Pattern Recognition 90 (2019) 337–345

Contents lists available at ScienceDirect

Pattern Recognition

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

Curved scene text detection via transverse and longitudinal sequence connection

Yuliang Liu, Lianwen Jin*, Shuaitao Zhang, Canjie Luo, Sheng Zhang

College of Electronic Information Engineering, South China University of Technology, China

ARTICLE INFO

Article history:
Received 17 March 2018
Revised 17 January 2019
Accepted 4 February 2019
Available online 05 February 2019

Keywords:
Scene text
Curved dataset
LSTM
CNN
Deep learning

ABSTRACT

Curved text detection is a difficult problem that has not been addressed sufficiently. To highlight the difficulties in reading curved text in a real environment, we constructed a curved text dataset called CTW1500, which includes over 10,000 text annotations in 1500 images, and used it to formulate a polygon-based curved text detector that can detect curved text without using an empirical combination. With the seamless integration of recurrent transverse and longitudinal offset connection, our method explores context information instead of predicting points independently, resulting in smoother and more accurate detection. Our approach is designed as a universal method, meaning it can be trained using rectangular or quadrilateral bounding boxes, requiring no extra effort. Experimental results on the CTW1500 dataset and Total-text demonstrated that our method with only a light backbone can outperform state-of-the-art methods by a large margin. Our method also achieved state-of-the-art performance on the MSRA-TD500 dataset, demonstrating its promising generalization ability. Code, datasets, and label-tool are available at https://github.com/Yuliang-Liu/Curve-Text-Detector.

1. Introduction

Text in the real world conveys valuable information that can be used for real-time multi-lingual translation, behavior analysis, product identification, automotive assistance, and for many other purposes.

Recently, the emergence of many text datasets that are constructed for specific tasks and scenes has contributed to significant progress in text detection and recognition methods. Interestingly, it has been observed that labels for text bounding boxes in emerging datasets have also developed from rectangles to flexible quadrangles. Examples include horizontal rectangular labels in ICDAR 2013 “Focus Scene Text” [1] and SVT [2], rotated rectangular labels in MSRA-TD500 [3] and USTB-SV1K [4], and four-point labels in ICDAR 2015 “Incidental Scene Text” [5], RCTW-17 [6], and the recent multi-lingual text (MLT) competition dataset [7]. Similar advancements in scene text detection methods have shifted from axis-aligned-rectangle-based methods to rotated-rectangle-based and quadrangle-based methods. The study in [8] found that when the bounding box becomes tighter and more flexible, it can improve the detection confidence, reduce the risk of it being suppressed by post-processing techniques and benefit subsequent text recognition tasks.

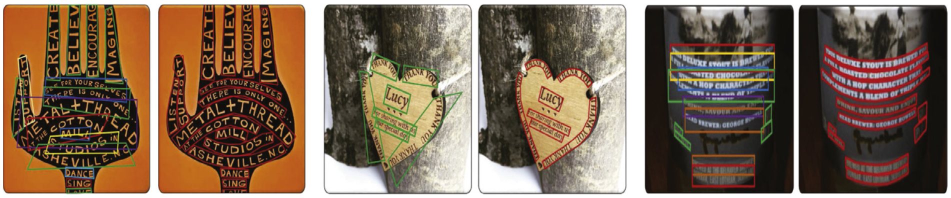
An important requirement for recognizing scene text is that the text should be tightly and robustly localized in advance. However, current datasets have very little curved text, and it is no sufficient to label that text with quadrangles let alone rectangles. As shown in Fig. 1, using a curved bounding box has three significant advantages:

- **Avoids needless overlap**. Because text may appear in many shapes, the traditional four-points localization method may not be able to handle such elusive peculiarities. As shown in Fig. 1(a), a quadrilateral bounding box cannot avoid a large amount of superfluous overlap, while a curved bounding box can.
- **Less background noise**. As shown in Fig. 1(b), if text appears in a curved form, a quadrilateral bounding box suffers from background noise.
- **Avoids multiple text lines**. Recent popular recognition methods [9–12] all require a single row of text in each bounding box. However, in some cases, as in Fig. 1(c), a quadrilateral bounding box cannot avoid containing multiple lines of text, whereas a curved bounding box can solve this problem.

Curved text is very common in the real world. For example, text on most types of columnar objects (e.g., bottles, stone piles),

* Corresponding author.
E-mail addresses: liu.yuliang@mail.scut.edu.cn (Y. Liu), eelwjin@scut.edu.cn (L. Jin).

https://doi.org/10.1016/j.patcog.2019.02.002



(a) Needless overlap. (b) Redundant background noise. (c) Stacked text.

Fig. 1. Comparison of quadrilateral bounding box and polygonal bounding box for localizing text. Left: using quadrilateral label. Right: using polygonal label.

spherical objects, plicated planes (e.g., clothes, streamers), coins, logos and signboards. However, to the best of our knowledge, current methods cannot directly detect curved text. Linking methods [13–15] can detect components of text and group them together to match the curved bounding box, but if there are many texts stacked together, such as in Fig. 1(a), empirical connection rules are unable to group small components properly, meaning these methods often encounter a large number of false positives in practice when compared with direct detection methods.

In this paper, we collected text from various natural scenes, websites and image libraries (Google Open-Image [16]) and constructed a new dataset called CTW1500. Although the first curved dataset CUTE80 [17] was constructed in 2014, it contains only 80 images with deficient annotation, and with much of the recognizable text being unlabeled. It was not until late 2017 that another curved dataset called Total-text [18] emerged. In contrast to Total-text, our annotation method is on the basis of relatively objective segmented equidistant points, and each bounding box is labeled with a sufficient number of points, whereas some bounding boxes in Total-text are visibly loose. In addition, our CTW1500 dataset contains both English and Chinese text.

This dataset contains 1500 images with over 10,000 text annotations, with each image containing at least one piece of curved text. For evaluation and comparison, we used 1000 images as a training set and 500 as a testing set. Based on our observations, a 14-point polygon is sufficient for localing all types of curved text regions, as shown in Figs. 1 and 2. By using equidistant reference lines, our method avoids the requirement for significant manpower for labeling, which is discussed in detail in Section 3.

Using the CTW1500 dataset, we formulated a simple but effective polygon-based curved text detector (CTD) that can directly detect curved text. Unlike traditional detection methods, the CTD separates the branches for width/height offsets prediction, and can be run on less than 4 GB of video memory at a speed of 13 FPS. The network architecture can also be seamlessly integrated using a novel method called transverse and longitudinal offset connection (TLOC), which uses a recurrent neural network (RNN) to learn the inherent connection between locating points, making the detection smoother and more accurate. The CTD is also designed as a universal method, meaning it can be trained using rectangular and quadrilateral bounding boxes without additional manual labels. Two simple but effective post-processing methods called non-polygon suppression (NPS) and polygonal non-maximum suppression (PNMS) are also proposed to further enhance the generalization ability of CTD.

On the proposed dataset and Total-text [18], the results demonstrate that the CTD with a lightly reduced ResNet-50 [19] can effectively detect the curved text and outperform state-of-the-art methods by a large margin. When evaluating our method on mere curved and non-curved test subsets (considering the other text as “not care” regions), CTD+TLOC still achieves the best results. Experiment on the MSRA-TD500 dataset further demonstrated that our method can achieve state-of-the-art performance when compared with recent state-of-the-art methods under the same settings.

Our main contributions can be summarized as follows:

- We proposed a new curved dataset to facilitate our curved detection method, whose annotation is based on relative objective method that is very accurate.
- We proposed a novel CTD method that can effectively detect both curved and non-curved text.
- Seamless integration of an RNN method (TLOC) significantly improved the detection performance.
- Implementation of polygonal post-processing methods (NPS and PNMS) further improved the results.
- Our method achieved state-of-the-art performance on curved and non-curved datasets.

2. Related work

In the past few decades, scene text detection methods have seen significant improvements. One of the main reasons for this progress is the evolution of benchmark datasets - the data become harder, the amounts become larger and the labels become tighter. Since 2003, rectangular labeled datasets such as ICDAR’03 [20], ICDAR’11 [21], ICDAR’13 [1] and COCO-Text [22] have attracted significant research efforts. After 2010, the multi-oriented datasets with rotated rectangular labels (NEOCR [23], OSTD [24], MSRA-TD500 [3] and USTB-SV1K [4]) emerged, which stimulated the presentation of many influential multi-oriented detection methods in the literature. In 2015, the first quadrilateral labeled dataset, named ICDAR 2015 “Incidental Scene Text” [5] appeared and attracted unprecedented amounts of attention according to its evaluating website [5] and encouraged much recent progress. Since then, several larger and more challenging quadrilateral labeled datasets have emerged in the ICDAR 2017 competition, such as RCTW-17 [6] (dataset for Chinese and English text), DOST [25] (scene texts observed by video in the real environment) and MLT [7] (dataset for multi-lingual text). These are quickly becoming the new mainstream datasets.

Interestingly, the development of detection methods show similar evolutionary tendency to that of annotation methods for datasets. Although rectangular methods still receive attention [26–30,32], they have become less mainstream and mostly focus on non-scene text. Methods with rotated rectangular bounding boxes have been proposed almost every year since 2011. In 2017, many quadrilateral-based detection methods [8,14,31,33–36,54] emerged. Basically, quadrilateral-based detection methods can achieve the best performance in both rotated datasets and horizontal datasets (by evaluating the circumscribed rectangle), and they have demonstrated that they can outperform strictly horizontal methods on multi-oriented datasets (by using the circumscribed rectangle to train and test), especially in terms of recall rate. This is mainly because the stronger supervision for quadrilateral labeled methods can avoid significant background noise, unreasonable suppression, and information loss. The viewpoint that stronger supervision aids in detection is supported by Mask-RCNN [37], which improves the detection results by jointly training

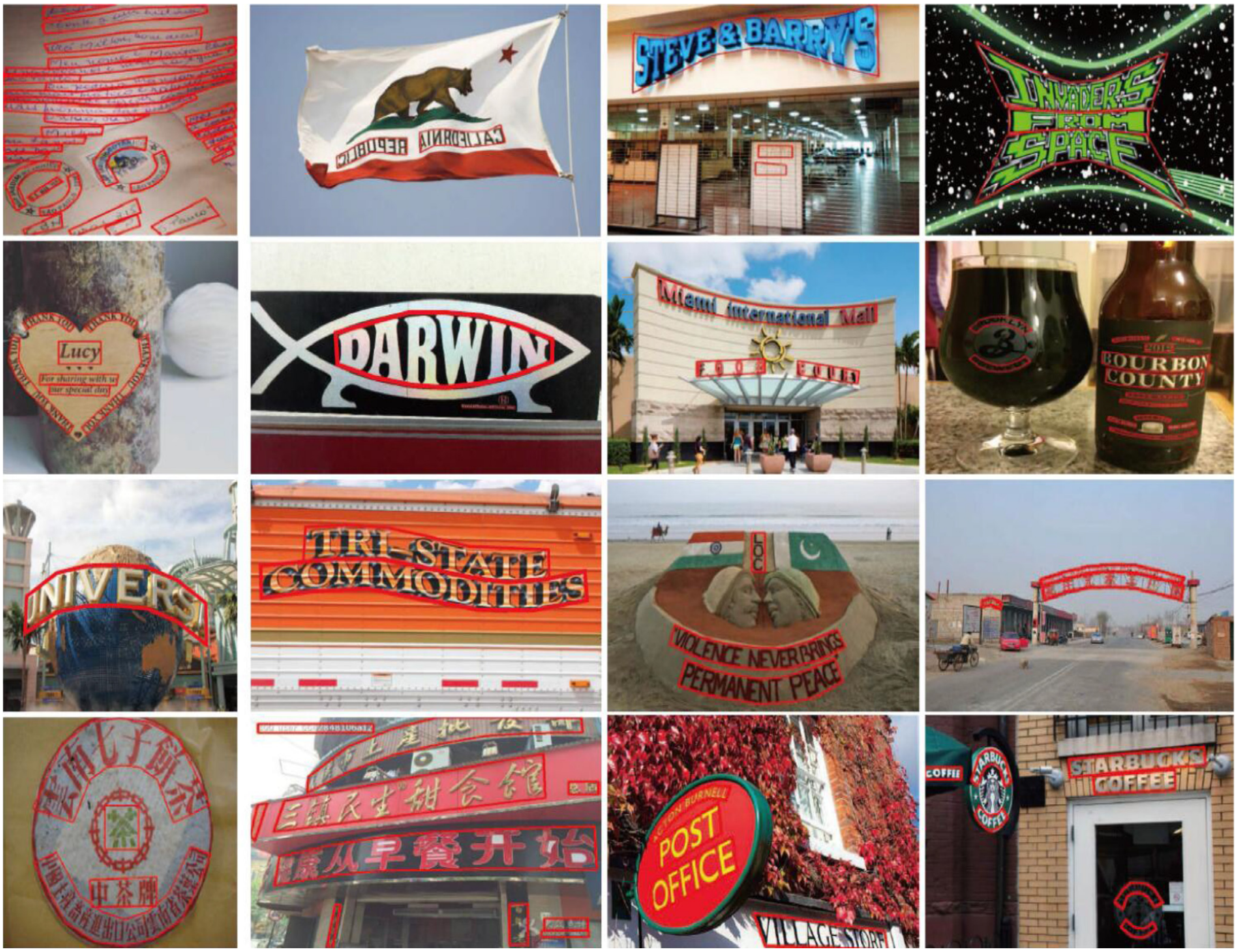


Fig. 2. Examples of annotations in the CTW1500 dataset.

on a branch of segmentation. Additionally, Ren et at. [38] demonstrated that training with recognition can be conducive to text detection.

However, current text detection methods, even quadrangle-based methods, all show disappointing performance on curved text, which commonly appears in the real world, as described in Section 1. Linking methods, such as the method in [14], cannot detect heavily curved text. One reason for this is that most current datasets contain very little curved text and much of the curved text is labeled with unsatisfactory rectangles. The other reason is that current four-points-based detection methods can only loosely detect curved text, which may cause severe mutual interference, as shown in Fig. 1(c). To address the challenging problem of detecting curved text in the real world, we constructed a curved text dataset called CTW1500, and then proposed a novel method that can directly detect the curved text.

3. CTW1500 Dataset and annotation

Data description. The CTW1500 dataset contains 1500 images, with 10,751 bounding boxes (3,530 are curved bounding boxes) and at least one curved text per image. The images were manually extracted from the Internet, image libraries, such as Google Open-Image [16], and our own data collected via phone cameras, which also contain a large amount of horizontal and multi-oriented text. The distribution contains indoor, outdoor, born digital, blurred, perspective distortion text and other text. Our dataset is multi-lingual, containing mostly Chinese and English text.

Annotation. The text was manually labeled by using a labeling tool. For labeling text with a horizontal or quadrilateral shape, only two or four clicks are required, respectively. To surround curved text, we created 10 equidistant reference lines to label 10 additional points. In practice, we found that 10 extra points were sufficient to label all types of curved text, as shown in Fig. 2. The reason we used equidistant lines was to ease the labeling effort and reduce subjective interference. To evaluate the localization performance, we simply follow the PASCAL VOC protocol [39], which uses a 0.5 intersection-over-union (IoU) threshold to determine true or false positives. The only difference was that we calculated the exact IoU between the polygons instead of axis-aligned rectangles.

The labeling procedure is illustrated in Fig. 3. First, we click the four vertexes marked as 1, 2, 3, and 4, and the dashed reference line (blue) is created automatically. Then, we move one of the reference lines (horizontal and vertical black dashed lines) to the appropriate position (intersection of two lines) and click to determine the next point. This is repeated for the remaining points. We roughly calculate the labeling time for the three shapes of text in Table 1, which shows that labeling one curved text consumed approximately triple the time required for labeling with a quadrangle. The CTW1500 dataset and the labeling tool can both be downloaded at https://github.com/Yuliang-Liu/Curve-Text-Detector.

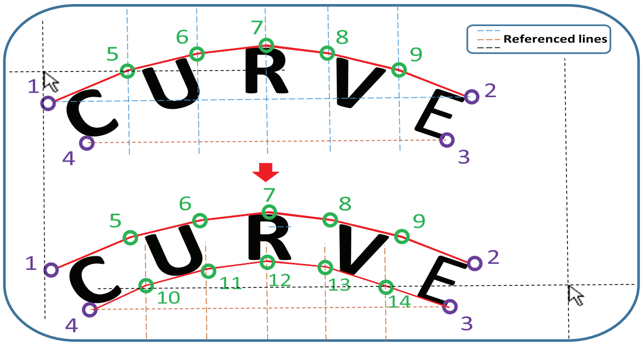


Fig. 3. Illustration of labeling curved text.

Table 1
Time cost of labeling different shapes of text.

Bounding Box	Horizontal	Quadrilateral	Curve
Labeling Time (s)	2.5	4	13

4. Methodology

This section presents details of the CTD. First, we illustrate the architecture of the CTD and describe how we make use of polygonal labels. Next, we describe how an RNN component is seamlessly connected to the CTD and discuss the associated universality of this method. Finally, we present two simple but effective post-processing methods that can further improve the text detection performance.

4.1. Network architecture

The overall architecture of our CTD is illustrated in Fig. 4 and can be divided into three sections: the backbone, region proposal network (RPN), and regression module. The backbone typically adopts popular models pre-trained from ImageNet [40] and then uses the corresponding models for finetuning, such as VGG-16 [41] and ResNet [19]. The RPN and regression module are connected to the backbone, where the former generates proposals for roughly recalling text and the latter adjusts the proposals to make it tighter.

In this paper, we used a reduced ResNet-50 (we simply removed the final residual block) as our backbone, because it requires less memory and is typically faster. In the RPN stage, we used default rectangular anchors to roughly recall text, but we set a very loose non-maximum suppression (NMS) [42] threshold in the RPN stage to avoid premature suppression. To detect the curved text with a polygonal label, the CTD only needs to modify the regression module by adding curved locating points; this is inspired by DMPNet [8] and East [33], which both adopted quadrilateral regressing branches separated by circumscribed rectangle regressions. A rectangular branch can be easily learned by the network and allows it to converse quickly. It can also roughly detect the text region in advanced and aid in the following regression. In contrast, the quadrilateral branch offers stronger supervision to guide the network to greater accuracy.

Similar to [8,43], we also regress the relative positions for each points. Unlike [8], we use the minimum x and minimum y of the circumscribed rectangle as the datum point. Therefore, the relative length w_i and h_i ($i \in 1, 2, \ldots, 14$) of every point is greater than zero, which makes the network easier to train in practice. Additionally, we separately predict the offsets w and h, which not only reduces the number of parameters but is also more reasonable for sequential learning, as discussed in the following subsection. The total number of regression items is 32; where 28 are the offsets of the 14 points and 4 are the (x, y) minimum and maximum of the circumscribed rectangle. The parameterizations of the 14 offsets (d_{w_i} and d_{h_i}) are defined as:

$$\begin{cases} d_{w_i} = \frac{p^*_{w_i} - p_{w_i}}{w_{chr}}, \\ d_{h_i} = \frac{p^*_{h_i} - p_{h_i}}{h_{chr}}, \end{cases} \quad (i \in (1, 2, \ldots, 14)) \tag{1}$$

where p^* and p are ground truth and predicted offsets, respectively. Additionally, w_{chr} and h_{chr} are the width and height of the circumscribed rectangle. For boundary regression, we follow the same as the Faster R-CNN [43]. It is worth noting that 28 values are sufficient to determine the position of 14 points, but in relative regression mode, 32 values can make it easier to retrieve the 14 points and offer stronger supervision.

4.2. Recurrent transverse and longitudinal offset connection (TLOC)

Using recurrent connection in the text detection task was demonstrated to be robust and effective by CTPN [13], which learns the latent sequences of small proposals and produces superior results. However, CTPN is a linking based method that requires empirical connection. Additionally, its connectionist proposal requires fixed image sizes to ensure a fixed number of input time sequences for its RNN. Unlike CTPN, our method can directly localize a curved region without an exterior connection and the number of time sequences for the RNN is not constrained by the input image size. This is because the RNN is connected to the output of the position-sensitive RoI Pooling (PSROIPooling) [44] module and the number of output targets is fixed (14 width offsets and 14 height offsets). PSROIPooling is used to predict and vote for the class probabilities and localization offsets, and evenly partitions each RoI into $p \times p$ bins to estimate position information. The dimension of the input convolutional layer should be $(class + 1)p^2$, meaning the PSROIPooling can produce a p^2 score map for each category. For the classification branch, the class represents the number of foreground classes (In our case, it is one because we have only one foreground class: “Text.”). For offset regression branches, the class is the number of offsets (four for a boundary regression branch and 14 for offset regression branches.) For transverse and longitudinal offset prediction, we remove the background class localization score map and use a 7×7 bin, and thus the input convolutional dimension is $14 \times 7 \times 7$ for height and width, separately. Each value of the (i; j)-th bin ($0 < i$; $j < p - 1$) is computed from the corresponding position in the (i; j)-th score map by using an average pooling operation:

$$r_c(i, j|\Theta) = \sum_{(x,y) \in bin(i,j)} \frac{s_{i,j,c}(x + x_{min}, y + y_{min}|\Theta)}{n}, \tag{2}$$

where $r_c(i, j|\Theta)$ is the pooled value in the (i, j)-th bin for category c, and $s_{i,j,c}$ represents a score map for the corresponding dimension. $(x_{\min}, y_{\min})$ denotes the left-top coordinate of an RoI, n denotes the number of pixels in the bin, and Θ represents all the network parameters. After the PSROIPooling procedure, the CTD will receive the scores or estimated offsets for each RoI via global pooling on the p^2 position-sensitive score maps: $r_c(\Theta) = \sum \frac{r_c(i,j|\Theta)}{p^2}$, which produces a $(C + 1)$-dimensional vector. The voting class score is then computed by the softmax operation across all categories and the final confidence is outputted as follows:

$$s_c(\Theta) = e^{r_c(\Theta)} \Big/ \sum_{c'=0}^{C} e^{r_{c'}(\Theta)}. \tag{3}$$

The localization offsets are then fed into a localization loss function. During training phase, we choose similar multi-task loss functions for score and offset prediction as follows:

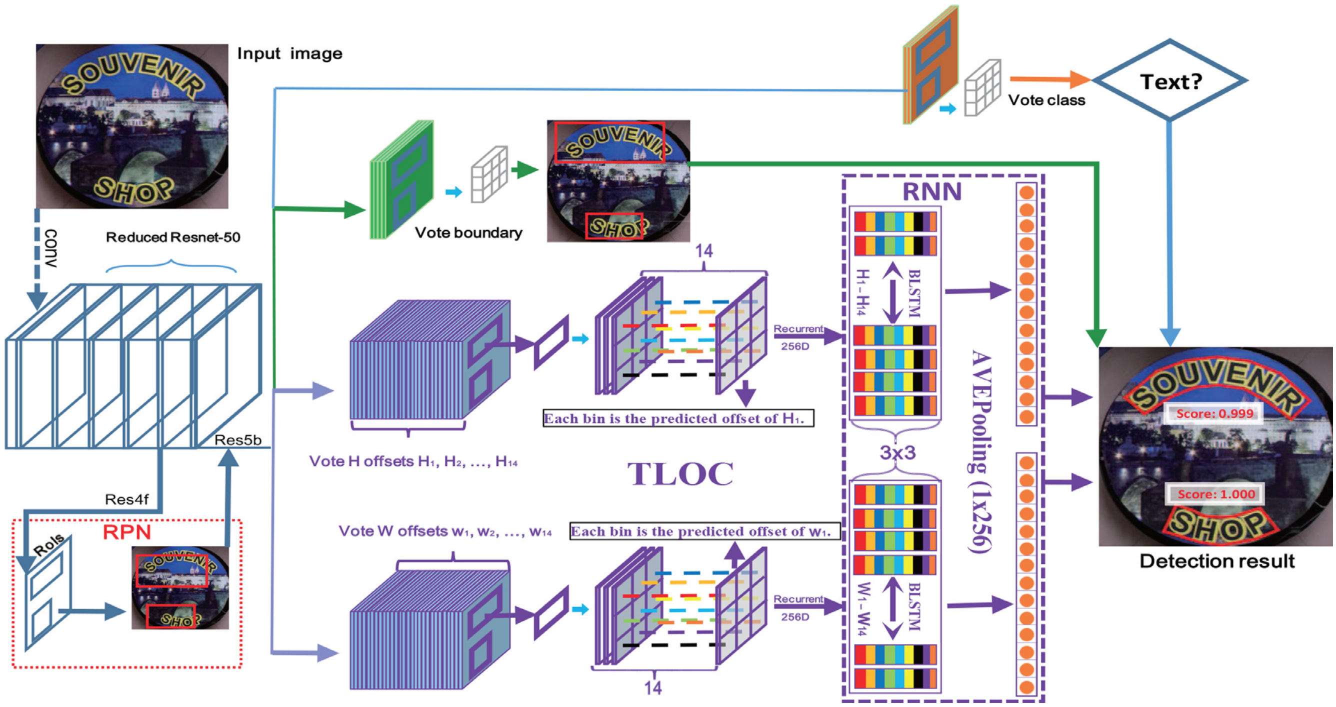


Fig. 4. Overall structure of our Curve Text Detector (CTD).

$$L(c, c^*, b, b^*, w, w^*, h, h^*) = \frac{1}{N}(\lambda \times L_{soft}(c, c^*) + L_{loc}(b, b^*)) + \frac{\mu}{N_p}(L_{loc}(h, h^*) + L_{loc}(w, w^*)) \quad (4)$$

where N is the number of both positive and negative proposals that match a specific overlapping range, and N_p is only the number of positive proposals because it is not necessary to refine negative proposals. Additionally, λ and μ are balance factors that weight the importance of the classification and detection losses (L_{soft} represents the SoftMax loss function and L_{loc} is a localization loss function that can be a smooth-$L1$ loss or smooth-Ln [8] loss). In practice, we set λ to 3 or more to balance localization loss, which has many more targets. Furthermore, (c, b, w, h) represent the predicted class, estimated bounding-box, and width and height offset, respectively, and (c^*, b^*, w^*, h^*) denote the corresponding ground-truth.

To improve the detection performance, we separate the transverse and longitudinal branches to predict the offsets for localizing the text region. Intuitively, each point is restricted by the previous and subsequent points, as well as the textual region. For example, in the case of Fig. 3, the offset width of the sixth labeling point should be larger than that of the fifth point and smaller than that of the seventh point. Independently predicting each offset may lead to a rough text region and result in false detection. Therefore, we assume that the width and height of each point have associated contextual information [45] and use the RNN to learn their latent characteristics. We refer to this method as recurrent transverse and longitudinal offset connection (TLOC). The TLOC structure is illustrated by the purple patterns in Fig. 4. We also provide some examples in Fig. 5 to illustrate the results of introducing TLOC. To adopt TLOC, we must determine if the output of PSROIPooling is suitable for encoding the offsets contextual information. Considering the width offset branch as an example, first, PSROIPooling outputs the 14 p^2 score map for voting $w_1, \ldots, w_{14}$ for each proposal. The p^2 bins of the i-th score map have p^2 voting values from each respective position, which can be encoded as a feature of w_i. The RNN then takes the width offsets feature of each point as sequential inputs and recurrently updates the inherent state inside the hidden layers L_t as follows:

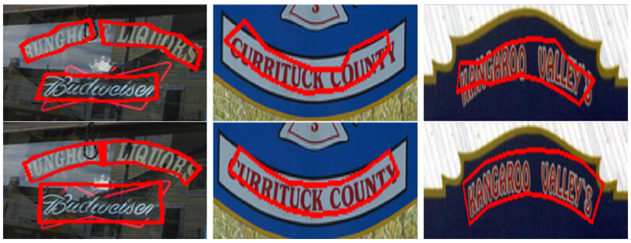


Fig. 5. Top: Detection results without TLOC. Bottom: Detection results with TLOC.

$$L_t = \varphi(L_{t-1}, O_t), t = 1, 2, \ldots, 14 \quad (5)$$

where $O_t \in \Re^{p^2}$ is the t-th prediction offset from the corresponding PSROIPooling output channel. L_t is a recurrent internal state computed from both the current input (O_t) and the previous state encoded in L_{t-1}. The recurrence is computed by using a nonlinear function φ. We adopt a bidirectional long short-term memory (BLSTM) architecture [46] for our RNN. The internal state inside the RNN hidden layer associates the sequential contextual information with all previous estimated offsets through the recurrent connection, and we empirically use a 256D BLSTM hidden layer, meaning $L_t \in \Re^{256}$. The final output of the BLSTM is a 14-dimensional 1×256 vector, which is globally pooled by a (1×256) kernel to output the final prediction.

4.3. Polygonal post processing

Non-polygon suppression (NPS). False positive detection results are one of the major factors impacting the performance of text detection. However, in our CTD, some false positives appear as invalid shapes (for a valid polygon, there are only two intersections at both ends of the side). Additionally, very little scene text appears intersecting the side, and these invalid polygons are nearly impossible to recognize. Therefore, we simply suppress all invalid

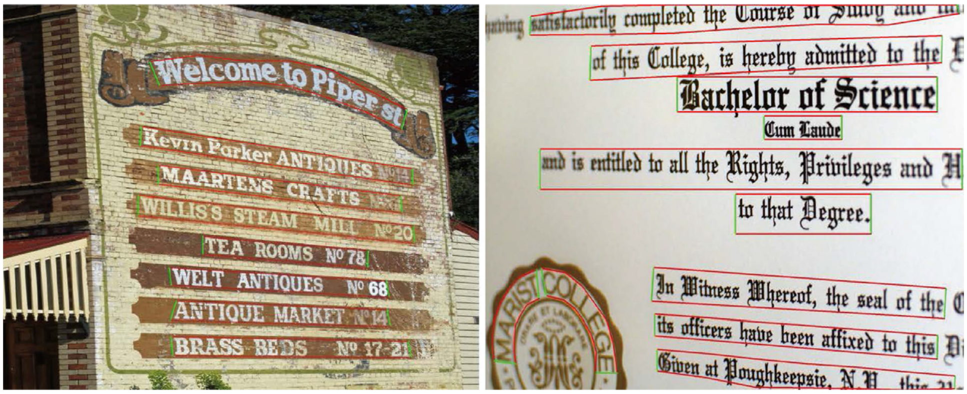

Welcome to Piper st
Kevin Parker ANTIQUES
MAARTENS CRAFTS
WILLIS'S STEAM MILL Nº20
TEA ROOMS Nº78
WELT ANTIQUES Nº 68
ANTIQUE MARKET Nº14
BRASS BEDS Nº 17-21
satisfactorily completed the Course of Study and
of this College, is hereby admitted to the
Bachelor of Science
Cum Laude
and is entitled to all the Rights, Privileges and
to that Degree.
In Witness Whereof, the seal of the
its officers have been affixed to this
Given at Poughkeepsie, N.Y.
MARIST COLLEGE

Fig. 6. Visualization of the interpolation for 4 points bounding boxes. The 10 equal division points will be respectively interpolated in the two **Red** sides of each bounding box. **Green** represents a straight line without interpolation.

polygons and we refer to this method as non-polygon suppression (NPS), which can slightly improve accuracy without influencing the recall rate.

Polygonal non-maximum suppression (PNMS). Non-maximum suppression (NMS) [42] has proved to be very effective for the object-detection task. Because of the particularity of curved scene text, rectangular NMS is limited when it comes to handling dense multi-oriented text, as shown in Fig. 1 (a) and (c). To solve this problem, [33] proposed locality-aware NMS and [35] devised Mask-NMS to suppress the final output results. In this paper, we also improved NMS by computing the overlapping area between polygons. We refer to our method as PNMS, the effectiveness of which was demonstrated in the following experiments.

5. Experiments

In this section, we carried out experiments on the curved and non-curved datasets to test our method. The testing environment was an Ubuntu 16.04 64bit system with a single Nvidia 1080 GPU. All text detection results were evaluated using the protocol introduced in Section 3, which exactly calculates the IoU between the polygons instead of axis-aligned rectangles. All predicted bounding boxes are first suppressed by NPS and then post-processed by the PNMS method. The remaining bounding boxes are regarded as the final detection results.

As described in Section 4.1, for each bounding box, the CTD outputs offsets for 14 points. However, nearly all current benchmark datasets only have labels for two or four vertexes information. To train our method using these datasets, we can easily interpolate the equidistant points along the largest side and its opposite side as shown in Fig. 6.

By simply interpolating the points of the bounding box, our CTD can be effectively trained with all text region.

Implementing details. During the training phase, the learning rate was set to 0.001, the momentum was 0.9, and the batch size was set to 1. In the RPN phase, the NMS threshold was set to 0.5, the overlapping threshold of the positive proposal was set to 0.7, and the negative overlapping threshold was set to 0.3. The batch size of the RPN was 256. In the curved datasets, only the provided training data was used without any data augmentation. On the MSRA-TD500 dataset, because its training set contains only 300 images, which are not sufficient to train our model, therefore, we trained the model with synthetic data created following the method in [50] and finetuned on its training set. The scale of test image was set to 600, and the max size of the image is restricted to 1000. All the regression targets in the polygonal regression branch have the same learning weight, which was set to 1.0 in this paper.

Table 2
Evaluating TLOC and PNMS on the proposed dataset. Here, we use NMS because CTD remains the circumscribed rectangle branch, and the float number is the threshold. R: Recall rate. P: Precision. H: Hmean. The float numbers after PNMS and NMS represent the value of the thresholds.

Algorithm	R (%)	P (%)	H (%)
CTD + NMS0.3	64.4	74.9	69.3
CTD + PNMS0.1	65.2	74.3	69.5
CTD + TLOC + PNMS0.1	69.8	77.4	**73.4**
CTD + TLOC + PNMS0.2	70.1	75.0	72.4
CTD + TLOC + PNMS0.3	70.8	71.6	71.2
CTD + TLOC + PNMS0.4	71.7	65.3	68.3
CTD + TLOC + NMS0.1	63.7	78.4	70.3
CTD + TLOC + NMS0.2	68.6	78.1	73.1
CTD + TLOC + NMS0.3	69.7	77.1	73.2
CTD + TLOC + NMS0.4	70.8	74.7	72.7

5.1. Curved datasets

5.1.1. CTW1500

Effectiveness of TLOC and PNMS. We first evaluated the performance of the proposed TLOC and PNMS on the dataset. The results are listed in Table 2, which shows that irrespective of whether CTD or CTD + TLOC was used, the PNMS slightly outperformed classic NMS. However, the results also demonstrate that by simply adding TLOC, the proposed CTD could be improved by approximately 4% in terms of the harmonic mean (Hmean). Note that we did not compare NPS here because it is a prerequisite post-processing method for evaluating the polygonal overlapping area, which can slightly improve the accuracy.

Comparison with state-of-the-arts methods. To comprehensively evaluate our method, we compared the proposed CTD and CTD + TLOC methods with several state-of-the-art and well-known text detection methods. Note that for East [33], we re-implement the method based on the unofficial source codes from GitHub. For DMPNet [8] and CTPN [13], we used the Caffe [47] framework for re-implementation. Additionally, because none of these methods can be trained using curved text regions, we used traditional circumscribed rectangular bounding boxes to create the labels trainable for these methods. Table 3 lists the experimental results. The results on the entire CTW-1500 test set reveal that the proposed CTD + TLOC can outperform state-of-the-art methods by more than 10% in terms of Hmean. To further evaluate the effectiveness of our method, we also split the curved and non-curved text in the test set by simply considering the other type of text as difficult ("don't care") and making a comparison. For the curved subset, the proposed CTD + TLOC can outperform state-of-the-art methods

Table 3
Experiments on the proposed CTW1500 test set, curved subset, and non-curved subset. R: Recall. P: Precision. H: Hmean. S: Speed. The speed column evaluates the time required for the forward procedure without post-processing.

Algorithm	Entire set			Non-curved subset			Curved subset			S (FPS)
	R (%)	P (%)	H (%)	R (%)	P (%)	H (%)	R (%)	P (%)	H (%)	
SWT [48]	9.0	20.7	12.5	5.8	13.4	8.1	6.4	7.0	6.7	-
CTPN [13]	53.8	60.4	56.9	59.4	54.3	56.7	37.7	34.1	35.8	7.14
EAST [33]	49.1	78.7	60.4	57.5	71.0	63.6	29.9	40.9	34.6	**21.2**
DMPNet [8]	56.0	69.9	62.2	61.7	63.9	62.7	39.3	35.5	37.3	12.3
AdaBoost [49]	4.4	6.7	5.3	-	-	-	-	-	-	-
CTD (ours)	65.2	74.3	69.5	60.3	67.3	63.5	73.9	52.9	61.6	15.2
CTD + TLOC (ours)	69.8	77.4	**73.4**	62.3	70.8	**66.3**	77.1	57.1	**65.6**	13.3

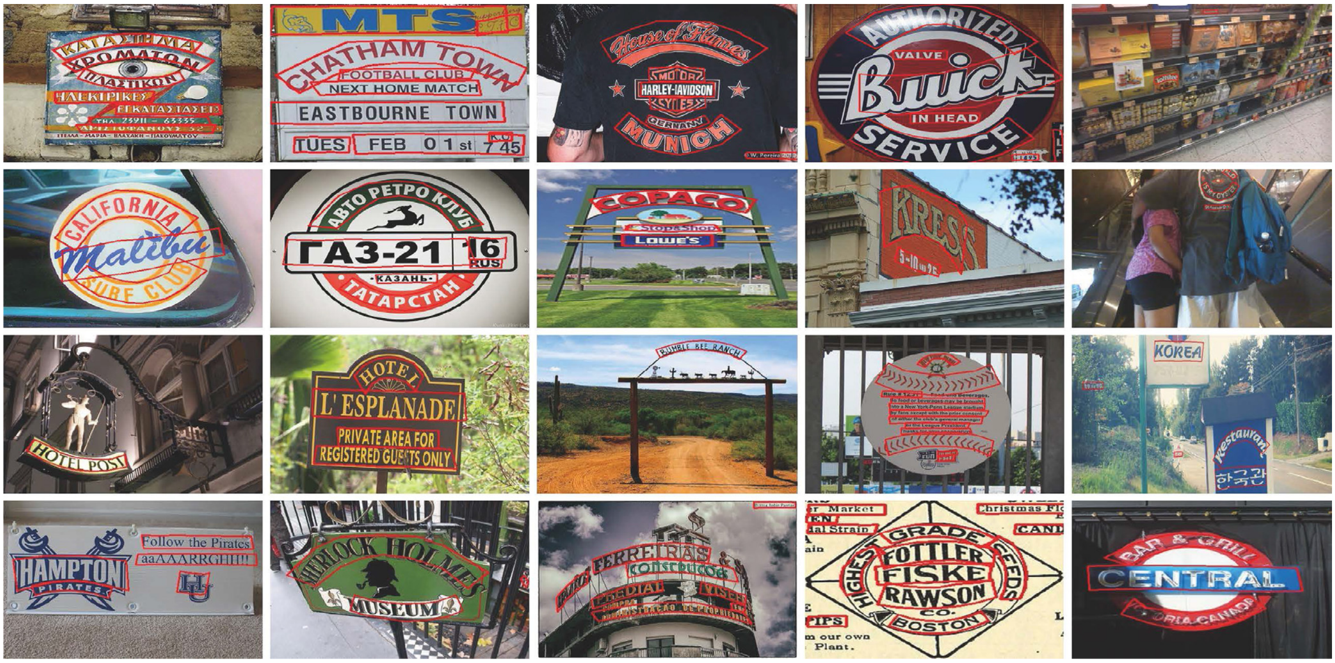


Fig. 7. Visualization of detection results. The fourth column contains some inferior results and the images in the last column come from other datasets for further tests on the generalization ability of our CTD.

by at least 28% in terms of Hmean. Additionally, our method also achieved the best results for detecting non-curved text, demonstrating its robustness and universality.

We also compared the detection speed in the final column of Table 3 and the results (13.3 or 15.2 FPS) revealed that our method is the second-fastest method, which further demonstrates its effectiveness.

Examples of the detection results are illustrated in Fig. 7. From the results, we can find the detections can automatically adaptive to any kind of text, even the distances between characters are different. Note that all detection results of rectangular-shaped text bounding boxes are also strictly 14 points. In the final column in this figure, we used our CTD to detect the curved text in other dataset, which qualitatively demonstrates its powerful ability for detecting curved texts. In addition, as shown in the middle image of the last column in Fig. 7, our method can detect Korean curved text without using Linguistic-corresponding training data, demonstrating its promising generalization ability.

5.1.2. Total-text

We also conducted experiments on another curved dataset (Total-text [18]). Total-text was constructed in late 2017 and collected curve text from various scenes, including complex scenes, such as text-like and low contrast background scenes. Most images in this dataset contain significant non-curved text with at least one curved text, which is closer to the real world. The dataset contains 1555 images (1,255 for training and 300 for testing), and the text is annotated in word-level granularity using polygons. The Total-text dataset contains mainly English. Because the annotated numbers for this dataset are not fixed and the point positions are subjectively determined, we extracted the text boundaries and used a similar LSI method to resample the annotated points. Hence, the point number of each annotating box was set strictly to 14. The results are listed in Table 4, which indicates that our method is significantly better than Ch'ng's method [18].

Table 4
Evaluation on the Total-text dataset.

Algorithm	Recall	Precision	Hmean
Proposed	**0.71**	**0.74**	**0.73**
Ch'ng [18]	0.33	0.40	0.36

5.2. Experiments on regular dataset - MSRA-TD500

To further demonstrate the robustness of our method, we also conducted experiments on the MSRA-TD500 [3] dataset. MSRA-TD500 contains 500 images with multi-oriented English and Chinese scene text, which has been widely evaluated by previous methods. The results of our method are listed in Table 5 along

Table 5
Experimental results on the MSRA-TD500.

Algorithm	Recall (%)	Precision (%)	F-measure (%)
Yao et al. [3]	63.0	63.0	60.0
Zhang et al. [51]	67.0	83.0	74.0
RRPN [52]	68.0	82.0	74.0
He el al [34].	70.0	77.0	74.0
Yao et al. [53]	75.3	76.5	75.9
EAST [33]	67.4	87.3	76.1
SegLink [14]	70.0	86.0	77.0
Proposed [35]	**77.1**	84.5	**80.6**

with other state-of-the-art methods. This table reveals that our method outperformed previous state-of-the-art approaches, especially in terms of recall rate, clearly demonstrating its powerful detection ability.

6. Conclusions and future work

Curved text is common in the real world, but few current datasets or methods are aimed at curved text detection. To highlight this new challenge of reading curved text in the real world, we proposed a new dataset called CTW1500, comprising mainly English and Chinese curved text. The curved text in this dataset is tightly labeled using polygons, which does not require significant manpower. Additionally, we proposed a novel CTD approach that may be the first attempt to directly detect curved text. By implementing our TLOC method, the CTD can be seamlessly connected to RNN, which significantly improves its detection performance. We also introduced a simple but effective long-side-interpolation technique, which allows the CTD to function as a universal method that can be trained using rectangular or quadrilateral bounding boxes without additional manual effort. Finally, we designed two post-processing methods that were demonstrated to be effective. Experiments on curved datasets, including the proposed CTW1500 and Total-text, revealed that the proposed method could outperform state-of-the-art methods by a large margin, demonstrating its powerful ability to detect curved text. On the multi-oriented MSRA-TD500 dataset, our method still achieved the best performance, demonstrating its robustness and impressive generalization ability.

Because the labeling scheme is suitable for recognition, our dataset could be enlarged as a curved-text-based recognition dataset. Additionally, although the flexible detection method used by our CTD may be slightly slower than a rigid rectangle-based detector, our method can solve more complicated problems, such as detecting curved text, and achieve superior results, making it worthy of further exploration.

Acknowledgment

This research is supported in part by the National Key R&D Program of China (No. 2016YFB1001405), GD-NSF (no. 2017A030312006), NSFC (Grant No.: 61472144, 61673182), GDSTP (Grant No.: 2015B010101004, 2015B010130003, 2017A030312006), GZSTP (No.: 201607010227).

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.patcog.2019.02.002.

References

[1] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L.G. i Bigorda, S.R. Mestre, J. Mas, D.F. Mota, J.A. Almazan, L.P. De Las Heras, ICDAR 2013 robust reading competition, in: Proceedings of the ICDAR, 2013, pp. 1484–1493.

[2] K. Wang, S. Belongie, Word spotting in the wild, in: Proceedings of the ECCV, 2010, pp. 591–604.

[3] C. Yao, X. Bai, W. Liu, Y. Ma, Detecting texts of arbitrary orientations in natural images, in: Proceedings of the CVPR, 2012, pp. 1083–1090.

[4] X.C. Yin, W.Y. Pei, J. Zhang, H.W. Hao, Multi-orientation scene text detection with adaptive clustering, IEEE Trans. Pattern Anal. Mach.Intell. 37 (9) (2015) 1930.

[5] D. Karatzas, L. Gomez-Bigorda, A. Nicolaou, S. Ghosh, A. Bagdanov, M. Iwamura, J. Matas, L. Neumann, V.R. Chandrasekhar, S. Lu, et al., ICDAR 2015 competition on robust reading, in: Proceedings of the ICDAR, 2015, pp. 1156–1160.

[6] B.G. Shi, C. Yao, M.H. Liao, M.K. Yang, P. Xu, L.Y. Cui, S. Belongie, S.J. Lu, X. Bai, ICDAR2017 competition on reading Chinese text in the wild (RCTW-17), in: Proceedings of the ICDAR, 2017, pp. 1429–1434.

[7] N. Nayef, F. Yin, I. Bizid, H. Choi, Y. Feng, D. Karatzas, Z.B. 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: Proceedings of the ICDAR, 2017, pp. 1454–1459.

[8] Y. Liu, L. Jin, Deep matching prior network: toward tighter multi-oriented text detection, in: Proceedings of the CVPR, 2017, pp. 3454–3461.

[9] B. Shi, X. Wang, P. Lyu, C. Yao, X. Bai, Robust scene text recognition with automatic rectification, in: Proceedings of the CVPR, 2016, pp. 4168–4176.

[10] B. Shi, X. Bai, C. Yao, An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition, IEEE Trans. Pattern Anal. Mach. Intell. 39 (11) (2017) 2298–2304.

[11] A. Ablavatski, S. Lu, J. Cai, Enriched deep recurrent visual attention model for multiple object recognition, in: Proceedings of the WACV, 2017, pp. 971–978.

[12] C.Y. Lee, S. Osindero, Recursive recurrent nets with attention modeling for Ocr in the wild, in: Proceedings of the CVPR, 2016, pp. 2231–2239.

[13] Z. Tian, W. Huang, T. He, P. He, Y. Qiao, Detecting text in natural image with connectionist text proposal network, in: Proceedings of the ECCV, 2016, pp. 56–72.

[14] B. Shi, X. Bai, S. Belongie, Detecting oriented text in natural images by linking segments, in: Proceedings of the CVPR, 2017.

[15] H. Hu, C. Zhang, Y. Luo, Y. Wang, J. Han, E. Ding, Wordsup: exploiting word annotations for character based text detection, in: Proceedings of the CVPR, 2017, pp. 3482–3490.

[16] I. Krasin, T. Duerig, N. Alldrin, A. Veit, S. Abu-El-Haija, S. Belongie, D. Cai, Z. Feng, V. Ferrari, V. Gomes, et al., Openimages: a public dataset for large-scale multi-label and multi-class image classification, 2016, Dataset available from https://github.com/openimages.

[17] A. Risnumawan, P. Shivakumara, C.S. Chan, C.L. Tan, A robust arbitrary text detection system for natural scene images, Expert Syst. Appl. 41 (18) (2014) 8027–8048.

[18] C.K. Ch'ng, C.S. Chan, Total-text: a comprehensive dataset for scene text detection and recognition, in: Proceedings of the ICDAR, 2017, pp. 935–942.

[19] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the CVPR, 2016, pp. 770–778.

[20] S.M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, R. Young, ICDAR 2003 robust reading competitions, in: Proceedings of the ICDAR, 2003, pp. 682–687.

[21] A. Shahab, F. Shafait, A. Dengel, ICDAR 2011 robust reading competition challenge 2: reading text in scene images, in: Proceedings of ICDAR, 2011, pp. 1491–1496.

[22] A. Veit, T. Matera, L. Neumann, et al., Coco-text: dataset and benchmark for text detection and recognition in natural images. arXiv:1601.07140.

[23] R. Nagy, A. Dicker, K. Meyer-Wegener, NEOCR: a configurable dataset for natural image text recognition, in: International Workshop on Camera-Based Document Analysis and Recognition, 2011, pp. 150–163.

[24] C. Yi, Y. Tian, Text string detection from natural scenes by structure-based partition and grouping, IEEE Trans. Image Process. 20 (9) (2011) 2594–2605.

[25] M. Iwamura, N. Morimoto, K. Tainaka, D. Bazazian, L. Gomez, D. Karatzas, ICDAR2017 robust reading challenge on omnidirectional video, in: Proceedings of the ICDAR, 2017, pp. 1448–1453.

[26] L. Gomez-Bigorda, D. Karatzas, Textproposals: a text-specific selective search algorithm for word spotting in the wild, Pattern Recognit. 70 (2016) 60–74.

[27] L. Sun, Q. Huo, W. Jia, K. Chen, A robust approach for text detection from natural scene images, Pattern Recognit. 48 (9) (2015) 2906–2920.

[28] Q. Ye, D. Doermann, Text detection and recognition in imagery: a survey, IEEE Trans. Pattern Anal. Mach. Intell. 37 (7) (2015) 1480–1500.

[29] V. Khare, P. Shivakumara, P. Raveendran, M. Blumenstein, A blind deconvolution model for scene text detection and recognition in video, Pattern Recognit. 54 (2016) 128–148.

[30] P. Shivakumara, R. Raghavendra, L. Qin, K.B. Raja, T. Lu, U. Pal, A new multi--modal approach to bib number/text detection and recognition in marathon images, Pattern Recognit. 61 (2017) 479–491.

[31] X. Bai, B. Shi, C. Zhang, X. Cai, L. Qi, Text/non-text image classification in the wild with convolutional neural networks, Pattern Recognit. 66 (2017) 437–446.

[32] B.B. Chaudhuri, C. Adak, An approach for detecting and cleaning of struck-out handwritten text, Pattern Recognit. 61 (2017) 282–294.

[33] 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 CVPR, 2017, pp. 2642–2651.

[34] W. He, X.Y. Zhang, F. Yin, C.L. Liu, Deep direct regression for multi-oriented scene text detection, in: Proceedings of the ICCV, 2017.

[35] Y. Dai, Z. Huang, Y. Gao, K. Chen, Fused text segmentation networks for multi-oriented scene text detection, in: Proceedings of the ICCV, 2017.

[36] P. Shivakumara, L. Wu, T. Lu, C.L. Tan, M. Blumenstein, B.S. Anami, Fractals based multi-oriented text detection system for recognition in mobile video images, Pattern Recognit. 68 (2017) 158–174.
[37] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask R-cnn, in: Proceedings of the ICCV, 2017, pp. 2980–2988.
[38] H. Li, P. Wang, C. Shen, Towards end-to-end text spotting with convolutional recurrent neural networks, in: Proceedings of the ICCV, 2017, pp. 5238–5246.
[39] M. Everingham, L.V. Gool, C.K. Williams, J. Winn, A. Zisserman, The pascal visual object classes (voc) challenge, Int. J. Comput. Vis. 88 (2) (2010) 303–338.
[40] J. Deng, W. Dong, R. Socher, L.J. Li, K. Li, L. Fei-Fei, Imagenet: a large-scale hierarchical image database, in: Proceedings of the CVPR, 2009, pp. 248–255.
[41] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: Proceedings of the Computer Science, 2014.
[42] A. Neubeck, L.V. Gool, Efficient non-maximum suppression, in: Proceedings of the ICPR, 2006, pp. 850–855.
[43] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: towards real-time object detection with region proposal networks, in: Advances in Neural Information Processing Systems, 2015, pp. 91–99.
[44] J. Dai, Y. Li, K. He, J. Sun, R-fcn: object detection via region-based fully convolutional networks, in: Proceedings of the NIPS, 2016, pp. 379–387.
[45] A. Zhu, R. Gao, S. Uchida, Could scene context be beneficial for scene text detection? Pattern Recognit. 58 (2016) 204–215.
[46] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
[47] Y. Jia, E. Shelhamer, Caffe: convolutional architecture for fast feature embedding, in: Proceedings of the 22nd ACM International Conference on Multimedia, ACM, 2014, pp. 675–678.
[48] B. Epshtein, E. Ofek, Y. Wexler, Detecting text in natural scenes with stroke width transform, in: Proceedings of the CVPR, 2010, pp. 2963–2970.
[49] X. Chen, A.L. Yuille, Detecting and reading text in natural scenes, in: Proceedings of the CVPR, 2004. II–II.
[50] A. Gupta, A. Vedaldi, A. Zisserman, Synthetic data for text localisation in natural images, in: Proceedings of the CVPR, 2016, pp. 2315–2324.
[51] Z. Zhang, C. Zhang, W. Shen, C. Yao, W. Liu, X. Bai, Multi-oriented text detection with fully convolutional networks, in: Proceedings of the CVPR, 2016, pp. 4159–4167.
[52] J. Ma, W. Shao, H. Ye, L. Wang, H. Wang, Y. Zheng, X. Xue, Arbitrary-oriented scene text detection via rotation proposals, IEEE Trans. Multimedia (2018).
[53] C. Yao, X. Bai, N. Sang, X. Zhou, S. Zhou, Z. Cao, Scene text detection via holistic, multi-channel prediction. arXiv:1606.09002.
[54] M.H. Liao, B.G. Shi, X. Bai, Textboxes++: a single-shot oriented scene text detector, IEEE Trans. Image Process. 27 (8) (2018) 3676–3690.

Yuliang Liu is currently a second-year Ph.D student at the Deep Learning and Vision Computing lab (DLVClab), South China university of Technology, Guangdong, China, and the supervisor is Professor Lianwen Jin. He received the BS degree in electronic and information engineering from South China University of Technology in 2016. He works on scene text understanding, handwritten character recognition, document analysis, deep learning-based text detection and recognition. His team won the champion of the competition for ICDAR 2017 Multi-lingual scene text detection task and end to end scene text detection and classification task. He is also the constructor of the recent SCUT-CTW1500 dataset.

Lianwen Jin received the B.S. degree from the University of Science and Technology of China, Anhui, China, and the Ph.D. degree from the South China University of Technology, Guangzhou, China, in 1991 and 1996, respectively. He is currently a Professor with the School of Electronic and Information Engineering, South China University of Technology. He is the author of more than 100 scientific papers. Dr. Jin was a recipient of the award of New Century Excellent Talent Program of MOE in 2006 and the Guangdong Pearl River Distinguished Professor Award in 2011. His research interests include image processing, handwriting analysis and recognition, machine learning, cloud computing, and intelligent systems.

Shuaitao Zhang received the B.S. degree from school of Information Science and Engineering at the Henan University of Technology, Henan, China in 2017. He is currently pursuing the Master degree in electronic communication engineering at the South China University of Technology, Guangzhou, China. His current research interests include machine learning, natural scene text detection and recognition.

Canjie Luo received the BS degree in electronic and information engineering from South China University of Technology in 2017. He is currently a first-year Ph.D student in DLVC-Lab(laboratory of deep learning and vision calculations) in South China University of Technology and his supervisor is Professor Lianwen Jin. His research interests include image processing, machine learning and deep learning. He works on scene text recognition and layout analysis of documents. He is a member of the team who won the champion of the competition for ICDAR 2017 Multilingual scene text detection task and end-to-end scene text detection and classification task.

Sheng Zhang is currently a second-year Ph.D student at the Deep Learning and Vision Computing laboratory (DLVClab), South China University of Technology, Guangdong, China, and the supervisor is Professor Lianwen Jin. He received the B.S.degree in Communication Engineering, School of Internet of Things Engineering from Hohai University, Nanjing, China, in 2014. He works on scene text understanding, object tracking, deep learning, and image processing algorithms.