methods may fall short when dealing with more challenging text instances, such as extremely long text and arbitrarily-shaped text.

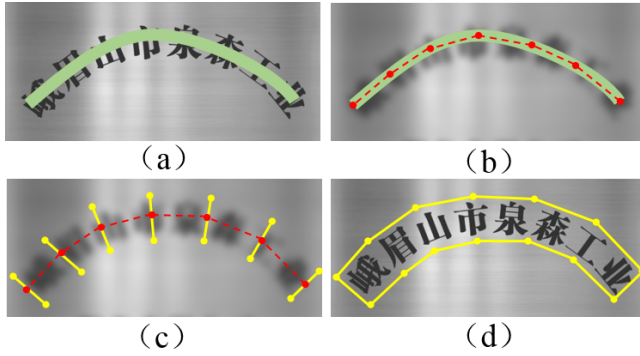


Figure 2: Arbitrary Shape Representation: a) The text line in TCL map; b) Sample adaptive number of points in the line; c) Calculate the corresponding border point pairs with TBO map; d) Link all the border points as final representation.

General Instance Segmentation. Instance segmentation is a challenging task, which involves both segmentation and classification tasks. The most recent and successful two-stage representative is Mask R-CNN [7], which achieves amazing results on public benchmarks, but requires relatively long execution time due to the per-proposal computation and its deep stem network. Other frameworks rely mostly on pixel-features generated by a single FCN forward pass, and employ post-processing like graphical models, template matching, or pixel embedding to cluster pixels belonging to the same instance. More specifically, Non-local Networks [36] utilizes a self-attention [34] mechanism to enable a pixel-feature to perceive features from all the other positions, while the CCNet [13] harvests the contextual information from all pixels more efficiently by stacking two criss-cross attention modules, which augments the feature representation a lot. In post-processing step, Liu et al. [22] present a pixel affinity scheme and cluster pixels into instances with a simple yet effective graph merge algorithm. Instance-Cut [16] and the work of [41] predict object boundaries intentionally to facilitate the separation of object instances.

Our method, SAST, employs a FCN-based framework to predict TCL, TCO, TVO, and TBO maps in parallel. With the efficient CAB and point-to-quad assignment, where the high-level object knowledge is combined with low-level pixel information, SAST can detect text of arbitrary shapes with high accuracy and efficiency.

3 METHODOLOGY

In this section, we will describe our SAST framework for detecting scene text of arbitrary shapes in details.

3.1 Arbitrary Shape Representation

The bounding boxes, rotated rectangles, and quadrilaterals are used as classical representations in most detection-based text detectors, which fails to precisely describe the text instances of arbitrary shapes, as shown in Fig. 1 (b). The segmentation-based methods formulate the detection of arbitrarily-shaped text as a binary segmentation problem. Most of them directly extracted the contours of instance mask as the representation of text, which is easily affected by the completeness and consistency of segmentation. However,

PSENet [35] and TextSnake [23] attempted to progressively reconstruct the polygonal representation of detected text based on a shrunk text region, of which the post-processing is complex and tended to be slow. Inspired by those efforts, we aim to design an effective method for arbitrarily-shaped text representation.

In this paper, we extract the center line of text region (TCL map) and reconstruct the precise shape representation of text instances with a regressed geometry property, i.e. TBO, which indicates the offset between each pixel in TCL map and corresponding point pair in upper and lower edge of its text region. More specifically, as depicted in Fig. 2, the representation strategy consists of two steps: text center point sampling and border point extraction. Firstly, we sample n points at equidistance intervals from left to right on the center line region of text instance. By taking a further operation, we can determine the corresponding border point pairs based on the sampled center line point with the information provided by TBO maps in the same location. By linking all the border points clockwise, we can obtain a complete text polygon representation. Instead of setting n to a fixed number, we assign it by the ratio of center line length to the average of length of border offset pairs adaptively. Several experiments on curved text datasets prove that our method is efficient and flexible for arbitrarily-shaped text instances.

3.2 Pipeline

The network architecture of FCN-based text detectors are limited to the local receptive fields and short-range contextual information, and makes it struggling to segment some challenging text instances. Thus, we design a **Context Attention Block** to integrate the long-range dependencies of pixels to obtain a more representative feature. As a substitute for the Connected Component Analysis, we also propose **Point-to-Quad Assignment** to cluster the pixels in TCL map into text instances, where we use TCL and TVO maps to restore the minimum quadrilateral bounding boxes of text instances as high-level information.

An overview of our framework is depicted in Fig. 3. It consists of three parts, including a stem network, multi-task branches, and a post-processing part. The stem network is based on ResNet-50 [8] with FPN [19] and CABs to produce context-enhanced representation. The TCL, TCO, TVO, and TBO maps are predicted for each text region as a multi-task problem. In the post-processing, we segment text instances by point-to-quad assignment. Concretely, similar to EAST [47], the TVO map regresses the four vertices of bounding quadrangle of text region directly, and the detection results is considered as high-level object knowledge. For each pixel in TCL map, a corresponding offset vector from TCO map will point to a low-level center which the pixel belongs to. Computing the distance between lower-level center and high-level object centers of the detected bounding quadrangle, pixels in the TCL map will be grouped into several text instances. In contrast to the connected component analysis, it takes high-level object knowledge into account, and is proved to be more efficient. More details about the mechanism of point-to-quad assignment will be discussed in this Section 3.4. We sample a adaptive number of points in the center line of each text instance, calculate corresponding points in upper and lower borders with the help of TBO map, and reconstruct the representation of arbitrarily-shaped scene text finally.

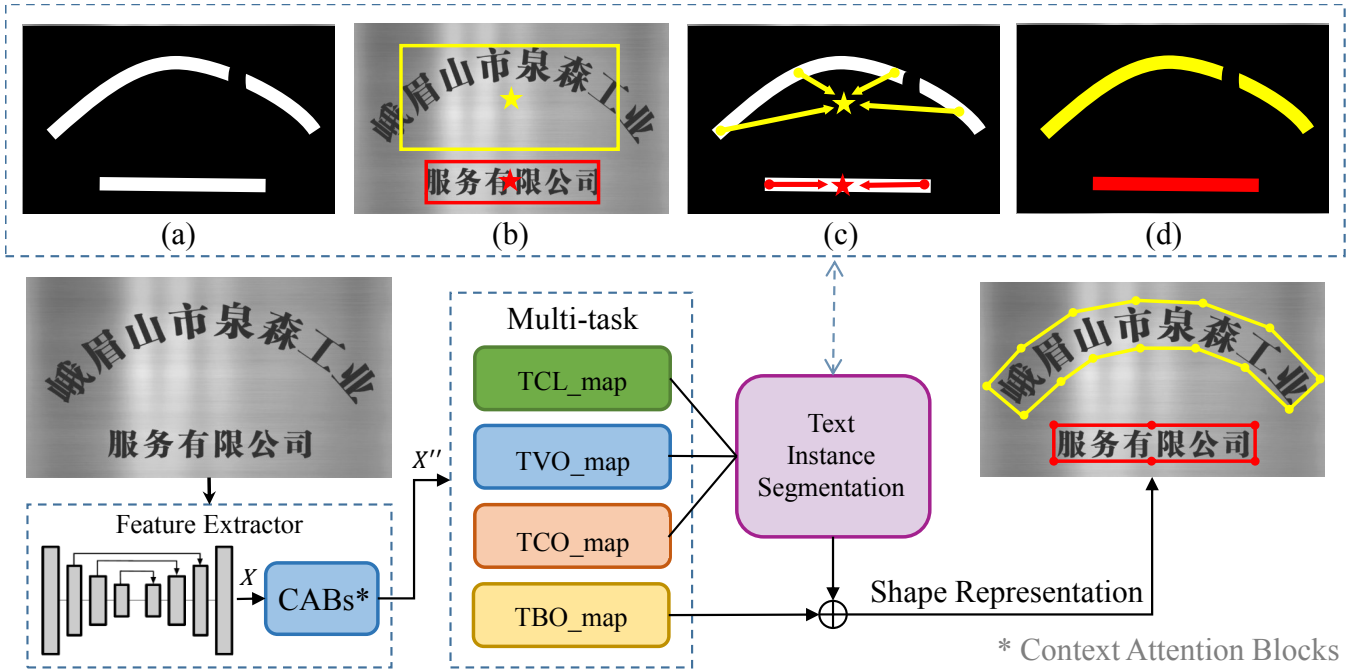


Figure 3: The pipeline of proposed method: 1) Extract feature from input image, and learn TCL, TBO, TCO, TVO maps as a multi-task problem; 2) Achieve instance segmentation by Text Instance Segmentation Module, and the mechanism of point-to-quadrangle assignment is illustrated in c; 3) Restore polygonal representation of text instances of arbitrary shapes.

3.3 Network Architecture

In this paper, we employ ResNet-50 as the backbone network with the additional fully-connected layers removed. With different levels of feature map from the stem network gradually merged three-times in the FPN manner, a fused feature map X is produced at $1/4$ size of the input images. We serially stack two CABs behind to capture rich contextual information. Adding four branches behind the context-enhanced feature maps X'' , the TCL and other geometric maps are predicted in parallel, where we adopt a 1×1 convolution layer with the number of output channel set to $\{1, 2, 8, 4\}$ for TCL, TCO, TVO, and TBO map respectively. It is worth mentioning that all the output channels of convolution layers in the FPN is set to 128 directly, regardless of whether the kernel size is 1 or 3.

Context Attention Block. The segmentation results of FCN and the post-processing steps depend mainly on local information. The proposed CAB utilizes a self-attention mechanism [34] to aggregate the contextual information to augment the feature representation, of which the details is demonstrated in Fig. 4. In order to alleviate the huge computational overhead caused by direct use of self-attention, CAB only considers the similarity between each location in feature map and others in the same horizontal or vertical column. The feature map X is the output of ResNet-50 backbone which is in size of $N \times H \times W \times C$. To collect contextual information horizontally, we adopt three convolution layers behind X in parallel to get $\{f_\theta, f_\phi, f_g\}$ and reshape them into $\{N \times H\} \times W \times C$, then multiply f_ϕ by the transpose of f_θ to get an attention map of size $\{N \times H\} \times W \times W$, which is activated by a Sigmoid function. A horizontal contextual information enhanced

feature, which is resized to $N \times H \times W \times C$ finally, is integrated by multiplication of f_g and the attention map. It is slightly different to get the vertical contextual information that $\{f_\theta, f_\phi, f_g\}$ is transposed to $\{N \times H\} \times C \times W$ at the beginning, as shown in cyan boxes in Fig. 4. Meanwhile, a short-cut path is used to preserve local features. Concatenating the horizontal contextual map, vertical contextual map, and short-cut map together and reducing channel number of X' with a 1×1 convolutional layer, the CAB aggregates long-range pixel-wise contextual information in both horizontal and vertical directions. Besides, the convolutional layers denoted by purple and cyan boxes share weights. By serially connecting two CABs, each pixel can finally capture long-range dependencies from all pixels, as depicted in the bottom of Fig. 4, leading to a more powerful context-enhanced feature map X'' , which also helps alleviate the problems caused by the limited receptive field when dealing with more challenging text instances, such as long text.

3.4 Text Instance Segmentation

For most proposal free text detector of arbitrary shapes, the morphological post-processing such as connected component analysis are adopted to achieve text instance segmentation, which do not explicitly incorporate high-level object knowledge and easily fail to detect complex scene text. In this section, we describe how to generate a text instance semantic segmentation with TCL, TCO and TVO maps with high-level object information.

Point-to-Quad Assignment. As depicted in Fig. 3, the first step of text instance segmentation is detecting candidate text quadrangles based on TCL and TVO maps. Similar to EAST [47], we binarize

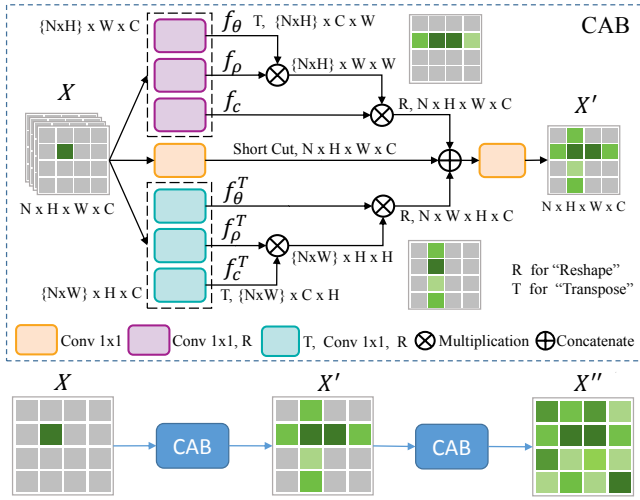


Figure 4: Context Attention Blocks: a single CAB module aggregates pixel-wise contextual information both horizontally and vertically, and long-range dependencies from all pixels can be captured by serially connecting two CABs.

the TCL map, whose pixel values are in the range of $[0, 1]$, with a given threshold, and restore the corresponding quadrangle bounding boxes with the four vertex offsets provided by TVO map. Of course, NMS is adopted to suppress overlapping candidates. The final quadrangle candidates shown in Fig. 3 (b) can be considered to depend on high-level knowledge. The second and last step in text instance segmentation is clustering the responses of text region in the binarized TCL map into text instances. As Fig. 3 (c) shows, the TCO map is a pixel-wise prediction of offset vectors pointing to the center of bounding boxes which the pixels in the TCL map should belong to. With a strong assumption that pixels in TCL map belonging to the same text instance should point to the same object-level center, we cluster TCL map into several text instances by assigning the response pixel to the quadrangle boxes generated in the first step. Moreover, we do not care about whether predicted boxes in the first step are fully bounding text region in the input image, and the pixels outside of the predicted box will be mostly assigned to the corresponding text instances. Integrated with high-level object knowledge and low-level pixel information, the proposed post-processing clusters each pixel in TCL map to its best matching text instance efficiently, and can help to not only separate text instances that are close to each other, but also alleviate fragments when dealing with extremely long text.

3.5 Label Generation and Training Objectives

In this part, the generation of TCL, TCO, TVO, and TBO maps will be discussed. TCL is the shrunk version of text region, and it is an one channel segmentation map for text/non-text. The other label maps such as TCO, TVO, and TBO, are per-pixel offsets with reference to those pixels in TCL map. For each text instance, we calculate the center and four vertices of the minimum enclosing quadrangle from its annotation polygon, as depicted in Fig. 5 (c) (d). TCO map is the offset between pixels in TCL map and the

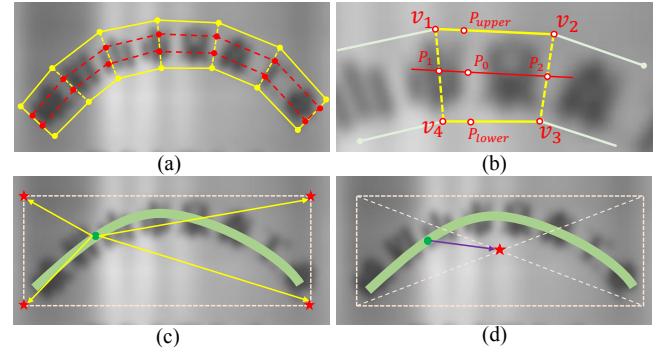


Figure 5: Label Generation: (a) Text center region of a curved text is annotated in red; (b) The generation of TBO map; The four vertices (red stars in c) and center point (red star in d) of bounding box, to which the TVO and TCO map refer.

center of bounding box, while TVO is the offset between the four vertices of bounding box and pixels in TCL map. Hence, the channel numbers of TCO and TVO maps are 2 and 8, respectively, because each point pair requires two channels to represent the offsets $\{\Delta x_i, \Delta y_i\}$. Meanwhile, TBO determines the upper and lower boundaries of text instances in TCL map, thus it is a four channel offset map.

Here are more details about the generation of a TBO map. For a quadrangle text annotation with vertices $\{V_1, V_2, V_3, V_4\}$ in clockwise and V_1 is the top left vertex, as shown in Fig. 5 (b), the generation of TBO mainly contains two steps: first, we find a corresponding point pair on the top and bottom boundaries for each point in TCL map, then calculate the corresponding offset pair. With average slope of the upper side and lower side of quadrangle, the line cross a point P_0 in TCL map can be determined. And it is easy to directly calculate the intersection points $\{P_1, P_2\}$ of the line in the left and right edges of bounding quadrangle with algebraic methods. A pair of corresponding points $\{P_{upper}, P_{lower}\}$ for P_0 can be determined from:

$$\frac{P_0 - P_1}{P_2 - P_1} = \frac{P_{upper} - V_1}{V_2 - V_1} = \frac{P_{lower} - V_4}{V_3 - V_4}.$$

In the second step, the offsets between P_0 and $\{P_{upper}, P_{lower}\}$ can be easily determined. Polygons of more than four vertices are treated as a series of quadrangles connected together, and TBO of polygons can be generated gradually from quadrangles as described before. For non-TCL pixels, their corresponding geometry attributes are set to 0 for convenience.

At the stage of training, the whole network is trained in an end-to-end manner, and the loss of the model can be formulated as:

$$L_{total} = \lambda_1 L_{tcl} + \lambda_2 L_{tco} + \lambda_3 L_{tvo} + \lambda_4 L_{tbo},$$

where L_{tcl} , L_{tco} , L_{tvo} and L_{tbo} represent the loss of TCL, TCO, TVO, and TBO maps, and the first one is the binary segmentation loss while the other are regression loss.

We train segmentation branch by minimizing the Dice loss [27], and the Smooth L_1 loss [5] is adopted for regression loss. The loss weights λ_1 , λ_2 , λ_3 , and λ_4 are a tradeoff between four tasks which are equally important in this work, so we determine a set of values $\{1.0, 0.5, 0.5, 1.0\}$ by making the four loss gradient norms close in back-propagation.

4 EXPERIMENTS

To compare the effectiveness of SAST with existing methods, we perform thorough experiments on four public text detection datasets, i.e., ICDAR 2015, ICDAR2017-MLT, SCUT-CTW1500 and Total-Text.

4.1 Datasets

The datasets used for the experiments in this paper are briefly introduced below.

SynthText. The SynthText dataset [6] is composed of 800,000 natural images, on which text in random colors, fonts, scales, and orientations is rendered carefully to have a realistic look. We use the dataset with word-level labels to pre-train our model.

ICDAR 2015. The ICDAR 2015 dataset [15] is collected for the ICDAR 2015 Robust Reading Competition, with 1,000 natural images for training and 500 for testing. The images are acquired using Google Glass and the text accidentally appear in the scene. All the text instances annotated with word-level quadrangles.

ICDAR2017-MLT. The ICDAR2017-MLT [28] is a large scale multi-lingual text dataset, which includes 7,200 training images, 1,800 validation images and 9,000 test images. The dataset consists of multi-oriented and multi-lingual aspects of scene text. The text regions in ICDAR2017-MLT are also annotated by quadrangles.

SCUT-CTW1500. The SCUT-CTW1500 [42] is a challenging dataset for curved text detection. It consists of 1,000 training images and 500 test images, and text instances are largely in English and Chinese. Different from traditional datasets, the text instances in SCUT-CTW1500 are labelled by polygons with 14 vertices.

Total-Text. The Total-Text [1] is another curved text benchmark, which consists of 1,255 training images and 300 testing images with more than 3 different text orientations: Horizontal, Multi-Oriented, and Curved. The annotations are labelled in word-level.

Evaluation Metrics. The performance on ICDAR2015, Total-Text, SCUT-CTW1500, and ICDAR2017-MLT is evaluated using the protocols provided in [1, 15, 28, 42], respectively.

4.2 Implementation Details

Training. ResNet-50 is used as the network backbone with pre-trained weight on ImageNet [3]. The skip-connection is in FPN fashion with output channel numbers of the convolutional layers set to 128 and the final output is at 1/4 size of input images. All upsample operators are the bilinear interpolation and the classification branch is activated with sigmoid while the regression branches, i.e. TCO, TVO, and TBO maps, is the output of the last convolution layer directly. The training process is divided into two steps, i.e., the warming-up and fine-tuning steps. In the warming-up step, we apply Adam optimizer to train our model with learning rate $1e-4$, and the learning rate decay factor is 0.94 on the SynthText dataset. In the fine-tuning step, the learning rate is re-initiated to $1e-4$ and the model is tuned on ICDAR 2015, ICDAR2017-MLT, SCUT-CTW1500 and Total-Text.

All the experiments are performed on a workstation with the following configuration, CPU: Intel(R) Xeon(R) CPU E5-2620 v2 @ 2.10GHz x16; GPU: NVIDIA TITAN Xp x4; RAM: 64GB. During the training time, we set the batch size to 8 per GPU in parallel.

Data Augmentation. We randomly crop the text image regions, then resize and pad them to 512×512 . Specially for curved polygon

labeled datasets, we crop images without crossing text instances to avoid the destruction of polygon annotations. The cropped image regions will be rotated randomly in 4 directions (0° , 90° , 180° , and 270°) and standardized by subtracting the RGB mean value of ImageNet dataset. The text region, which is marked as "DO NOT CARE" or its minimum length of edges is less than 8 pixels, will be ignored in the training process.

Testing. In inference phase, unless otherwise stated, we set the longer side to 1536 for single-scale testing, and to 512, 768, 1536, and 2048 for multi-scale testing, while keeping the aspect ratio unchanged. A specified range is assigned to each testing scale and detections from different scales are combined using NMS, which is inspired by SNIP[31].

4.3 Ablation Study

We conduct several ablation experiments to analyze SAST. The details are discussed as follows.

The Effectiveness of TBO, TCO and TVO. To verify the efficiency of Text Instance Segmentation Module (TVO and TCO maps) and Arbitrary Shape Representation Module (TBO map) in SAST, we conduct several experiments with the following configurations: 1) TCL + CC + Expand: It is a naive way to predict center region of text, use connected component analysis to achieve text instance segmentation, and expand the contours of connected components by a shrinking rate as the final text geometric representation. 2) TCL + CC + TBO: Instead of expanding the contours directly, we reconstruct the precise polygon of a text instance with Arbitrary Shape Representation Module. 3) TCL + TVO + TCO + TBO: As a substitute for connected component analysis, we use the method of point-to-quad assignment in Text Instance Segmentation Module, which is supposed to incorporate high-level object knowledge and low-level information, and assign each pixel on the TCL map to its best matching text instances. The efficiency of the proposed method is demonstrated on SCUT-CTW1500, as shown in Tab. 1. It surpasses the first two methods by 21.75% and 1.46% in Hmean, respectively. Meanwhile, the proposed point-to-quad assignment cost almost the same time as connected component analysis.

The Trade-off between Speed and Accuracy. There is a trade-off between speed and accuracy, and the mainstream segmentation methods maintain high resolution, which is usually in the same size of input image, to achieve a better result at a correspondingly high cost in time. Several experiments are made on SCUT-CTW1500 benchmark. We compare the performance with different resolution of output, i.e., $\{1, 1/2, 1/4, 1/8\}$, and find a rational trade-off between speed and accuracy at the 1/4 scale of input images. The detail configuration and results are shown in Tab. 2. Note that the feature extractor in those experiments is not equipped with Context Attention Block.

The Effectiveness of Context Attention Blocks. We introduce the CABs into the network architecture to capture long-range dependencies of pixel information. We conduct two experiments on SCUT-CTW1500 by replacing the CABs with several convolutional layers stacked together as the baseline experiment, which has almost the same number of trainable variables. The input size of image is 512×512 , and the output of images is at 1/4 of input

Table 1: Ablation study for the effectiveness of TBO, TCO, and TVO in the proposed method.

Method	Recall	Precision	Hmean	T (ms)
TCL + CC + expand	55.51	61.55	58.37	–
TCL + CC + TBO	74.65	83.13	78.66	5.84
TCL + TVO + TCO + TBO	76.69	83.89	80.12	6.26

Table 2: Ablation study for the trade-off between speed and accuracy on SCUT-CTW1500.

Method	Recall	Precision	Hmean	FPS
1s	76.34	86.23	80.98	10.54
2s	70.86	86.30	77.82	21.68
4s	76.69	83.86	80.12	30.21
8s	71.54	80.67	75.83	38.05

Table 3: Ablation study for the effectiveness of CABs on SCUT-CTW1500.

Method	Recall	Precision	Hmean	FPS
baseline	76.69	83.86	80.12	30.21
with CABs	77.05	85.31	80.97	27.63

size. Tab. 3 demonstrates the performance and speed of both experiments. The experiment with CABs achieves 80.97% in Hmean at a speed of 27.63 FPS, which exceeds the baseline by 0.85% in Hmean at a bit slower frame rate.

4.4 Evaluation on Curved Text Benchmark

On SCUT-CTW1500 and Total-Text, we evaluate the performance of SAST for detecting text lines of arbitrary shapes. We fine-tune our model for about 10 epochs on SCUT-CTW1500 and Total-Text training set, respectively. In testing phase, the number of vertices of text polygons is adaptively counted and we set the scale of the longer side to 512 for single-scale testing on both datasets.

The quantitative results are shown in Tab. 4 and Tab. 5. With the help of the efficient post-processing, SAST achieves 80.97% and 78.08% in Hmean on SCUT-CTW1500 and Total-Text, respectively, which is comparable to the state-of-the-art methods. In addition, multi-scale testing can further improves Hmean to 81.45% and 80.21% on SCUT-CTW1500 and Total-Text. The visualization of curved text detection are shown in Fig. 6 (a) and (b). As can be seen, the proposed text detector SAST can handle curved text lines well.

4.5 Evaluation on ICDAR 2015

In order to verify the validity for detecting oriented text, we compare SAST with the state-of-the-art methods on ICDAR 2015 dataset, a standard oriented text dataset. Compared with previous arbitrarily-shaped text detectors [35, 38, 39], which detect text on the same size as input image, SAST achieves a better performance in a much faster speed. All the results are listed in Tab. 6. Specifically, for single-scale testing, SAST achieves 86.91% in Hmean, surpassing

Table 4: Evaluation on SCUT-CTW1500 for detecting text lines of arbitrary shapes.

Method	Recall	Precision	Hmean	FPS
CTPN [32]	53.80	60.40	56.90	–
EAST [47]	49.10	78.70	60.40	–
DMPNet [21]	56.00	69.90	62.20	–
CTD [42]	65.20	74.30	69.50	15.20
CTD + TLOC [42]	69.80	74.30	73.40	13.30
SLPR [48]	70.10	80.10	74.80	–
TextSnake [23]	85.30	67.90	75.60	12.07
PSENet-4s [35]	78.13	85.49	79.29	8.40
PSENet-2s [35]	79.30	81.95	80.60	–
PSENet-1s [35]	79.89	82.50	81.17	3.90
TextField [38]	79.80	83.00	81.40	–
SAST	77.05	85.31	80.97	27.63
SAST MS	81.71	81.19	81.45	–

Table 5: Evaluation on Total-Text for detecting text lines of arbitrary shapes.

Method	Recall	Precision	Hmean
SegLink [30]	23.80	30.30	26.70
DeconvNet [1]	33.00	40.00	36.00
EAST [47]	36.20	50.00	42.00
TextSnake [23]	74.50	82.70	78.40
TextField [38]	79.90	81.20	80.60
SAST	76.86	83.77	80.17
SAST MS	75.49	85.57	80.21

most competitors (these pure detection methods without the assistance of recognition task). Moreover, multi-scale testing increases about 0.53% in Hmean. Some detection results are shown in Fig. 6 (c), and indicate that SAST is also capable to detect multi-oriented text accurately.

4.6 Evaluation on ICDAR2017-MLT

To demonstrate the generalization ability of SAST on multilingual scene text detection, we evaluate SAST on ICDAR2017-MLT. Similar to the above training methods, the detector is fine-tuned for about 10 epochs on the SynthText pre-trained model. At the single scale testing, our proposed method achieves a Hmean of 68.76%, and it increases to 72.37% for multi-scale testing. The quantitative results are shown in Tab. 7. The visualization of multilingual text detection is as illustrated in the Fig. 6 (d), which shows the robustness of the proposed method in detecting multilingual scene text.

4.7 Runtime

In this paper, we make a trade-off between speed and accuracy. The TCL, TVO, TCO, and TBO maps are predicted in the 1/4 size of input images. With the proposed post-processing step, SAST is supposed to detect text of arbitrary shapes in real-time speed with a commonly used GPU. To demonstrate the runtime of the proposed



Figure 6: Some qualitative results by the proposed method. From left to right: ICDAR2015, SCUT-CTW1500, Total-Text, and ICDAR17-MLT. Blue contours: ground truths; Cyan contours: quads from TVO map; Red contours: final detection results.

Table 6: Evaluation on ICDAR 2015 for detecting oriented text.

Method	Recall	Precision	Hmean
DMPNet [21]	68.22	73.23	70.64
SegLink [30]	76.50	74.74	75.61
SSTD [9]	73.86	80.23	76.91
WordSup [11]	77.03	79.33	78.16
RRPN [26]	77.13	83.52	80.20
EAST [47]	78.33	83.27	80.72
He et al. [10]	80.00	82.00	81.00
TextField [38]	80.50	84.30	82.40
TextSnake [23]	80.40	84.90	82.60
PixelLink [2]	82.00	85.50	83.70
RRD [18]	80.00	88.00	83.80
Lyu et al. [25]	79.70	89.50	84.30
PSENet-4s [35]	83.87	87.98	85.88
IncepText [39]	84.30	89.40	86.80
PSENet-1s [35]	85.51	88.71	87.08
PSENet-2s [35]	85.22	89.30	87.21
SAST	87.09	86.72	86.91
SAST MS	87.34	87.55	87.44

Table 7: Evaluation on ICDAR2017-MLT for the generalization ability of SAST on multilingual scene text detection.

Method	Recall	Precision	Hmean
Lyu et al. [25]	56.60	83.80	66.80
AF-RPN [46]	66.00	75.00	70.00
PSENet-4s [35]	67.56	75.98	71.52
PSENet-2s [35]	68.35	76.97	72.40
Lyu et al. MS [25]	70.60	74.30	72.40
PSENet-1s [35]	68.40	77.01	72.45
SAST	67.56	70.00	68.76
SAST MS	66.53	79.35	72.37

method, we run testing on SCUT-CTW1500 with a workstation equipped with NVIDIA TITAN Xp. The test image is resized to 512×512 and the batch size is set to 1 on a single GPU. It takes 29.58 ms and 6.61 ms in the process of network inference and post-processing respectively, which is written in Python code¹ and can be further optimized. It runs at 27.63 FPS with a Hmean of 80.97%, surpassing most of the existing arbitrarily-shaped text detectors in both accuracy and efficiency², as depicted in Tab. 4.

5 CONCLUSION AND FUTURE WORK

In this paper, we propose an efficient single-shot arbitrarily-shaped text detector together with Context Attention Blocks and a mechanism of point-to-quad assignment, which integrates both high-level object knowledge and low-level pixel information to obtain text instances from a context-enhanced segmentation. Several experiments demonstrate that the proposed SAST is effective in detecting arbitrarily-shaped text, and is also robust in generalizing to multilingual scene text datasets. Qualitative results show that SAST helps to alleviate some common challenges in segmentation-based text detector, such as the problem of fragments and the separation of adjacent text instances. Moreover, with a commonly used GPU, SAST runs fast and may be sufficient for some real-time applications, e.g., augmented reality translation. However, it is difficult for SAST to detect some extreme cases, which mainly are very small text regions. In the future, we are interested in improving the ability of small text detection and developing an end-to-end text reading system for text of arbitrary shapes.

ACKNOWLEDGMENTS

This work is supported in part by the National Natural Science Foundation of China (NSFC) under Grant 61572387, Grant 61632019, Grant 61836008, and Grant 61672404, and the Foundation for Innovative Research Groups of the NSFC under Grant 61621005.

¹The NMS part is written in C++.

²The speed of different methods is depicted for reference only, which might be evaluated with different hardware environments.

REFERENCES

- [1] Chee Kheng Ch'ng and Chee Seng Chan. 2017. Total-Text: A comprehensive dataset for scene text detection and recognition. In *Int. Conf. Doc. Anal. Recognit. (ICDAR)*, Vol. 1. IEEE, 935–942.
- [2] Dan Deng, Haifeng Liu, Xuelong Li, and Deng Cai. 2018. PixelLink: Detecting scene text via instance segmentation. In *Proc. AAAI Conf. Artif. Intell. (AAAI)*.
- [3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. In *IEEE Conf. Comp. Vis. Patt. Recognit. (CVPR)*. IEEE, 248–255.
- [4] Alireza Fathi, Zbigniew Wojna, Vivek Rathod, Peng Wang, Hyun Oh Song, Sergio Guadarrama, and Kevin P Murphy. 2017. Semantic instance segmentation via deep metric learning. arXiv:1703.10277
- [5] R. Girshick. 2015. Fast R-CNN. In *IEEE Int. Conf. Comp. Vis. (ICCV)*. 1440–1448.
- [6] Ankush Gupta, Andrea Vedaldi, and Andrew Zisserman. 2016. Synthetic data for text localisation in natural images. In *IEEE Conf. Comp. Vis. Patt. Recognit. (CVPR)*. 2315–2324.
- [7] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. 2017. Mask R-CNN. In *IEEE Conf. Comp. Vis. Patt. Recognit. (CVPR)*. 2961–2969.
- [8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *IEEE Conf. Comp. Vis. Patt. Recognit. (CVPR)*. 770–778.
- [9] Pan He, Weilin Huang, Tong He, Qile Zhu, Yu Qiao, and Xiaolin Li. 2017. Single shot text detector with regional attention. In *IEEE Conf. Comp. Vis. Patt. Recognit. (CVPR)*. 3047–3055.
- [10] Wenhao He, Xu-Yao Zhang, Fei Yin, and Cheng-Lin Liu. 2017. Deep direct regression for multi-oriented scene text detection. In *IEEE Conf. Comp. Vis. Patt. Recognit. (CVPR)*. 745–753.
- [11] Han Hu, Chengquan Zhang, Yuxuan Luo, Yuzhuo Wang, Junyu Han, and Errui Ding. 2017. WordSup: Exploiting Word Annotations for Character Based Text Detection. In *IEEE Int. Conf. Comp. Vis. (ICCV)*. 4950–4959.
- [12] Lichao Huang, Yi Yang, Yafeng Deng, and Yinan Yu. 2015. DenseBox: Unifying landmark localization with end to end object detection. arXiv:1509.04874
- [13] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. 2018. CCNet: Criss-cross attention for semantic segmentation. arXiv:1811.11721
- [14] Zhida Huang, Zhuoyao Zhong, Lei Sun, and Qiang Huo. 2019. Mask R-CNN with pyramid attention network for scene text detection. In *Winter Conf. Appl. Comp. Vis. (WACV)*. IEEE, 764–772.
- [15] Dimosthenis Karatzas, Lluís Gomez-Bigorda, Angelos Nicolaou, Suman Ghosh, Andrew Bagdanov, Masakazu Iwamura, Jiri Matas, Lukas Neumann, Vijay Ramaseshan Chandrasekhar, Shijian Lu, et al. 2015. ICDAR 2015 competition on robust reading. In *Int. Conf. Doc. Anal. Recognit. (ICDAR)*. IEEE, 1156–1160.
- [16] Alexander Kirillov, Evgeny Levinkov, Bjoern Andres, Bogdan Savchynskyy, and Carsten Rother. 2017. InstanceCut: from edges to instances with multicut. In *IEEE Conf. Comp. Vis. Patt. Recognit. (CVPR)*. IEEE, 7322–7331.
- [17] Minghui Liao, Baoguang Shi, Xiang Bai, Xinggang Wang, and Wenyu Liu. 2017. TextBoxes: A fast text detector with a single deep neural network. In *Proc. AAAI Conf. Artif. Intell. (AAAI)*. 4161–4167.
- [18] Minghui Liao, Zhen Zhu, Baoguang Shi, Gui-song Xia, and Xiang Bai. 2018. Rotation-sensitive regression for oriented scene text detection. In *IEEE Conf. Comp. Vis. Patt. Recognit. (CVPR)*. 5909–5918.
- [19] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. 2017. Feature pyramid networks for object detection. In *IEEE Conf. Comp. Vis. Patt. Recognit. (CVPR)*. 2117–2125.
- [20] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. 2016. SSD: Single shot multibox detector. In *Eur. Conf. Comp. Vis. (ECCV)*. Springer, 21–37.
- [21] Yuliang Liu and Lianwen Jin. 2017. Deep matching prior network: Toward tighter multi-oriented text detection. In *IEEE Conf. Comp. Vis. Patt. Recognit. (CVPR)*. 1962–1969.
- [22] Yiding Liu, Siyu Yang, Bin Li, Wengang Zhou, Jizheng Xu, Houqiang Li, and Yan Lu. 2018. Affinity derivation and graph merge for instance segmentation. In *Eur. Conf. Comp. Vis. (ECCV)*. 686–703.
- [23] Shangbang Long, Jiaqiang Ruan, Wenjie Zhang, Xin He, Wenhao Wu, and Cong Yao. 2018. TextSnake: A flexible representation for detecting text of arbitrary shapes. In *Eur. Conf. Comp. Vis. (ECCV)*. 20–36.
- [24] Pengyuan Lyu, Minghui Liao, Cong Yao, Wenhao Wu, and Xiang Bai. 2018. Mask TextSpotter: An end-to-end trainable neural network for spotting text with arbitrary shapes. In *Eur. Conf. Comp. Vis. (ECCV)*. 67–83.
- [25] Pengyuan Lyu, Cong Yao, Wenhao Wu, Shuicheng Yan, and Xiang Bai. 2018. Multi-oriented scene text detection via corner localization and region segmentation. In *IEEE Conf. Comp. Vis. Patt. Recognit. (CVPR)*. 7553–7563.
- [26] Jianqi Ma, Weiyuan Shao, Hao Ye, Li Wang, Hong Wang, Yingbin Zheng, and Xiangyang Xue. 2018. Arbitrary-oriented scene text detection via rotation proposals. *IEEE Trans. Multimedia* 20, 11 (2018), 3111–3122.
- [27] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. 2016. V-Net: Fully convolutional neural networks for volumetric medical image segmentation. In *4th Int. Conf. 3D Vision (3DV)*. IEEE, 565–571.
- [28] Nibal Nayef, Fei Yin, Imen Bizid, Hyunsoo Choi, Yuan Feng, Dimosthenis Karatzas, Zhenbo Luo, Umapada Pal, Christophe Rigaud, Joseph Chazalon, et al. 2017. ICDAR2017 robust reading challenge on multi-lingual scene text detection and script identification-rrc-mlt. In *Int. Conf. Doc. Anal. Recognit. (ICDAR)*, Vol. 1. IEEE, 1454–1459.
- [29] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Adv. Neural Inf. Process. Syst. (NIPS)*. 91–99.
- [30] Baoguang Shi, Xiang Bai, and Serge Belongie. 2017. Detecting oriented text in natural images by linking segments. In *IEEE Conf. Comp. Vis. Patt. Recognit. (CVPR)*. 2550–2558.
- [31] Bharat Singh and Larry S Davis. 2018. An analysis of scale invariance in object detection snip. In *IEEE Conf. Comp. Vis. Patt. Recognit. (CVPR)*. 3578–3587.
- [32] Zhi Tian, Weilin Huang, Tong He, Pan He, and Yu Qiao. 2016. Detecting text in natural image with connectionist text proposal network. In *Eur. Conf. Comp. Vis. (ECCV)*. Springer, 56–72.
- [33] Jonas Uhrig, Eike Rehder, Björn Fröhlich, Uwe Franke, and Thomas Brox. 2018. Box2Pix: Single-shot instance segmentation by assigning pixels to object boxes. In *IEEE Intell. Veh. Symp. (IV)*. IEEE, 292–299.
- [34] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Adv. Neural Inf. Process. Syst. (NIPS)*. 5998–6008.
- [35] Wenhao Wang, Enze Xie, Xiang Li, Wenbo Hou, Tong Lu, Gang Yu, and Shuai Shao. 2019. Shape Robust Text Detection With Progressive Scale Expansion Network. In *IEEE Conf. Comp. Vis. Patt. Recognit. (CVPR)*. 9336–9345.
- [36] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. 2018. Non-local neural networks. In *IEEE Conf. Comp. Vis. Patt. Recognit. (CVPR)*. 7794–7803.
- [37] Yue Wu and Prem Natarajan. 2017. Self-organized text detection with minimal post-processing via border learning. In *IEEE Conf. Comp. Vis. Patt. Recognit. (CVPR)*. 5000–5009.
- [38] Yongchao Xu, Yukang Wang, Wei Zhou, Yongpan Wang, Zhibo Yang, and Xiang Bai. 2019. TextField: Learning A Deep Direction Field for Irregular Scene Text Detection. *IEEE Trans. Image Process.* (2019). arXiv:1812.01393
- [39] Qiangpeng Yang, Mengli Cheng, Wenmeng Zhou, Yan Chen, Minghui Qiu, and Wei Lin. 2018. IncepText: a new inception-text module with deformable PSROI pooling for multi-oriented scene text detection. In *Int. Joint Conf. Artif. Intell. (IJCAI)*. IJCAI, 1071–1077.
- [40] Qixiang Ye and David Doermann. 2015. Text detection and recognition in imagery: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* 37, 7 (2015), 1480–1500.
- [41] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. 2018. Learning a discriminative feature network for semantic segmentation. In *IEEE Conf. Comp. Vis. Patt. Recognit. (CVPR)*. IEEE, 1857–1866.
- [42] Liu Yuliang, Jin Lianwen, Zhang Shuai, and Zhang Sheng. 2017. Detecting curve text in the wild: New dataset and new solution. arXiv:1712.02170
- [43] Chengquan Zhang, Borong Liang, Zuming Huang, Mengyi En, Junyu Han, Errui Ding, and Xinghao Ding. 2019. Look More Than Once: An Accurate Detector for Text of Arbitrary Shapes. In *IEEE Conf. Comp. Vis. Patt. Recognit. (CVPR)*.
- [44] Zheng Zhang, Chengquan Zhang, Wei Shen, Cong Yao, Wenyu Liu, and Xiang Bai. 2016. Multi-oriented text detection with fully convolutional networks. In *IEEE Conf. Comp. Vis. Patt. Recognit. (CVPR)*. 4159–4167.
- [45] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip HS Torr. 2015. Conditional random fields as recurrent neural networks. In *IEEE Conf. Comp. Vis. Patt. Recognit. (CVPR)*. 1529–1537.
- [46] Zhuoyao Zhong, Lei Sun, and Qiang Huo. 2018. An Anchor-Free Region Proposal Network for Faster R-CNN based Text Detection Approaches. arXiv:1804.09003
- [47] Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, and Jiajun Liang. 2017. EAST: An efficient and accurate scene text detector. In *IEEE Conf. Comp. Vis. Patt. Recognit. (CVPR)*. 5551–5560.
- [48] Yixing Zhu and Jun Du. 2018. Sliding line point regression for shape robust scene text detection. In *Int. Conf. Pattern Recognit. (ICPR)*. 3735–3740.
- [49] Yingying Zhu, Cong Yao, and Xiang Bai. 2016. Scene text detection and recognition: Recent advances and future trends. *Frontiers of Computer Science* 10, 1 (2016), 19–36.