Arbitrary Shape Scene Text Detection with Adaptive Text Region Representation

Xiaobing Wang[1], Yingying Jiang[1], Zhenbo Luo[1], Cheng-Lin Liu[2, 3], Hyunsoo Choi[4], Sungjin Kim[4]

[1]Samsung Research China - Beijing, Beijing 100028, China
[2]National Laboratory of Pattern Recognition
Institute of Automation of Chinese Academy of Sciences, Beijing 100190, China
[3]University of Chinese Academy of Sciences, Beijing 100049, China
[4]Samsung Research, Seoul 06765, Korea

{x0106.wang, yy.jiang, zb.luo, hsu.choi, sj9373.kim}@samsung.com, liucl@nlpr.ia.ac.cn

Abstract

Scene text detection attracts much attention in computer vision, because it can be widely used in many applications such as real-time text translation, automatic information entry, blind person assistance, robot sensing and so on. Though many methods have been proposed for horizontal and oriented texts, detecting irregular shape texts such as curved texts is still a challenging problem. To solve the problem, we propose a robust scene text detection method with adaptive text region representation. Given an input image, a text region proposal network is first used for extracting text proposals. Then, these proposals are verified and refined with a refinement network. Here, recurrent neural network based adaptive text region representation is proposed for text region refinement, where a pair of boundary points are predicted each time step until no new points are found. In this way, text regions of arbitrary shapes are detected and represented with adaptive number of boundary points. This gives more accurate description of text regions. Experimental results on five benchmarks, namely, CTW1500, TotalText, ICDAR2013, ICDAR2015 and MSRA-TD500, show that the proposed method achieves state-of-the-art in scene text detection.

1. Introduction

Text is the most fundamental medium for communicating semantic information. It appears everywhere in daily life: on street nameplates, store signs, product packages, restaurant menus and so on. Such texts in natural environment are known as scene texts. Automatically detecting and recognizing scene texts can be very rewarding with numerous applications, such as real-time text translation, blind person assistance, shopping, robots, smart cars and education. An end-to-end text recognition system usually consist-

s of two steps: text detection and text recognition. In text detection, text regions are detected and labeled with their bounding boxes. And in text recognition, text information is retrieved from the detected text regions. Text detection is an important step for end-to-end text recognition, without which texts can not be recognized from scene images. Therefore, scene text detection attracts much attention these years.

While traditional optical character reader (OCR) techniques can only deal with texts on printed documents or business cards, scene text detection tries to detect various texts in complex scenes. Due to complex backgrounds and variations of font, size, color, language, illumination condition and orientation, scene text detection becomes a very challenging task. And its performance was poor when hand designed features and traditional classifiers were used before deep learning methods become popular. However, the performance has been much improved in recent years, significantly benefitted from the development of deep learning. Meanwhile, the research focus of text detection has shifted from horizontal scene texts [10] to multi-oriented scene texts [9] and more challenging curved or arbitrary shape scene texts [19]. Therefore, arbitrary shape scene text detection is focused on in this paper.

In this paper, we propose an arbitrary shape scene text detection method using adaptive text region representation, as shown in Figure 1. Given an input image, a text region proposal network (Text-RPN) is first used for obtaining text proposals. The Convolutional Neural Network (CNN) feature maps of the input image are also obtained in this step. Then, text proposals are verified and refined with a refinement network, whose input are the text proposal features obtained by using region of interest (ROI) pooling to the CNN feature maps. Here, three branches including text/non-text classification, bounding box refinement and recurrent neural network (RNN) based adaptive text region representation exist in the refinement network. In the RNN, a pair

of boundary points are predicted each time step until the stop label is predicted. In this way, arbitrary shape text regions can be represented with adaptive number of boundary points. For performance evaluation, the proposed method is tested on five benchmarks, namely, CTW1500, TotalText, ICDAR2013, ICDAR2015 and MSRA-TD500. Experimental results show that the proposed method can process not only multi-oriented scene texts but also arbitrary shape scene texts including curved texts. Moreover, it achieves state-of-the-art performances on the five datasets.

2. Related work

Traditional sliding window based and Connected component (CC) based scene text detection methods had been widely used before deep learning became the most promising machine learning tool. Sliding window based methods [27, 32] move a multi-scale window over an image and classify the current patch as text or non-text. CC based methods, especially the Maximally Stable Extremal Regions (MSER) based methods [26, 30], get character candidates by extracting CCs. And then, these candidate CCs are classified as text or non-text. These methods usually adopt a bottom-up strategy and often need several steps to detect texts (e.g., character detection, text line construction and text line classification). As each step may lead to misclassification, the performances of these traditional text detection methods are poor.

Recently, deep learning based methods have become popular in scene text detection. These methods can be divided into three groups, including bounding box regression based methods, segmentation based methods, and combined methods. Bounding box regression based methods [5, 8, 11, 12, 13, 16], which are inspired by general object detection methods such as SSD [14] and Faster R-CNN [23], treat text as a kind of object and directly estimate its bounding box as the detection result. Segmentation based methods [3, 19, 33] try to solve the problem by segmenting text regions from the background and an additional step is needed to get the final bounding boxes. Combined methods [20] use a similar strategy as Mask R-CNN [4], in which both segmentation and bounding box regression are used for better performance. However, its processing time is increased because more steps are needed than previous methods. Among the three kinds of methods, bounding box regression based methods are the most popular in scene text detection, benefitted from the development of general object detection.

For bounding box regression based methods, they can be divided into one-stage methods and two-stage methods. One-stage methods including Deep Direct Regression [5], TextBox [12], TextBoxes++ [11], DMPNet [16], SegLink [24] and EAST [34], directly estimate bounding boxes of text regions in one step. Two-stage methods include R2CNN [8], RRD [13], RRPN [22], IncepText [28] and FEN [31]. They consist of text proposal generation stage, in which candidate text regions are generated, and bounding box refinement stage, in which candidate text regions are verified and refined to generate the final detection result. Two-stage methods usually achieve higher performances than one-stage methods. Therefore, the idea of two-stage detection is used in this paper.

While most proposed scene text detection methods can only deal with horizontal or oriented texts, detecting arbitrary shape texts such as curved text attracts more attention recently. In CTD [17], a polygon of fixed 14 points are used to represent text region. Meanwhile, recurrent transverse and longitudinal offset connection (TLOC) is proposed for accurate curved text detection. Though a polygon of fixed 14 points is enough for most text regions, it is not enough for some long curve text lines. Besides, 14 points are too many for most horizontal and oriented texts, while 4 points are enough for these texts. In TextSnake [19], a text instance is described as a sequence of ordered, overlapping disks centered at symmetric axes of text regions. Each disk is associated with potentially variable radius and orientation, which are estimated via a Fully Convolutional Network (FCN) model. Moreover, Mask TextSpotter [20] which is inspired by Mask R-CNN, can handle text instances of irregular shapes via semantic segmentation. Though TextSnake and Mask TextSpotter both can deal with text of arbitrary shapes, pixel-wise predictions are both needed in them, which need heavy computation.

Considering a polygon of fixed number of points is not suitable for representing text regions of different shapes, an adaptive text region representation using different numbers of points for texts of different shapes is proposed in this paper. Meanwhile, a RNN is employed to learn the adaptive representation of each text region, with which text regions can be directly labeled and pixel-wise segmentation is not needed.

3. Methodology

Figure 1 shows the flowchart of the proposed method for arbitrary shape text detection, which is a two-stage detection method. It consists of two steps: text proposal and proposal refinement. In text proposal, a Text-RPN is used to generate text proposals of an input image. Meanwhile, the CNN feature maps of the input image are obtained here, which can be used in the following. Then, text proposals are verified and refined through a refinement network. In this step, text/non-text classification, bounding box regression and RNN based adaptive text region representation are included. Finally, text regions labeled with polygons of adaptive number of points are output as the detection result.
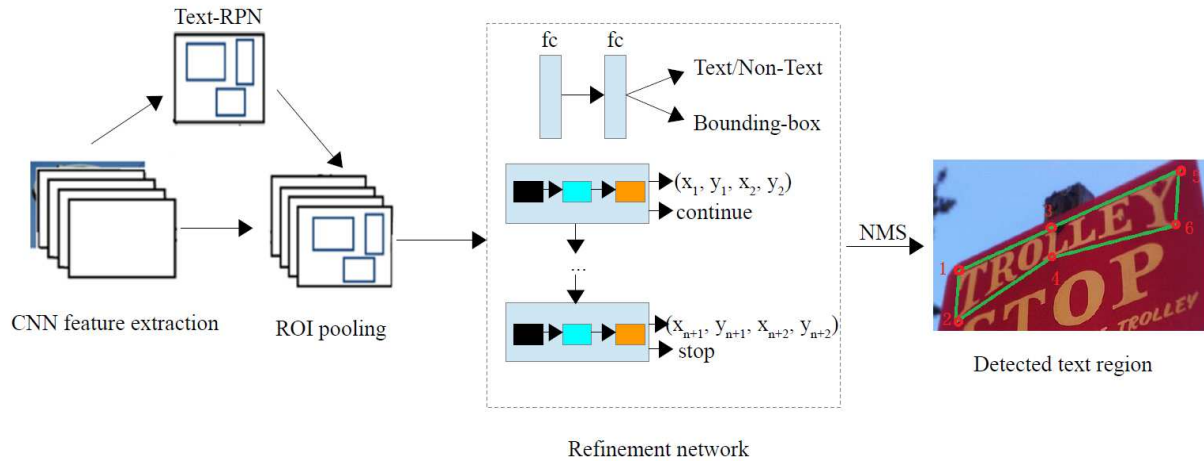
Figure 1. Flowchart of the proposed method for arbitrary shape scene text detection. With adaptive text region representation used, detected text regions can be labeled with adaptive number of pairwise points.

3.1. Adaptive text region representation

The existing scene text detection methods use polygons of fixed number of points to represent text regions. For horizontal texts, 2 points (left-top point and bottom-right point) are used to represent the text regions. For multi-oriented texts, the 4 points of their bounding boxes are used to represent these regions. Moreover, for curved texts, 14 points are adopted in CTW1500 [17] for text region representation. However, for some very complex scene texts, such as curved long text, even 14 points may be not enough to represent them well. While for most scene texts such as horizontal texts and oriented texts, less than 14 points are enough and using 14 points to represent these text regions is a waste.

Therefore, it is reasonable to consider using polygons of adaptive numbers of points to represent text regions. Easily, we can imagine that corner points on the boundary of a text region can be used for region representation, as shown in Figure 2 (a). And this is similar as the method for annotating general objects [1]. However, the points in this way are not arranged in a direction and it may be difficult to learn the representation. In the method for annotating general objects, human correction may be needed for accurate segmentation. Considering text regions usually have approximate symmetry top boundary and down boundary as shown in Figure 3, using the pairwise points from the two boundaries for text region representation may be more suitable. It is much easier to learn the pairwise boundary points from one end to the other end of text region, as shown in Figure 2 (b). In this way, different scene text regions can be represented by different numbers of points precisely, as shown in Figure 3. Moreover, to our knowledge, we are the first to use adaptive numbers of pairwise points for text

region representation.

Figure 2. Two methods for adaptive text region representation. (a) text region represented by corner points; (b) text region represented by pairwise points on its top and down boundaries.
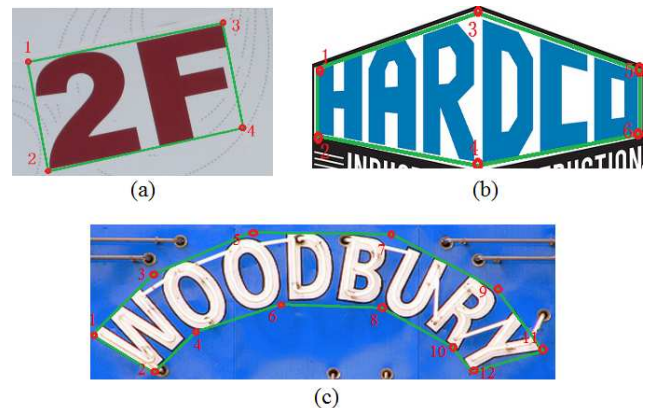

Figure 3. Samples of text regions with adaptive representation. (a) text region represented by 4 points; (b) text region represented by 6 points; (c) text region represented by 12 points.

3.2. Text proposal

When an input image is given, the first step of the proposed method is text proposal, in which text region candidates called text proposals are generated by Text-RPN. The Text-RPN is similar as RPN in Faster R-CNN [23] except different backbone networks and anchor sizes. In the proposed method, the backbone network is SE-VGG16 as shown in Table 1, which is obtained by adding Squeeze-and-Excitation (SE) blocks [7] to VGG16 [25]. As shown in Figure 4, SE blocks adaptively recalibrate channel-wise feature responses by explicitly modelling interdependencies between channels, which can produce significant performance improvement. Here, FC means fully connected layer and ReLU means Rectified Linear Unit function. Moreover, because scene texts usually have different sizes, anchor sizes are set as $\{32, 64, 128, 256, 512\}$ for covering more texts while aspect ratios $\{0.5, 1, 2\}$ are kept.

Layer	Kernel
Conv1	$[3 \times 3, 64] \times 2$
Pool1	2×2, stride 2
SE1	4, 64
Conv2	$[3 \times 3, 128] \times 2$
Pool2	2×2, stride 2
SE2	8, 128
Conv3	$[3 \times 3, 256] \times 3$
Pool3	2×2, stride 2
SE3	16, 256
Conv4	$[3 \times 3, 512] \times 3$
Pool4	2×2, stride 2
SE4	32, 512
Conv5	$[3 \times 3, 512] \times 3$
Pool5	2×2, stride 2
SE5	32, 512

Table 1. The architecture of SE-VGG16 network. For SE block, its kernel means the channel numbers of the two FC layers in it.
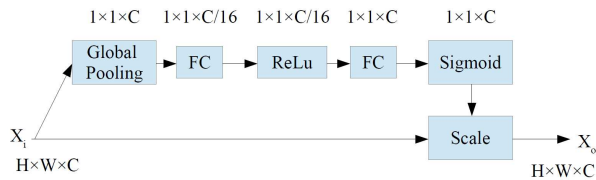

Figure 4. The architecture of SE block.

3.3. Proposal refinement

After text proposal, text region candidates in the input image are generated, which will be verified and refined in this step. As shown in Figure 1 a refinement network is employed for proposal refinement, which consists of several

branches: text/non-text classification, bounding box regression and RNN based adaptive text region representation. Here, text/non-text classification and bounding box regression are similar as other two-stage text detection methods, while the last branch is proposed for arbitrary shape text representation.

For the proposed branch, the input are the features of each text proposal, which are obtained by using ROI pooling to the CNN feature maps generated with SE-VGG16. The output target of this branch is the adaptive number of boundary points for each text region. Because the output length changes for different text regions, it is reasonable to use RNN to predict these points. Therefore, Long Short-Term Memory (LSTM) [6] is used here, which is a kind of RNN and popular for processing sequence learning problem, such as machine translation, speech recognition, image caption and text recognition.

Though it is proposed that pairwise boundary points are used for text region representation, different ways can be used for pairwise points representation. Easily, we can imagine that using the coordinates of two pairwise points $(x_i, y_i, x_{i+1}, y_{i+1})$ to represent them. In this way, the coordinates of pairwise points are used as the regression targets as shown in Figure 5. However, pairwise points can be represented in a different way, using the coordinate of their center point (x_i^c, y_i^c), the distance from the center point to them h_i, and their orientation θ_i. However, the angle target is not stable in some special situations. For example, angle near $90°$ is very similar to angle near $-90°$ in spatial, but their angles are quite different. This makes the network hard to learn the angle target well. Besides, the orientation can be represented by $\sin\theta_i$ and $\cos\theta_i$, which can be predicted stably. However, more parameters are needed. Therefore, the coordinates of points $(x_i, y_i, x_{i+1}, y_{i+1})$ are used as the regression targets in the proposed method.
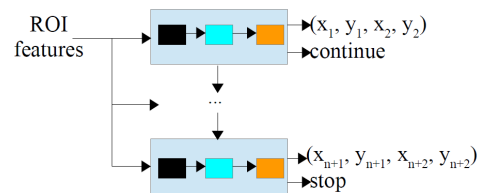

Figure 5. LSTM used for learning text region representation. The input of each time step in the LSTM are the ROI pooling features of the corresponding text proposals.

The inputs of all time steps in the LSTM used here are the same, which are the ROI pooling features of the corresponding text proposals. And the output of each time step are the coordinates of the pairwise points on text region boundary. Meanwhile, as adaptive numbers of points are used for different text regions, a stop label is needed

to represent when the predicting network stops. Because stop label prediction is a classification problem while coordinates prediction is a regression problem, it is not appropriate to put them in the same branch. Therefore, there are two branches in each time step of the LSTM: one for point coordinate regression and one for stop label prediction. At each time step the coordinates of two pairwise boundary points of text region and the label stop/continue are predicted. If the label is continue, the coordinates of another two points and a new label are predicted in the next time step. Otherwise, the prediction stops and text region is represented with the points predicted before. In this way, text regions in the input image can be detected and represented with different polygons made up by the predicted pairwise points.

While Non-Maximum Suppression (NMS) is extensively used to post-process detection candidates by general object detection methods, it is also needed in the proposed method. As the detected text regions are represented with polygons, normal NMS which is computed based on the area of horizontal bounding box is not suitable here. Instead, a polygon NMS is used, which is computed based on the area of the polygon of text region. After NMS, the remaining text regions are output as the detection result.

3.4. Training objective

As Text-RPN in the proposed method is similar as the RPN in Faster R-CNN [23], the training loss of Text-RPN is also computed in the similar way as it. Therefore, in this section, we only focus on the loss function of refinement network in proposal refinement. The loss defined on each proposal is the sum of a text/non-text classification loss, a bounding box regression loss, a boundary points regression loss and a stop/continue label classification loss. The multi-task loss function on each proposal is defined as:

$$
\begin{aligned}
L_{sum} = & L_{cls}(p, t) + \lambda_1 t \sum_{i \in \{x, y, w, h\}} L_{reg}(v_i, v_i^*) \\
& + \lambda_2 t \sum_{i \in \{x_1, y_1, x_2, y_2, \ldots, x_n, y_n\}} L_{reg}(u_i, u_i^*) \\
& + \lambda_3 t \sum_{i \in \{l_1, l_2, \ldots, x_{n/2}\}} L_{cls}(l_i, l_i^*)
\end{aligned}
\tag{1}
$$

λ_1, λ_2 and λ_3 are balancing parameters that control the trade-off between these terms and they are set as 1 in the proposed method.

For the text/non-text classification loss term, t is the indicator of the class label. Text is labeled as 1 ($t = 1$), and background is labeled as 0 ($t = 0$). The parameter $p = (p_0, p_1)$ is the probability over text and background classes computed after softmax. Then, $L_{cls}(p, t) = -\log p_t$ is the log loss for true class t.

For the bounding box regression loss term, $v = (v_x, v_y, v_w, v_h)$ is a tuple of true bounding box regression

targets including coordinates of the center point and its width and height, and $v^* = (v_x^*, v_y^*, v_w^*, v_h^*)$ is the predicted tuple for each text proposal. We use the parameterization for v and v^* given in Faster R-CNN [23], in which v and v^* specify scale-invariant translation and log-space height/width shift relative to an object proposal.

For the boundary points regression loss term, $u = (u_{x_1}, u_{y_1}, \ldots, u_{x_n}, u_{y_n})$ is a tuple of true coordinates of boundary points, and $u^* = (u_{x_1}^*, u_{y_1}^*, \ldots, u_{x_n}^*, u_{y_n}^*)$ is the predicted tuple for the text label. To make the points learned suitable for text of different scales, the learning targets should also be processed to make them scale invariant. The parameters $(u_{x_i}^*, u_{y_i}^*)$ are processed as following:

$$
u_{x_i}^* = (x_i^* - x_a)/w_a, u_{y_i}^* = (y_i^* - y_a)/h_a, \tag{2}
$$

where x_i^* and y_i^* denote the coordinates of the boundary points, x_a and y_a denote the coordinates of the center point of the corresponding text proposal, w_a and h_a denote the width and height of this proposal.

Let (w, w^*) indicates (v_i, v_i^*) or (u_i, u_i^*), $L_{reg}(w, w^*)$ is defined as the smooth L1 loss as in Faster R-CNN [23]:

$$
L_{reg}(w, w^*) = smooth_{L1}(w - w^*), \tag{3}
$$

$$
smooth_{L1}(x) = \begin{cases} 0.5x^2 & if |x| < 1 \\ |x| - 0.5 & otherwise \end{cases} \tag{4}
$$

For the stop/continue label classification loss term, it is also a binary classification and its loss is formatted similar as text/non-text classification loss.

4. Experiments

4.1. Benchmarks

Five benchmarks are used in this paper for performance evaluation, which are introduced in the following:

- **CTW1500**: The CTW1500 dataset [17] contains 500 test images and 1000 training images, which contain multi-oriented text, curved text and irregular shape text. Text regions in this dataset are labeled with 14 scene text boundary points at sentence level.

- **TotalText**: The TotalText dataset [2] consists of 300 test images and 1255 training images with more than 3 different text orientations: horizontal, multi-oriented, and curved. The texts in these images are labeled at word level with adaptive number of corner points.

- **ICDAR2013**: The ICDAR2013 dataset [10] contains focused scene texts for ICDAR Robust Reading Competition. It includes 233 test images and 229 training images. The scene texts are horizontal and labeled with horizontal bounding boxes made up by 2 points at word level.

- **ICDAR2015**: The ICDAR2015 dataset [9] focuses on incidental scene text in ICDAR Robust Reading Competition. It includes 500 testing images and 1000 training images. The scene texts have different orientations, which are labeled with inclined boxes made up by 4 points at word level.

- **MSRA-TD500**: The MSRA-TD500 dataset [29] contains 200 test images and 300 training images, that contain arbitrarily-oriented texts in both Chinese and English. The texts are labeled with inclined boxes made up by 4 points at sentence level. Some long straight text lines exist in the dataset.

The evaluation for text detection follows the ICDAR evaluation protocol in terms of Recall, Precision and Hmean. Recall represents the ratio of the number of correctly detected text regions to the total number of text regions in the dataset while Precision represents the ratio of the number of correctly detected text regions to the total number of detected text regions. Hmean is single measure of quality by combining recall and precision. A detected text region is considered as correct if its overlap with the ground truth text region is larger than a given threshold. The computation of the three evaluation terms is usually different for different datasets. While the results on ICDAR 2013 and ICDAR 2015 can be evaluated through ICDAR robust reading competition platform, the results of the other three datasets can be evaluated with the given evaluation methods corresponding to them.

4.2. Implementation details

Our scene text detection network is initialized with pre-trained VGG16 model for ImageNet classification. When the proposed method is tested on the five datasets, different models are used for them, which are trained using only the training images of each dataset with data augmentation. All models are trained 10×10^4 iterations in total. Learning rates start from 10^{-3}, and are multiplied by 1/10 after 2×10^4, 6×10^4 and 8×10^4 iterations. We use 0.0005 weight decay and 0.9 momentum. We use multi-scale training, setting the short side of training images as {400, 600, 720, 1000, 1200}, while maintaining the long side at 2000.

Because adaptive text region representation is used in the proposed method, it can be simply used for these datasets with text regions labeled with different numbers of points. As ICDAR 2013, ICDAR 2015 and MSRA-TD500 are labeled with quadrilateral boxes, they are easy to be transformed into pairwise points. However, for CTW1500 dataset and TotalText dataset, some operations are needed to transform the ground truthes into the form we needed.

Text regions in CTW1500 are labeled with 14 points, which are needed to be transformed into adaptive number of pairwise points. First, the 14 points are grouped into 7

point pairs. Then, we compute the intersection angle for each point, which is the angle of the two vectors from current point to its nearby two points. And for each point pair, the angle is the smaller one of the two points. Next, point pairs are sorted according to their angles in descending order and we try to remove each point pair in the order. If the ratio of the polygon areas after removing operation to the original area is larger than 0.93, this point pair can be removed. Otherwise, the operation stops and the remaining points are used in the training for text region representation.

Moreover, text regions in TotalText are labeled with adaptive number of points, but these points are not pairwise. For text regions labeled with even number of points, it is easy to process by group them into pairs. For text regions labeled with odd number of points, the start two points and the end two points should be found first, and then the corresponding points to the remaining points are found based on their distances to the start points on the boundary.

The results of the proposed method are obtained on single scale input image with one trained model. Because test image scale has a deep impact on the detection results, such as FOTS [15] uses different scales for different datasets, we also use different test scales for different datasets for best performance. In our experiments, the scale for ICDAR 2013 is 960×1400, the scale for ICDAR 2015 is 1200×2000 and the scales for other datasets are all 720×1280.

The proposed method is implemented in Caffe and the experiments are finished using a Nvidia P40 GPU.

4.3. Ablation study

In the proposed method the backbone network is SE-VGG16, while VGG16 is usually used by other state-of-the-art methods. To verify the effectiveness of the backbone network, we test the proposed method with different backbone networks (SE-VGG16 vs VGG16) on CTW1500 dataset and ICDAR 2015 dataset as shown in Table 2. The results show that SE-VGG16 is better than VGG16, with which better performances achieved on the two datasets.

Backbone	Recall	Precision	Hmean
CTW1500			
VGG16	79.1	79.7	79.4
SE-VGG16	80.2	80.1	80.1
ICDAR2015			
VGG16	83.3	90.4	86.8
SE-VGG16	86.0	89.2	87.6

Table 2. Ablation study on backbone network.

Meanwhile, an adaptive text region representation is proposed for text of arbitrary shapes in this paper. To validate its effectiveness for scene text detection, we add an ablation study on text region representation on CTW1500

dataset. For comparison, fixed text region representation directly uses the fixed 14 points as the regression targets in the experiment. Table 3 shows the experimental results of different text region representation methods on CTW1500 dataset. The recall of the method with adaptive representation is much higher than fixed representation (80.2% vs 76.4%). It justifies that the adaptive text region representation is more suitable for texts of arbitrary shapes.

Representation	Recall	Precision	Hmean
Fixed	76.4	80.0	78.2
Adaptive	80.2	80.1	80.1

Table 3. Ablation study on text region representation.

4.4. Comparison with State-of-the-arts

To show the performance of the proposed method for different shape texts, we test it on several benchmarks. We first compare its performance with state-of-the-arts on CTW1500 and TotalText which both contains challenging multi-oriented and curved texts. Then we compare the methods on the two most widely used benchmarks: IC-DAR2013 and ICDAR2015. At last we compare them on MSRA-TD500 which contains long straight text lines and multi-language texts (Chinese+English).

Table 4 and Table 5 compare the proposed method with state-of-the-art methods on CTW1500 and TotalText, respectively. The propose method is much better than all other methods on CTW1500 including the methods designed for curved texts such as CTD, CTD+TLOC and TextSnake (H-mean: 80.1% vs 69.5%, 73.4% and 75.6%). Meanwhile, it also achieves better performance (Hmean: 78.5%) than all other methods on TotalText. The performances on the two datasets containing challenging multi-oriented and curved texts mean that the proposed method can detect scene text of arbitrary shapes.

Method	Recall	Precision	Hmean
SegLink [24]	40.0	42.3	40.8
EAST [34]	49.1	78.7	60.4
DMPNet [16]	56.0	69.9	62.2
CTD [17]	65.2	74.3	69.5
CTD+TLOC [17]	69.8	77.4	73.4
TextSnake [19]	**85.3**	67.9	75.6
Proposed	80.2	**80.1**	**80.1**

Table 4. Results on CTW1500.

Table 6 shows the experimental results on ICDAR2013 dataset. The proposed method achieves the best performance same as Mask Textspotter, whose Hmean both are 91.7%. Because the proposed method is tested on single

Method	Recall	Precision	Hmean
SegLink [24]	23.8	30.3	26.7
EAST [34]	36.2	50.0	42.0
DeconvNet [2]	44.0	33.0	36.0
Mask Textspotter [20]	55.0	69.0	61.3
TextSnake [19]	74.5	**82.7**	78.4
Proposed	**76.2**	80.9	**78.5**

Table 5. Results on TotalText.

scale input image with single model, only the results generated in this situation are used here. The results show that the proposed method can also process horizontal text well.

Method	Recall	Precision	Hmean
TextBoxes [12]	83.0	88.0	85.0
SegLink [24]	83.0	87.7	85.3
He et al. [5]	81.0	92.0	86.0
Lyu et al. [21]	84.4	92.0	88.0
FOTS [15]	-	-	88.2
RRPN [22]	87.9	94.9	91.3
FEN [31]	89.1	93.6	91.3
Mask Textspotter [20]	88.6	**95.0**	91.7
Proposed	**89.7**	93.7	**91.7**

Table 6. Results on ICDAR2013.

Table 7 shows the experimental results on ICDAR 2015 dataset and the proposed method achieve the second best performance, which is only a little lower than FOTS (H-mean: 87.6% vs 88.0%). While FOTS is trained end-to-end by combining text detection and recognition, the proposed method is only trained for text detection, which is much easier to train than FOTS. And the results tested on single scale input image with single mode are used here. The results show that the proposed method achieves comparable performance with state-of-the-arts, which means it can also process multi-oriented text well.

Method	Recall	Precision	Hmean
SegLink [24]	76.8	73.1	75.0
RRPN [22]	77.0	84.0	80.0
He et al. [5]	81.0	92.0	86.0
R2CNN [8]	79.7	85.6	82.5
TextSnake [19]	80.4	84.9	82.6
PixelLink [3]	82.0	85.5	83.7
InceptText [28]	80.6	90.5	85.3
Mask Textspotter [20]	81.0	**91.6**	86.0
Proposed	**86.0**	89.2	87.6
FOTS [15]	-	-	**88.0**

Table 7. Results on ICDAR2015.

Table 8 shows the results on MSRA-TD500 dataset and

it shows that our detection method can support long straight text line detection and Chinese+English detection well. It achieves Hmean of 83.6% and is better than all other methods.

Method	Recall	Precision	Hmean
EAST [34]	67.4	87.3	76.1
SegLink [24]	70.0	86.0	77.0
PixelLink [3]	73.2	83.0	77.8
TextSnake [19]	73.9	83.2	78.3
InceptText [28]	79.0	87.5	83.0
MCN [18]	79.0	**88.0**	83.0
Proposed	**82.1**	85.2	**83.6**

Table 8. Results on MSRA-TD500.

4.5. Speed

The speed of the proposed method is compared with two other methods as shown in Table 9, which are all able to deal with arbitrary shape scene text. From the results, we can see that the speed of the proposed method is much faster than the other two methods. While pixel-wise prediction is needed in Mask Textspotter and TextSnake, it is not needed in the proposed method and less computation is needed.

Method	Scale	Speed
TextSnake [19]	768	1.1 fps
Mask Textspotter [20]	720	6.9 fps
Proposed	720	**10.0 fps**

Table 9. Speed compared on different detection methods supporting arbitrary shape texts.

4.6. Qualitative results

Figure 6 illustrates qualitative results on CTW1500, TotalText, ICDAR2013, ICDAR2015 and MSRA-TD500. It shows that the proposed method can deal with various texts of arbitrarily oriented or curved, different languages, non-uniform illuminations and different text lengths at word level or sentence level.

5. Conclusion

In this paper, we propose a robust arbitrary shape scene text detection method with adaptive text region representation. After text proposal using a Text-RPN, each text region is verified and refined using a RNN for predicting adaptive number of boundary points. Experiments on five benchmarks show that the proposed method can not only detect horizontal and oriented scene texts but also work well for arbitrary shape scene texts. Particularly, it outperforms existing methods significantly on CTW1500 and MSRA-TD500,

Figure 6. Results on different datasets. (a) results on CTW1500; (b) results on TotalText; (c) results on ICDAR2013; (d) results on ICDAR2015; (e) results on MSRA-TD500.

which are typical of curved texts and multi-oriented texts, respectively.

In the future, the proposed method can be improved in several aspects. First, arbitrary shape scene text detection may can be improved by using corner point detection. This will require easier annotations for training images. Second, to fulfill the final goal of text recognition end-to-end text recognition for arbitrary shape scene text will be considered.

References

[1] Lluis Castrejon, Kaustav Kundu, Raquel Urtasun, and Sanja Fidler. Annotating object instances with a polygon-rnn. In *IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, pages 5230–5238, 2017.

[2] Chee Kheng Chng and Chee Seng Chan. Total-text: A comprehensive dataset for scene text detection and recognition. In *International Conference on Document Analysis and Recognition(ICDAR)*, pages 935–942, 2017.

[3] Dan Deng, Haifeng Liu, Xuelong Li, and Deng Cai. Pixellink: Detecting scene text via instance segmentation. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[4] Kaiming He, Georgia Gkioxari, Piotr Dollr, and Ross Girshick. Mask r-cnn. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017.

[5] Wenhao He, Xu-Yao Zhang, Fei Yin, and Cheng-Lin Liu. Deep direct regression for multi-oriented scene text detection. In *IEEE International Conference on Computer Vision(ICCV)*, pages 745–753, 2017.

[6] Sepp Hochreiter and Jrgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[7] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, pages 7132–7141, 2018.

[8] Yingying Jiang, Xiangyu Zhu, Xiaobing Wang, Shuli Yang, Wei Li, Hua Wang, Pei Fu, and Zhenbo Luo. R2cnn: Rotational region cnn for arbitrarily-oriented scene text detection. In *International Conference on Pattern Recognition(ICPR)*, pages 3610–3615, 2018.

[9] Dimosthenis Karatzas, Lluis Gomez-Bigorda, Anguelos Nicolaou, Suman Ghosh, Andrew Bagdanov, Masakazu Iwamura, Jiri Matas, Lukas Neumann, Vijay Ramaseshan Chandrasekhar, Shijian Lu, Faisal Shafait, Seiichi Uchida, and Ernest Valveny. Icdar 2015 competition on robust reading. In *International Conference on Document Analysis and Recognition(ICDAR)*, pages 1156–1160, 2015.

[10] Dimosthenis Karatzas, Faisal Shafait, Seiichi Uchida, Masakazu Iwamura, Lluis Gomez i Bigorda, Sergi Robles Mestre, Joan Mas, David Fernandez Mota, Jon Almazan Almazan, and Lluis Pere de las Heras. Icdar 2013 robust reading competition. In *International Conference on Document Analysis and Recognition(ICDAR)*, pages 1484–1493, 2013.

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

[12] Minghui Liao, Baoguang Shi, Xiang Bai, Xinggang Wang, and Wenyu Liu. Textboxes: A fast text detector with a single deep neural network. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 4161–4167, 2017.

[13] Minghui Liao, Zhen Zhu, Baoguang Shi, Gui-Song Xia, and Xiang Bai. Rotation-sensitive regression for oriented scene text detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, pages 5909–5918, 2018.

[14] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. Ssd: Single shot multibox detector. In *European Conference on Computer Vision(ECCV)*, pages 21–37, 2016.

[15] Xuebo Liu, Ding Liang, Shi Yan, Dagui Chen, Yu Qiao, and Junjie Yan. Fots: Fast oriented text spotting with a unified network. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, pages 5676–5685.

[16] Yuliang Liu and Lianwen Jin. Deep matching prior network: Toward tighter multi-oriented text detection. In *IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, pages 3454–3461.

[17] Yuliang Liu, Lianwen Jin, Shuaitao Zhang, Canjie Luo, and Sheng Zhang. Curved scene text detection via transverse and longitudinal sequence connection. *Pattern Recognition*, 90(6):337–345, 2018.

[18] Zichuan Liu, Guosheng Lin, Sheng Yang, Jiashi Feng, Weisi Lin, and Wang Ling Goh. Learning markov clustering networks for scene text detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, 2018.

[19] Shangbang Long, Jiaqiang Ruan, Wenjie Zhang, Xin He, Wenhao Wu, and Cong Yao. Textsnake: A flexible representation for detecting text of arbitrary shapes. In *European Conference on Computer Vision(ECCV)*, pages 19–35, 2018.

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

[21] Pengyuan Lyu, Cong Yao, Wenhao Wu, Shuicheng Yan, and Xiang Bai. Multi-oriented scene text detection via corner localization and region. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, pages 7553–7563, 2018.

[22] Jianqi Ma, Weiyuan Shao, Hao Ye, Li Wang, Hong Wang, Yingbin Zheng, and Xiangyang Xue. Arbitrary-oriented scene text detection via rotation proposals. *IEEE Transactions on Multimedia*, 20(11):3111–3122, 2018.

[23] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, 2017.

[24] Baoguang Shi, Xiang Bai, and Serge Belongie. Detecting oriented text in natural images by linking segments. In *IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, pages 3482–3490, 2017.

[25] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[26] Lei Sun, Qiang Huo, and Wei Jia. A robust approach for text detection from natural scene images. *Pattern Recognition*, 48(9):2906–2920, 2015.

[27] Shangxuan Tian, Yifeng Pan, Chang Huang, Shijian Lu, Kai Yu, and Chew Lim Tan. Text flow: A unified text detection system in natural scene images. In *IEEE International Conference on Computer Vision (ICCV)*, pages 4651–4659, 2015.

[28] Qiangpeng Yang, Mengli Cheng, Wenmeng Zhou, Yan Chen, Minghui Qiu, and Wei Lin. Inceptext: A new inception-text module with deformable psroi pooling for multi-oriented scene text detection. pages 1071–1077, 2018.

[29] Cong Yao, Xiang Bai, Wenyu Liu, Yi Ma, and Zhuowen Tu. Detecting texts of arbitrary orientations in natural images. In *IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, pages 1083–1090, 2012.

[30] Xu-Cheng Yin, Xuwang Yin, Kaizhu Huang, and Hong-Wei Hao. Robust text detection in natural scene images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(5):970–983, 2014.

[31] Sheng Zhang, Yuliang Liu, Lianwen Jin, and Canjie Luo. Feature enhancement network: A refined scene text detector. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 2612–2619, 2017.

[32] Zheng Zhang, Wei Shen, Cong Yao, and Xiang Bai. Symmetry-based text line detection in natural scenes. In *IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, pages 2558–2567, 2015.

[33] Zheng Zhang, Chengquan Zhang, Wei Shen, Cong Yao, Wenyu Liu, and Xiang Bai. Multi-oriented text detection with fully convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, pages 4159–4167, 2016.

[34] Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, and Jiajun Liang. East: An efficient and accurate scene text detector. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2642–2651, 2017.