Big Data Research 34 (2023) 100410

Contents lists available at ScienceDirect

Big Data Research

journal homepage: www.elsevier.com/locate/bdr

# End-PolarT: Polar Representation for End-to-End Scene Text Detection

Yirui Wu [a,b], Qiran Kong [a], Cheng Qian [c], Michele Nappi [d], Shaohua Wan [e,*]

[a] College of Computer and Information, Hohai University, Nanjing, China
[b] Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun, China
[c] Jiangsu Hydraulic Research Institute, Nanjing, China
[d] University of Salerno, Via Giovanni Paolo II, 132, Fisciano, Salerno 84084, Italy
[e] Shenzhen Institute for Advanced Study, University of Electronic Science and Technology of China, Shenzhen, China

ARTICLE INFO

*Article history:*
Received 6 May 2021
Received in revised form 3 March 2023
Accepted 5 September 2023
Available online 15 September 2023

*Keywords:*
Text detection
Polar representation
Rotated text
Single-stage detection

ABSTRACT

Deep learning has achieved great success in text detection, where recent methods adopt inspirations from segmentation to detect scene texts. However, most segmentation based methods have high computation cost in pixel-level classification and post refinements. Moreover, they still faces challenges like arbitrary directions, curved texts, illumination and so on. Aim to improve detection accuracy and computation cost, we propose an end-to-end and single-stage method named as End-PolarT network by generating contour points in polar coordinates for text detection. End-PolarT not only regress locations of contour points instead of pixels to relieve high computation cost, but also fits with intrinsic characteristics of text instances by centers and contours to suppress mislabeling boundary pixels. To cope with polar representation, we further propose polar IoU and centerness as key parts of loss functions to generate effective paradigms for text detection. Compared with the existing methods, End-PolarT achieves superior results by testing on several public datasets, thus keeping balance between efficiency and effectiveness in complicated scenes.

## 1. Introduction

As part of scene understanding, goal of scene text detection is to spot text regions in natural images. Even though deep learning has made great progress in understanding images and videos [1–3], it's still challenging to detect texts from scene images. Firstly, appearance of text regions and backgrounds in natural scenarios are complicated to distinguish, leading to quantity of miscalculations. Secondly, shapes of text regions usually are arbitrary like curved and rotated texts, resulting in difficulties to accurately detect text regions.

Facing the above difficulties, many methods are proposed to detect texts in the wild, where they can be classified into two categories, regression and segmentation based methods. The former one [4] aims to detect text instances as common objects. Even though they can effectively detect horizontal and vertical texts, they require additional designs to detect rotated texts. Meanwhile, the latter one [5,6] regards text detection as segmentation task to obtain pixel-level masks progressively, which are capable to deal with oriented and curved texts.

Following with the idea to detect arbitrary texts with segmentation based methods, we further classify them into bottom-up and top-down methods based on whether directly generating pixel-level labels. Specifically, bottom-up methods regard text detection as a problem of semantic segmentation by directly assigning pixel-level labels to text or non-text regions. For example, Wang et al. [5] propose a novel Progressive Scale Expansion Network (PSENet), which is a segmentation-based detector with multiple predictions for each text instance. Recently, FCENet [6] represents enveloping curve of texts as parameters of Fourier transform, thus designing to predict text enveloping box of arbitrary shape with Fourier frequency representations. However, bottom-up methods may lead to missed categorizing in boundary pixels, due to cases of overlapped texts.

On the contrary, top-down methods firstly detect rectangular bounding boxes containing texts, and then perform pixel-level label prediction inside boxes. Inspired by Mask R-CNN, Huang et al. [7] present a method to robustly detect multi-oriented and curved text sfrom natural scene images in a unified manner. Afterwards, Wang et al. [8] propose an improved method based on RPN, which could solve the problem of wrong detections caused by scale variations. Most recently, Wu et al. [9] propose Self-Reliant

[*] Corresponding author.
*E-mail addresses:* wuyirui@hhu.edu.cn (Y. Wu), shawn_ji@163.com (Q. Kong), chengqian-research@outlook.com (C. Qian), mnappi@unisa.it (M. Nappi), shaohua.wan@ieee.org (S. Wan).

https://doi.org/10.1016/j.bdr.2023.100410
2214-5796/© 2023 Elsevier Inc. All rights reserved.

Scene Text Spotter, which is a single shot approach by sampling feature points around each potential text instance and performing text detection and recognition simultaneously guided by these points. However, top-down methods generally adopt dense anchors to refine bounding boxes, resulting in quantity of parameters to determine and slow computation speed.

Focusing to solve the problem of high computation cost brought by too many anchors, we propose a single-stage method, i.e., End-PolarT network to directly detect texts by generating centers and contour points of text instances with novel polar coordinates representation. Specifically, End-PolarT not only relieves the computation burden by generating a small subset of key points instead of regressing pixel-level labels, but also adopts efficient representations of texts with center and contours, which coincides with intrinsic characteristics of text instances for better discrimination. To cope with polar representation, we further propose polar IoU and centerness as part of loss functions, thus generalizing effective paradigms to detect bounding boxes under polar representation. Our main contribution could be concluded as follows:

- End-PolarT accurately detects text regions by generating centers and counter points under novel polar coordinates, which not only relieves high computation cost brought by pixel-level classification, but also fits with intrinsic characteristics of text instances with key points.
- Loss function with polar IoU and centerness enables End-PolarT to find patterns of texts under polar representation with a fast convergency speed.
- Bounding box branch is designed in End-PolarT to promote fast convergency, which considers aspect ratios as factors during training iterations.

The rest of the paper is organized as follows. Section 2 gives an overview of the related work on relative aspects. Section 3 analyzes problems of scene text detection. In Section 4, details of the proposed End-PolarT network are discussed, including network overall architecture, polar representation, polar cIoU loss function. Section 5 shows experimental results with several comparative methods, and Section 6 finally concludes the paper. End-PolarT has limitations when dealing with difficult cases of extreme curved texts or blur texts, which requires further improvements in structure design for better performance. Moreover, End-PolarT still suffers from relatively high computation cost with lots of network layers to compute, compared with methods equipped with only several layers.

## 2. Related work

In this section, we give a brief literature review of this paper, including scene text detection and single-stage object detector.

### 2.1. Scene text detection

Existing scene text detection methods can be roughly divided into two categories, namely regression and segmentation based methods. Inspired by object detectors SSD, TextBoxes [10] directly predict anchor scales and shape to handle texts with extreme aspect ratios. Afterwards, TextBoxes++ [11] propose a novel loss function to detect arbitrary texts by regressing quadrangles instead of horizontal bounding boxes. Later, RRD [12] adopt rotation-invariant and sensitive features for text classification and regression, thus improving detection accuracy for long texts from two separate running branches. To deal with tiny texts, SSRD [13] further generate text attention map to enhance text related feature map, thus better suppressing background information. Based on Faster R-CNN, Ma et al. [14] adopt label distribution learning to promote label ambiguity process of text annotation, thus achieving good performance without additional burden of post-processing. However, regression based methods generally fail to detect arbitrary texts, due to their initial ideas to directly regress anchors of bounding boxes in a fast manner.

To deal with texts of arbitrary shapes, segmentation methods can be classified as bottom-up and top-down methods. Bottom-up methods directly assign pixel-level labels to text and non-text regions. For example, TextSnake [15] use ordered disks and text center lines to represent text instances with arbitrary shapes. Later, PSENet [5] use FCN to predict pixel-level labels of text instances in multiple scales in a progressive manner. In fact, it's still challenging to accurately segment text instance by grouping pixels into regions. Most recently, Long et al. [16] introduce a unified detector, which can detect text entities and group them for layout analysis in an end-to-end manner. They also offer the first dataset that includes hierarchical annotations of text in both natural scenes and documents.

Top-down methods try to first accurately locate bounding boxes containing texts, and then predict pixel-level labels inside boxes. Early, Mask R-CNN [17] is designed with a novel step of ROI pooling on the basis of faster R-CNN, which further adopts a mask generation module for accurate segmentation of text instances. Owing to the guidance of semantic information, SPCNet [18] involves context information, leading to stronger detection capabilities in complex natural scenes. Regarding the issue of better feature map for detection, TextFuseNet [19] obtains richer text features by fusing three different categories of features, i.e., character-level, word-level and global-level, where rich features enhance detection capability and environmental adaptability. Afterwards, ContourNet [8] first generates more accurate anchors through Adaptive-RPN, and then uses Local Orthogonal Texture-aware Module to describe local texture information with two orthogonal directions. Recently, AE TextSpotter [20] incorporates linguistic knowledge into text detection by learning linguistic representation to reduce ambiguous proposals. However, top-down methods generally suffer from large amount of computations, due to dense generated predefined anchors. End-PolarT builds on the basis of top-down methods, which not only relieves high computation cost brought by pixel-level classification, but also fits with intrinsic characteristics of text instances by generating centers and counter points under polar coordinates.

### 2.2. Single stage object detector

To relieve high computation burden brought by multiple stage detector, researchers develop fast and accurate single stage object detector, which directly generates detections in one stage to predict bounding boxes and masks simultaneously [21].

Early, most single-stage methods generate rectangular boxes for detection. For example, YOLO [22] is short for You Only Look once, which divides an image into multiple grids and each grid is responsible to predict objects whose center are located in that gird. Later, FCOS [23] adopts a divide and conquer strategy, where different scales of feature maps are responsible for different sizes of boxes. However, these methods generally fail to predict arbitrary bounding boxes in one-stage.

Unlike former methods to detect rectangle objects, deep Watershed Transform [24] first uses fully convolutional network to predict energy map of the entire image, and then adopts the watershed algorithm to generate connected object instances based on the energy map. Later, YOLACT [25] introduce prototype masks that don't depend on any individual instances, where the resulting instances are generated by the linear combination of these prototype masks in real time. Afterwards, TensorMask [26] investigates the paradigm of dense sliding window instance segmentation, which outputs a geometric structure with its own spatial dimension at each location. Recently, CenterMask [27] first predicts bounding boxes together with box centerness on each location, and then predicts masks as instances segmentation inside bounding boxes. SOLO [28] reformulates the instance segmentation as a combination of category prediction and instance mask generation, which generates pixel-level segmentation masks instead of bounding boxes, requiring large computation cost.

Most recently, Xue et al. [29] propose OCR Contrastive Language-Image Pre-training, which leverages textual information to enhance visual text representations for improved scene text detection and spotting. Their approach designs a character-aware text encoder and a visual-textual decoder that can extract effective instance-level textual information and learn from partial text transcriptions without text bounding boxes. He et al. [30] propose Multi-Oriented Scene Text detector, which consists of a Text Feature Alignment Module (TFAM) and a Position-Aware Non-Maximum Suppression (PA-NMS) module. TFAM aligns image features with the coarse detection results, allowing for dynamic adjustment of the receptive field for the localization prediction layer. PA-NMS adaptively merges the raw detections based on their predicted positions, focusing on accurate predictions while discarding inaccurate ones. End-PolarT directly outputs masks with polar representation, thus formulating mask generation as a regression task instead.

## 3. Problem statement

Scene text detection refers to locating text information from complex scenes. How to acquire text related information from irregularly shaped objects like road signs, shop signs, price tags and so on, is a typical problem of scene text detection. After years of research, we conclude challenges of scene text detection as the following four tips.

Firstly, texture of text regions and backgrounds in natural scenarios are complicated, leading to quantity of miscalculations. For example, artist texts often appear where some letters would intertwine with others, which greatly hinders detection performance the segmentation-based methods.

Secondly, shapes of text regions are arbitrary, resulting in difficulties to detect curved and rotated texts. More precisely, methods that work well with regular or rectangle texts generally fail in dealing with arbitrary text regions.

Thirdly, dense anchors firstly detected in feature maps and then refine bounding boxes would bring quantity of parameters to determine during training, which requires high computational cost for pixel-wise segmentation. Such drawback would greatly harm the further usage of scene text detectors in applications.

Fourthly, mislabeling boundary pixels would result in wrong predictions, due to overlap texts or low distinguish ability of generated feature map. This phenomenon requires to take actions for more accurate labeling with counters. Meanwhile, the generated feature map for text detection should be enhanced for high distinguish capability.

## 4. The proposed method

In this section, we firstly introduce light-scale network architecture, and describe how to compute the proposed novel representation in polar coordinates. Afterwards, a assembly module is designed to generate text instances using polar centerness and polar distance regression. Finally, we describe loss function design, including Polar loss and cIoU loss design.

### 4.1. Network architecture design

We show workflow of End-PolarT in Fig. 1 to illustrate operations and key variances during detection. More precisely, End-PolarT transforms task of text detection into two sub tasks, i.e., locating centers of text instances, and predicting distance between contour points and text centers, where we show the overall structure in Fig. 2. Specifically, the input image is firstly fed into the backbone ResNet. After processing of attention module, we generate feature maps of different scales via FPN (Feature Pyramid Network). Therefore, feature map $F$ can be calculated with Equ. (1):

$$F = Backbone(I), \quad \text{where } F = \{F_i, i = 1, ..., n\} \tag{1}$$

where $I$ is the input image, $i$ refers to the scale index of output feature map, function $Backbone()$ refers to ResNet with multiple scales, and $n$ refers to the number of layers in FPN. Since different scales of feature maps denote information of text instances with variant sizes, feature map with larger or smaller index would contain global and local context information, respectively. They would be further used to predict small and large text instances, respectively.

After inputting feature maps into head with 4 parallel branches, we perform predictions to achieve classification results $O_{cls}$, polar centerness results $O_{cen}$, mask regression results $O_{pol}$ and box regression results $O_{box}$ with Equ. (2):

$$O_{cls} = H_1(F), \ O_{cen} = H_2(F), \ O_{pol} = H_3(F), \ O_{box} = H_4(F) \tag{2}$$

where functions $H_1()$, $H_2()$, $H_3()$, $H_4()$ denote the corresponding operations in head structure for different purposes.

For feature map $F$ with $H \times W \times C$, we obtain output $O_{pol}$ with $H \times W \times m$ and $O_{box}$ with $H \times W \times 4$, where $m$ refers to the total number of rays emitted from every pixel, and each of $m$ dimensions corresponds to a specific angle. Since most of text instances are shaped with rectangles, we design box regression branch, where the horizontal and vertical rays are calculated with edges of rectangles in an average way. By designing such structure, End-PolarT would pay attention on specific angles instead of treating all directions equally.

In classification branch, output of classification block is defined with $H \times W \times k$, where $k$ is the number of total classes and equals 2 in text detection, i.e., texts and non-texts. In polar centerness branch, output dimension is defined as $H \times W \times 1$, determining whether the corresponding pixel is near the center of text instances. Essentially, we adopt polar centerness and classification results to filter low-quality predictions. Afterwards, we calculate classification score $O_{fcls}$ with Equ. (3):

$$O_{fcls} = O_{cls} * O_{cen} \tag{3}$$

where $*$ denotes element-wise product.

Afterwards, we assemble results from four branches in head to compute text detection results $R$ with post-processing operations as shown in Equ. (4):

$$R = f_{post}(f_{ass}(O_{fcls}, O_{pol}, O_{box})) \tag{4}$$

where functions $f_{post}()$ and $f_{ass}()$ denote operations of post processing and assembly module.

### 4.2. Representation computing in polar coordinates

Inspired by PolarMask [31] to enhance feature representation for object detection, it's essential to offer abundant and enhanced representation in polar coordinates for accurate text detection. We

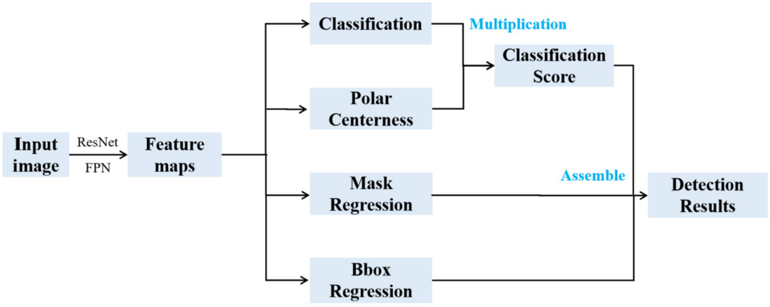

**Fig. 1.** Workflow of End-PolarT to detect texts in the wild. Firstly, feature maps are extracted through the ResNet and FPN network structure. Then, four branches are designed to achieve classification results, polar centerness, mask regression, bbox regression, respectively. Afterwards, classification score is obtained by multiplication between classification and polar centerness. Finally, all three results are assembled to compute final detection results.

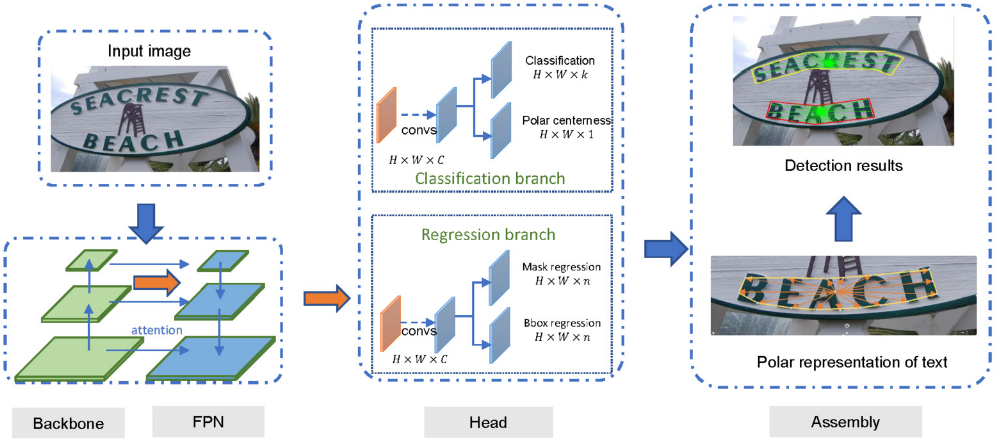
Input image
attention
Backbone
FPN
convs
$H \times W \times C$
Classification $H \times W \times k$
Polar centerness $H \times W \times 1$
Classification branch
Regression branch
Mask regression $H \times W \times n$
Bbox regression $H \times W \times n$
convs
$H \times W \times C$
Head
Detection results
Polar representation of text
Assembly

**Fig. 2.** Overall network structure of End-PolarT, which consists of backbone, FPN, head, and assembly module. Specifically, feature map is extracted by backbone and FPN, where features are processed by four parallel branches in head, thus obtaining polar representation for texts. Finally, Assembly module help achieve text detection results by involving different categories of information computed by the head.

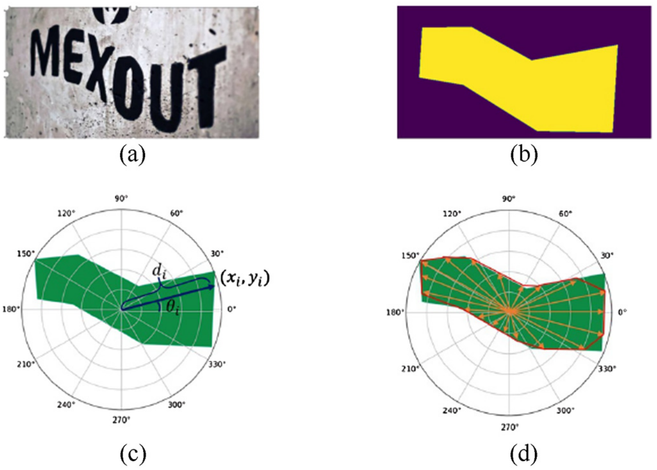

**Fig. 3.** Steps to compute representations of text instances in polar coordinates: (a) the input text instance, (b) the generated text mask, (c) calculate coordinates of contour points, and (d) mask segmentation with polar representation.

thus design to compute representations of text instances in polar representation as shown in Fig. 3, where we firstly locate polar centers, and then settle contour points corresponding to a specific angle. In such way, we repeat several times to obtain contour points, which are enough to generate text mask in arbitrary shapes.

Specifically, we firstly represent text instances as a set of contour points in polar coordinates. In fact, contour points are determined by the distance and angle emitted from polar center, where we can easily reconstruct text instances via contour points. Starting from the polar center, we emit $n$ rays uniformly, where $n$ is defined as 36 for convinced representation of texts.

Define polar center as $(x_c, y_c)$, coordinates of the $i$th contour point $(x_i, y_i)$ $i = 1, 2, ..., n$ can be calculated via Equ. (5):

$$\begin{cases} x_i = \cos\theta_i \times d_i + x_c \\ y_i = \sin\theta_i \times d_i + y_c \end{cases} \tag{5}$$

where $\theta_i$ and $d_i$ represents the corresponding angle and distance for the $i$th contour point.

For each ground-truth sample in training dataset, we predict distances between contour points and sample points as regression target. More precisely, we define contour point owns the largest distance from the sample point, if there are more than one point that intersect with the ray. If center point is rarely located outside the mask, we set the regression goal as the minimum value. A point can only be considered as a sample point, only if it's near the center point of a text instance, thus ensuring that we always sample from the polar center.

### *4.3. Structure design of assembly module*

As shown in Fig. 4, we design the structure of assembly module to detect texts, where a total of $H * W$ text instances with corresponding masks and bonding boxes will be generated. Similar with polar representation of masks, we generate bounding box with a box center and different edges emitted from box center.

To achieve convinced outputs, we only consider pixels near the box centers of text instances as positive samples. Therefore, the output classification results refer to probability whether the corre-

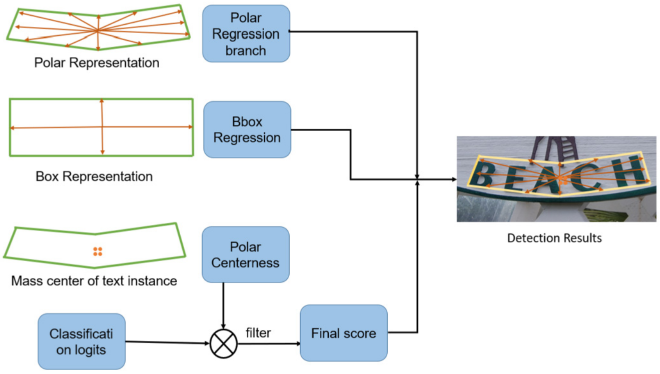

**Fig. 4.** Structure design of the proposed assembly module, where text detection results are generated based on polar regression, box regression, polar centerness, and classification results.

sponding pixel is a positive sample. Moreover, polar centerness $\xi$ is used to suppress low-quality outputs defined as Equ. (6):

$$\xi = \sqrt{d_{min}/d_{max}} \tag{6}$$

where $d_{min}$ and $d_{max}$ are the minimum and maximal length of all $n$ rays, respectively.

According to the $i$th pixel, we multiply its classification result and the corresponding polar centerness to obtain the final score. Then, we use the score to filter low-quality outputs, where we only keep 1000 top-scoring predictions on each scaled feature map with Non-maximal Suppressing algorithm.

### 4.4. Loss function design with polar and cIoU loss

Supposing $N$ pairs of predictions $D_i$ and the corresponding ground-truth labels $G_i$, loss function for End-PolarT could be defined as Equ. (7):

$$L = \frac{1}{N}\sum_{i=1}^{N} L_{cls}(D_i, G_i) + L_{cen}(D_i, G_i) + L_{pol}(D_i, G_i) + L_{box}(D_i, G_i) \tag{7}$$

where $L_{cls}$, $L_{cen}$, $L_{pol}$, $L_{box}$ are loss of classification, polar centerness regression, polar mask regression, bounding box regression, respectively. Note that we use polar loss and cIoU loss to calculate loss of polar mask regression and bounding box regression.

Since masks are represented in polar coordinates, it requires quantity of computation to firstly reconstruct the pixel-level mask instances and then calculate losses per pixel. Given $d_i$ and $d_i^*$ as the distance of the $i$th predicted ray, Polar IoU is thus designed as an approximation of IoU loss in polar coordinates, represented as Equ. (8):

$$L_{pol} = \log \frac{\sum_{i=1}^{n} \max(d_i, d_i^*)}{\sum_{i=1}^{n} \min(d_i, d_i^*)} \tag{8}$$

The Bounding box branch is parallel to mask regression branch, where we adopt cIoU loss [32] to suppress low overlap detected regions with ground-truth samples as represented in Equ. (9):

$$L_{box} = 1 - IoU + \frac{\parallel b, b^{gt} \parallel}{c^2} + \alpha v \tag{9}$$

where $b$, $b^{gt}$ denote center points of predicted boxes and ground-truth boxes respectively, $\|\|$ refers to Euclidean distance, and $c$ denotes the length of diagonal of the smallest enclosing box covering both boxes. $\alpha$ is a trade-off parameter and $v$ measures aspect ratio of both boxes, which are defined as Equ. (10):

**Table 1**
Performance comparisons with different structure designs on CTW1500 and ICDAR2015 dataset.

| Dataset | Method | Precision | Recall | F-score | FPS |
|---|---|---|---|---|---|
| CTW1500 | ResNet50+ IoU | 81.3 | 73.1 | 77.8 | **13.3** |
| | ResNet101 + IoU | 82.5 | 76.7 | 79.4 | 7.7 |
| | Attention + IoU | 83.2 | 78.7 | 80.8 | 8.9 |
| | Attention + cIoU | **83.5** | **78.8** | **81.0** | 8.6 |
| ICDAR2015 | ResNet50 + IoU | 81.3 | 73.1 | 77.8 | **13.3** |
| | ResNet101 + IoU | 82.5 | 76.7 | 79.4 | 7.7 |
| | Attention + IoU | 83.2 | 78.7 | 80.8 | 8.9 |
| | Attention + cIoU | **83.5** | **78.8** | **81.0** | 8.6 |

$$\begin{cases} v = \frac{4}{\pi^2}\left(\arctan \frac{w^{gt}}{h^{gt}} - \arctan \frac{w}{h}\right)^2 \\ \alpha = \frac{v}{(1-IoU)+v} \end{cases} \tag{10}$$

where $w$ and $h$ denote the width and height of the predicted boxes, respectively.

The reason that polar loss and cIoU Loss could be integrated in the total loss function is that both loss functions are re-scaled without information loss. Moreover, sub loss form can be easily trained in BP algorithm. Moreover, polar loss function enables shape of predicted texts and ground-truth texts to be similar in polar representation, meanwhile cIoU ensures that both shape are similar in rectangle representation form. By constraining shape similarity in both loss functions, End-PolarT is guided to train, resulting in the most shape-similar predictions.

## 5. Experiment analysis

In this section, we first introduce dataset and measurements. Then, we conduct ablation and parameter setting experiments to show designs of End-PolarT is highly effective. Afterwards, two groups of comparative studies on several public dataset are conducted to demonstrate End-PolarT is effective in text detection. Finally, we describe implementation details for readers' convenience.

### 5.1. Datasets and measurements

We choose five datasets for experiments, i.e., ICDAR2015, ICDAR17MLT, MSRA, Total Text, SCUT CTW1500. Annotations of ICDAR15 dataset are labeled as 4 vertices at word level, meanwhile annotations of CTW1500 and Total text are labeled with boundary points at text level. Annotations of MSRA are labeled as rectangle boxes and angles, we convert them to vertices of quadrilaterals for fairness.

Evaluation measurements are defined to obey rules of Pascal Voc, where any text instance that has IoU larger than 0.5 with any ground truth will be considered as positive, and each ground truth could have only one positive example. We use precision, recall and the F-value to evaluate the performance of text detection.

### 5.2. Ablation experiments

As shown in Table 1, we conduct experiments on CTW1500 and ICDAR2015 datasets to prove the effectiveness of different components. In CTW1500 dataset, we can observe that attention module improves detection performance by a large margin, which even outperforms ResNet101 network with much heavier structure design. It's noted that attention module gains less performance boosting in ICDAR15 dataset, since samples of CTW1500 are much more complicated in appearances and shapes than those of ICDAR2015, thus context information extracted by attention module better promoting detection performance. Moreover, it's noted that

**Table 2**
Performance comparison with different number of rays on CTW1500 dataset.

| No. of Rays | Precision | Recall | F-score |
|---|---|---|---|
| 18 | 82.3 | 74.1 | 78.0 |
| 36 | 83.5 | 78.8 | 81.0 |
| 48 | **83.7** | **78.9** | **81.2** |
| 64 | 83.4 | 78.7 | 81.0 |

**Table 3**
F-score Performance comparison in convergence speed on CTW1500 dataset.

| Methods | Epoch50 | Epoch100 | Epoch200 | Epoch300 | Epoch400 |
|---|---|---|---|---|---|
| IoU | 34.3 | 56.1 | 68.7 | **78.5** | 78.3 |
| cIoU | 34.6 | 60.2 | 76.2 | **80.7** | 80.6 |

cIoU has small impact on detection performance. However, it could help the network converge in a fast manner during training.

### 5.3. Parameter setting experiments

Since rays are emitted from center to represent text instances in polar coordinates, more rays would improve representation capability. In Table 2, we show the impacts of setting different number of rays. When increasing the number of rays from 18 to 36, there is a great improvement in performance. Meanwhile, the performance gain is relatively small, when increasing from 36 to 48. Performance even drops when increasing from 48 to 64. Therefore, 48 rays are enough to represent text instance for accurate detection.

As shown in Table 3, comparisons of the convergence speed prove that cIoU converges much faster during training. After training with 300 epoches, End-PolarT would be converged to a stable situation where we thus choose 300 as the number of training epoches.

### 5.4. Comparative experiments

Comparison of results are shown in Table 4 and 5. In ICDAR2015 and CTW1500 with quadrilateral text instances, End-PolarT outperforms most of the existing methods. Moreover, performance is significantly improved in ICDAR2017MLT, MSRA, and Total-text datasets, where most text instances are rotated or curved. Such phenomenon indicates that End-PolarT is rotation invariant for text detection.

Moreover, ICDAR17MLT dataset contains multiple languages with complex and diverse scene for detecting. Experiments show that End-PolarT works well on this challenging dataset, which proves the power of utilizing polar representations for text detection. INn curved text dataset like ICDAR2015 and CTW1500, End-PolarT gain competitive performance with Mask R-CNN. Moreover, End-PolarT achieve an FPS performance of 8.8, which is 4 times faster than that of Mask R-CNN. All these phenomena can be explained that Mask R-CNN is a two-stage method, requiring to generate dense predefined anchors for quantity of computation cost, thus being much slower than End-PolarT, i.e., single-stage text detector. Note that Mask R-CNN tends to have a higher recall than End-PolarT, since Mask R-CNN use predefined anchors to ensure the existence of most text instances.

In all datasets, End-PolarT outperforms Bottom-up methods like PSENet [5] and TextSnake [15], due to high difficulties to achieve convinced pixel-level segmentation results where text instances nearby can be easily distinguished. We show results of text detection achieved by End-PolarT in Fig. 5 and Fig. 6, where detecting texts in polar representation could greatly improve accuracy performance.

**Table 4**
Performance comparisons with the existing methods on CTW-1500 and ICDAR2015 dataset.

| Datasets | Method | Precision | Recall | F-score | FPS |
|---|---|---|---|---|---|
| CTW1500 | CTPN [18] | 60.4 | 53.8 | 56.9 | 7.1 |
| | SegLink [33] | 42.3 | 40.0 | 40.8 | 10.7 |
| | EAST [34] | 78.7 | 49.1 | 60.4 | – |
| | CTD [35] | 74.3 | 65.2 | 69.5 | – |
| | CTD+TLOC [35] | 77.4 | 69.8 | 73.4 | **13.3** |
| | DMPNet [36] | 69.9 | 56.0 | 62.2 | – |
| | TextSnake [15] | 67.9 | 85.3 | 75.6 | 8.2 |
| | PSENet [5] | 80.6 | 75.6 | 78.0 | 3.9 |
| | LOMO [37] | 85.7 | 69.6 | 76.8 | 4.4 |
| | Mask R-CNN [17] | 80.8 | **83.1** | **81.9** | 1.8 |
| | End-PolarT | **83.5** | 78.8 | 81.0 | 8.6 |
| ICDAR2015 | CTPN [18] | 74.2 | 51.6 | 60.9 | 7.1 |
| | Zhang et al. [37] | 70.8 | 43.0 | 53.6 | 0.5 |
| | PixelLink [33] | 82.9 | 81.7 | 82.3 | 7.3 |
| | MSR [38] | 86.6 | 78.4 | 82.3 | – |
| | EAST [34] | 83.6 | 73.5 | 78.2 | **13.2** |
| | TextDragon [39] | 84.8 | 81.8 | 83.1 | 7.5 |
| | PSENet [5] | 81.5 | 79.7 | 80.6 | 1.6 |
| | PAN [40] | 77.8 | **82.9** | 80.3 | – |
| | Mask R-CNN [17] | 86.3 | 81.5 | 83.8 | 1.9 |
| | End-PolarT | **88.1** | 80.2 | **84** | 8.7 |

**Table 5**
Performance comparisons with the existing methods on ICDAR17MLT, MSRA, Total-Text datasets.

| Dataset | Method | Precision | Recall | F-score | FPS |
|---|---|---|---|---|---|
| ICDAR17MLT | TDN SJTU2017 [41] | **86.0** | **70.0** | **77.0** | – |
| | He et al. [42] | 76.7 | 57.9 | 66.0 | – |
| | Pixellink [33] | 70.9 | 61.7 | 65.4 | 7.3 |
| | Mask R-CNN [17] | 74.8 | 61.1 | 67.2 | 2.1 |
| | End-PolarT | 75.6 | 62.8 | 68.6 | **9.7** |
| MSRA | SegLink [43] | 86.0 | 70.0 | 77.0 | – |
| | [34] | 81.7 | 61.6 | 70.2 | 6.5 |
| | TextSnake [15] | 83.2 | 73.9 | 78.3 | 1.1 |
| | Zhang et al. [37] | 83.0 | 67.0 | 74.0 | 0.48 |
| | He et al. [42] | 77.0 | 70.0 | 74.0 | 1.1 |
| | Pixellink [33] | 83.0 | 73.2 | 77.8 | 3.0 |
| | Mask R-CNN [17] | 84.6 | 80.5 | 82.5 | 1.9 |
| | End-PolarT | **87.0** | **81.2** | **83.9** | **9.5** |
| Total-text | SegLink [33] | 30.3 | 23.8 | 26.7 | 7.7 |
| | EAST [34] | 50.0 | 36.2 | 42.0 | – |
| | MSR [38] | 83.8 | 74.8 | 79.0 | 4.3 |
| | TextSnake [15] | 82.7 | 74.5 | 78.4 | 3.6 |
| | PSENet [5] | 81.8 | 75.1 | 78.3 | 3.9 |
| | Mask R-CNN [17] | 82.3 | **84.5** | **83.3** | 1.5 |
| | End-PolarT | **82.4** | 76.6 | 79.3 | **7.9** |

### 5.5. Implementation details

End-PolarT network is trained with stochastic gradient descent, by setting initial learning to 0.01. Warm-up policy is adopted to prevent to get trapped into local minimum. The positive and negative IoU threshold is set to 0.4 and 0.5, respectively. During training, simple data augmentation is used such as random resize, crop and clip. We use ResNet 50 as backbones and non-local networks as our attention module. All our experiments are conducted on 4 Nvidia GTX 1080 TI GPUs.

## 6. Conclusion

This paper proposes End-PolarT network to detect texts by directly generating contour points of text instances in polar coordinates representation. End-PolarT not only relieves burden of high computation cost brought by pixel-level classification, but also fits with intrinsic characteristics of text instances to eliminate mislabeled boundary pixels. Comparing with the existing methods, we

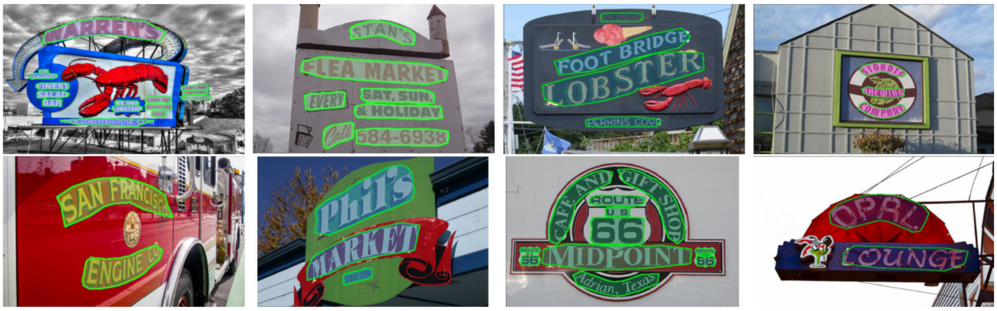

**Fig. 5.** Detection results achieved by End-PolarT on CTW1500 dataset.

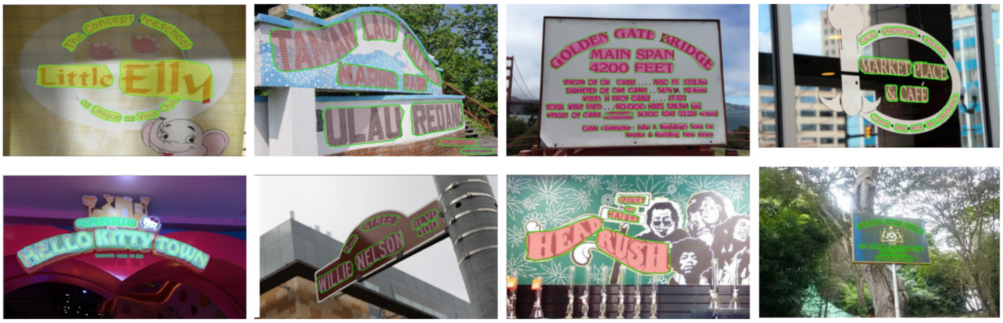

**Fig. 6.** Detection results achieved by End-PolarT on Total-text dataset.

conduct experiments on multiple scene text datasets. It's noted that attention module largely improves text detection performance and cIoU loss design helps coverage in a fast speed. By testing with variant numbers, number of rays and epoches should be defined as 48 and 300 for best performance. In comparisons, End-PolarT outperforms most of the existing methods, especially in curved and rotated texts. End-PolarT keeps balance between computing speed and performance, comparing with Mask-RCNN. Inspired by digital twin [44], our future work is to design special representation of shape for texts with complex contours, thus greatly improving accuracy for difficult cases.

## Declaration of competing interest

The authors declare that there is no conflict of interests regarding the publication of this article.

## Data availability

Data will be made available on request.

## Acknowledgement

This work was supported by National Natural Science Foundation of China under Grant No. 62172438, National Key R&D Program of China under Grant No. 2021YFB3900601, the Fundamental Research Funds for the Central Universities under Grant B220202074, and the Fundamental Research Funds for the Central Universities, JLU.

## References

[1] Y. Zhang, F. Zhang, Y. Jin, Y. Cen, V. Voronin, Local correlation ensemble with gcn based on attention features for cross-domain person re-id, ACM Trans. Multimed. Comput. Commun. Appl. 19 (2) (2023), https://doi.org/10.1145/3542820.

[2] J. Gou, L. Sun, B. Yu, S. Wan, W. Ou, Z. Yi, Multi-level attention-based sample correlations for knowledge distillation, IEEE Trans. Ind. Inform. (2022).

[3] S. Ding, S. Qu, Y. Xi, S. Wan, A long video caption generation algorithm for big video data retrieval, Future Gener. Comput. Syst. 93 (2019) 583–595.

[4] M. Cao, C. Zhang, D. Yang, Y. Zou, All you need is a second look: towards arbitrary-shaped text detection, IEEE Trans. Circuits Syst. Video Technol. 32 (2) (2022) 758–767.

[5] W. Wang, E. Xie, X. Li, W. Hou, T. Lu, G. Yu, S. Shao, Shape robust text detection with progressive scale expansion network, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 9336–9345.

[6] Y. Zhu, J. Chen, L. Liang, Z. Kuang, L. Jin, W. Zhang, Fourier contour embedding for arbitrary-shaped text detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 3123–3131.

[7] Z. Huang, Z. Zhong, L. Sun, Q. Huo, Mask r-cnn with pyramid attention network for scene text detection, in: Proceedings of IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE, 2019, pp. 764–772.

[8] Y. Wang, H. Xie, Z. Zha, M. Xing, Z. Fu, Y. Zhang, Contournet: taking a further step toward accurate arbitrary-shaped scene text detection, in: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 11750–11759.

[9] J. Wu, P. Lyu, G. Lu, C. Zhang, K. Yao, W. Pei, Decoupling recognition from detection: single shot self-reliant scene text spotter, in: Proceedings of the 30th ACM International Conference on Multimedia, 2022, pp. 1319–1328.

[10] M. Liao, B. Shi, X. Bai, X. Wang, W. Liu, Textboxes: a fast text detector with a single deep neural network, in: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, 2017, pp. 4161–4167.

[11] M. Liao, B. Shi, X. Bai, Textboxes++: a single-shot oriented scene text detector, IEEE Trans. Image Process. 27 (8) (2018) 3676–3690.

[12] M. Liao, Z. Zhu, B. Shi, G.s. Xia, X. Bai, Rotation-sensitive regression for oriented scene text detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 5909–5918.

[13] M. Liao, Z. Wan, C. Yao, K. Chen, X. Bai, Real-time scene text detection with differentiable binarization, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol 34, 2020, pp. 11474–11481.

[14] H. Ma, N. Lu, J. Mei, T. Guan, Y. Zhang, X. Geng, Label distribution learning for scene text detection, Front. Comput. Sci. 17 (6) (2023) 176339.

[15] S. Long, J. Ruan, W. Zhang, X. He, W. Wu, C. Yao, Textsnake: a flexible representation for detecting text of arbitrary shapes, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 20–36.

[16] S. Long, S. Qin, D. Panteleev, A. Bissacco, Y. Fujii, M. Raptis, Towards end-to-end unified scene text detection and layout analysis, in: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 1039–1049.

[17] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask r-cnn, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2961–2969.

[18] E. Xie, Y. Zang, S. Shao, G. Yu, C. Yao, G. Li, Scene text detection with supervised pyramid context network, in: Proceedings of AAAI Conference on Artificial Intelligence, 2019, pp. 9038–9045.

[19] J. Ye, Z. Chen, J. Liu, B. Du, Textfusenet: scene text detection with richer fused features, in: Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, 2020, pp. 516–522.

[20] W. Wang, X. Liu, X. Ji, E. Xie, D. Liang, Z. Yang, T. Lu, C. Shen, P. Luo, Ae textspotter: learning visual and linguistic representation for ambiguous text spotting, in: Proceedings of European Conference on Computer Vision, vol. 12359, 2020, pp. 457–473.

[21] Y. Wu, H. Guo, C. Chakraborty, M. Khosravi, S. Berretti, S. Wan, Edge computing driven low-light image dynamic enhancement for object detection, IEEE Trans. Netw. Sci. Eng. (2022) 1, https://doi.org/10.1109/TNSE.2022.3151502.

[22] A. Bochkovskiy, C. Wang, H.M. Liao, Yolov4: optimal speed and accuracy of object detection, CoRR, arXiv:2004.10934 [abs], 2020.

[23] Z. Tian, C. Shen, H. Chen, T. He, Fcos: fully convolutional one-stage object detection, in: Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 9627–9636.

[24] M. Bai, R. Urtasun, Deep watershed transform for instance segmentation, in: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2017, pp. 2858–2866.

[25] D. Bolya, C. Zhou, F. Xiao, Y.J. Lee, Yolact: real-time instance segmentation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 9157–9166.

[26] X. Chen, R. Girshick, K. He, P. Dollár, Tensormask: a foundation for dense object segmentation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 2061–2069.

[27] Y. Lee, J. Park, Centermask: real-time anchor-free instance segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 13906–13915.

[28] R. Zhang, Z. Tian, C. Shen, M. You, Y. Yan, Mask encoding for single shot instance segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 10226–10235.

[29] C. Xue, W. Zhang, Y. Hao, S. Lu, P.H.S. Torr, S. Bai, Language matters: a weakly supervised vision-language pre-training approach for scene text detection and spotting, in: Proceedings of European Conference on Computer Vision, 2022, pp. 284–302.

[30] M. He, M. Liao, Z. Yang, H. Zhong, J. Tang, W. Cheng, C. Yao, Y. Wang, X. Bai, Most: a multi-oriented scene text detector with localization refinement, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 8813–8822.

[31] E. Xie, P. Sun, X. Song, W. Wang, X. Liu, D. Liang, C. Shen, P. Luo, Polarmask: single shot instance segmentation with polar representation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 12193–12202.

[32] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, D. Ren, Distance-iou loss: faster and better learning for bounding box regression, in: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), 2020, pp. 12993–13000.

[33] D. Deng, H. Liu, X. Li, D. Cai, Pixellink: detecting scene text via instance segmentation, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32, 2018.

[34] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, J. Liang, East: an efficient and accurate scene text detector, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2017, pp. 5551–5560.

[35] Y. Liu, L. Jin, S. Zhang, C. Luo, S. Zhang, Curved scene text detection via transverse and longitudinal sequence connection, Pattern Recognit. 90 (2019) 337–345.

[36] Y. Liu, L. Jin, Deep matching prior network: toward tighter multi-oriented text detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1962–1969.

[37] C. Zhang, B. Liang, Z. Huang, M. En, J. Han, E. Ding, X. Ding, Look more than once: an accurate detector for text of arbitrary shapes, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 10552–10561.

[38] C. Xue, S. Lu, W. Zhang, MSR: multi-scale shape regression for scene text detection, in: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, 2019, pp. 989–995.

[39] W. Feng, W. He, F. Yin, X.Y. Zhang, C.L. Liu, Textdragon: an end-to-end framework for arbitrary shaped text spotting, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 9076–9085.

[40] W. Wang, E. Xie, X. Song, Y. Zang, W. Wang, T. Lu, G. Yu, C. Shen, Efficient and accurate arbitrary-shaped text detection with pixel aggregation network, in: Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 8440–8449.

[41] N. Nayef, F. Yin, I. Bizid, H. Choi, Y. Feng, D. Karatzas, Z. Luo, U. Pal, C. Rigaud, J. Chazalon, et al., Icdar2017 robust reading challenge on multi-lingual scene text detection and script identification-rrc-mlt, in: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), vol. 1, IEEE, 2017, pp. 1454–1459.

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

[43] B. Shi, X. Bai, S. Belongie, Detecting oriented text in natural images by linking segments, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2550–2558.

[44] Y. Wu, H. Cao, G. Yang, T. Lu, S. Wan, Digital twin of intelligent small surface defect detection with cyber-manufacturing systems, ACM Trans. Internet Technol. (2022), https://doi.org/10.1145/3571734.