# TK-Text: Multi-shaped Scene Text Detection via Instance Segmentation

Xiaoge Song[1], Yirui Wu[2], Wenhai Wang[1], and Tong Lu[1](✉)

[1] National Key Lab for Novel Software Technology, Nanjing University, Nanjing, China
sxg514@163.com, lutong@nju.edu.cn

[2] College of Computer and Information, Hohai University, Nanjing, China

**Abstract.** Benefit from the development of deep neural networks, scene text detectors have progressed rapidly over the past few years and achieved outstanding performance on several standard benchmarks. However, most existing methods adopt quadrilateral bounding boxes to represent texts, which are usually inadequate to deal with multi-shaped texts such as the curved ones. To keep consist detection performance on both quadrilateral and curved texts, we present a novel representation, i.e., text kernel, for multi-shaped texts. On the basis of text kernel, we propose a simple yet effective scene text detection method, named as TK-Text. The proposed method consists of three steps, namely text-context-aware network, segmentation map generation and text kernel based post-clustering. During text-context-aware network, we construct a segmentation-based network to extract feature map from natural scene images, which are further enhanced with text context information extracted from an attention scheme TKAB. In segmentation map generation, text kernels and rough boundaries of text instances are segmented based on the enhanced feature map. Finally, rough text instances are gradually refined to generate accurate text instances by performing clustering based on text kernel. Experiments on public benchmarks including SCUT-CTW1500, ICDAR 2015 and ICDAR 2017 MLT demonstrate that the proposed method achieves competitive detection performance comparing with the existing methods.

**Keywords:** Multi-shaped scene text detection · Instance segmentation · Text-context-aware network · Text kernel

## 1 Introduction

Recently, deep-learning based scene text detectors have achieved significant progress on standard benchmarks. Former, most methods are designed on the basis of assumption that text instances have quadrilateral shapes, which fails to handle texts with arbitrary shapes. For instance, EAST and some other methods [9,11,17] predict rectangular bounding boxes for scene texts. Yao and several

Y. M. Ro et al. (Eds.): MMM 2020, LNCS 11962, pp. 201–213, 2020.
https://doi.org/10.1007/978-3-030-37734-2_17

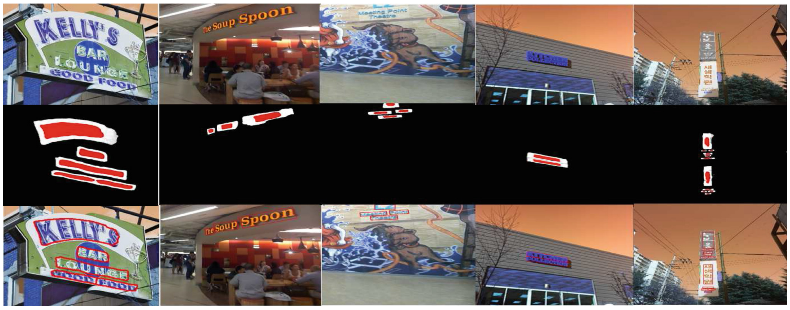

**Fig. 1.** Scene text detection results achieved by the proposed TK-Text. For text of arbitrary orientions and shapes in **first row**, TK-Text first predicts text kernel segmentation (in red color) and rought text segmentation (in white color) as shown in **second row**, and then draws bounding box for text instances as shown in **third row**. (Color figure online)

typical approaches [14,16] group text pixels into text instance of linear shape, which can not be applied to texts of irregular shapes.

With the development of research on curved text detecting problem, more curved text detector [5,12,15,18] have been proposed recently. However, CTD [15] and several curved text detectors extract curved texts by regressing polygonal bounding boxes with 14 vertices, which are inadequate to give smooth text boundaries for texts with extremely irregular shapes. Moreover, performance of most existing methods for curved texts are highly affected by the complex background. Therefore, text related context information on how to accurately locate text boundary still required to be modelled and involved for higher performance.

To address all these issues, we introduce a text detection method named as TK-Text, which not only keeps consistent performance on multi-shaped scene texts with text kernel, but also properly models text-context information on the basis of accurately located text kernel with an attention scheme, named as Text Kernel Attention Block (TKAB). Text kernel are defined as a non-overlap region inside the center of a text, which could help represent quadrilateral and curved text instances as a cluster of text pixels around its shape. To clearly show the effect of text kernel, we show a brief detection process of TK-Text in Fig. 1, where text kernel and rough text segmentation are generated by the Text-Context-Aware network and bounding boxes representing text instances are achieved after post-processing. Moreover, the proposed TAKB models text context information by describing both channel and spatial interdependencies between text and text kernel features, which help enhance distinguish ability of TK-Text.

We propose text kernel based on three considerations. Acting as the central region of text instances, text kernel remains unchanged on border and background variations, which fits for curved text with complex boundaries. Since they are generally far from each other, text kernels can be easily separated to help accurately represent and locate text instances. Last but not least, text kernel provide unique task-specific contextual information, which helps the proposed network to generate more discriminative feature map for higher performance.

The proposed TK-Text has achieved state-of-the-art performance (Precision 0.805, Recall 0.782 and F-measure 0.793) on SCUT-CTW1500, which indicates the consistent and effective detection performance on multi-shaped texts. To summarize, the contributions of this paper are included as below:

- We propose a novel instance segmentation based method TK-Text for multi-shaped scene text detection, which has achieved state-of-the-art performance on SCUT-CTW1500 comparing with existing methods.
- A novel representation for text, i.e., text kernel has carefully designed to process multi-shaped texts, which remain consistent on border and background variations and can be easily distinguished from complex background.
- The proposed TKAB has been constructed with a dual combination form between channel and spatial attention block, which successfully involves text context information into the TK-Text to reduce false positives that occur at the border of text.

## 2 Related Work

Current deep learning methods for text detection can be categorized into two types, i.e., Regression based Methods and Segmentation based Methods.

### 2.1 Regression Based Methods

Mainly inspired by the innovations of end-to-end trainable DNN models on generic object detection, some state-of-the-art scene text detection methods seek to adapt the mechanism of bounding box regression in order to meet the unique attributes of scene text. [9,11,17] successfully adopt the pipelines of object detection into text detection and achieve better performance on public benchmarks than traditional functions. [17] regresses text sides or vertexes on text center, based on shrunk text line segmentation map.

Since in natural scene images there are both quadrangle text lines and curved ones, [15] releases a curve text dataset called SCUT-CTW1500 which contains scene curved texts with polygon labels, and proposes a method called CTD which predicts bounding boxes for curved text by regressing the relative positions for vertices of a 14-sided polygon. Another method SLPR [18] slides a line along horizontal box, then regresses points of intersection of sliding lines and text polygon. TextSnake [5] also concentrates on multi-shaped text lines, it predicts text/non-text and local geometries to reconstruct text instances.

### 2.2 Segmentation Based Methods

Inspired by FCN [4], the key of segmentation based text detectors is to identify all text instances on a saliency map, which indicates whether a pixel is text or

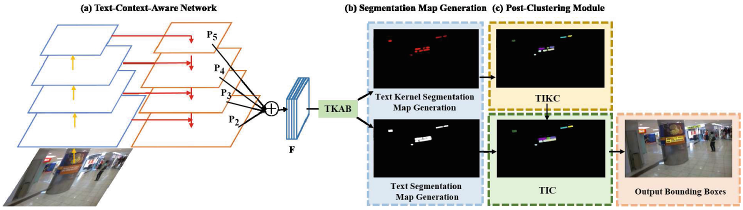
(a) Text-Context-Aware Network
P5
P4
P3
P2
F
TKAB
(b) Segmentation Map Generation
(c) Post-Clustering Module
Text Kernel Segmentation Map Generation
Text Segmentation Map Generation
TIKC
TIC
Output Bounding Boxes

**Fig. 2.** Workflow of the proposed TK-Text network.

background. [14] first predicts segmentation map of text regions and centroid of each character, then generate text instances based on assumption that text lines are linearly arranged characters. To better distinguish text instances, [1] proposes the 8 direction links between pixels and uses them to group and distinguish text instances. [6] creatively adopts corner detection via position-sensitive segmentation map, groups corners to determine the position and shape of a text bounding box.

Different from existing segmentation methods on object, our method present the concept text kernel to capture general characteristics of multi-shaped texts and use it to extract text instances for both quadrilateral and curve texts through shape-independent post-clustering modules. Also we propose a text context aware unit based on text kernel to enhance the network representational power and segmentation performance.

## 3 Proposed Method

The whole pipeline of TK-Text is shown in Fig. 2, which is composed of three steps: (a) text-context-aware network, (b) segmentation map generation and (c) text kernel based post clustering. During step (a), a text-context-aware network is conducted to extract feature map, which consists of FCN network and TKAB unit for feature enhancement. During step (b) Text segmentation and Text Kernel segmentation map are generated from enhanced feature map, which is further applied to produce text kernels and rough boundaries of text instances. In step (c), a text kernel based post-clustering module is proposed, which contains two clustering module, i.e., Text Instance Kernel Clustering (TIKC) and Text Instance Clustering (TIC). TIKC generates results of text instance kernels, while TIC utilizes text kernels as cluster centers to assemble text pixels into clusters, which can be further regarded as individual instances, i.e., text lines.

### 3.1 Architecture of Text-Context-Aware Network

In this subsection, we first describe architecture of the proposed Text-Context-Aware Network, and then give details of the proposed TKAB module.

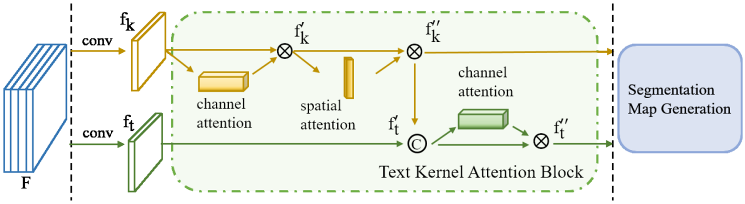

**Fig. 3.** Overview of TKAB: the scheme first refines text kernel features $f_k$ by sequential channel attention and spatial attention. Then it concatenates the refined features and $f_t$, which is further enhanced by another channel attention operation.

**Network Architecture.** The architecture of our network is shown in Fig. 1(a). First we gradually merge features from different layers of backbone network through a standard FPN structure, following the concept of [3]. The backbone can be the widely-used networks proposed for image classification, e.g. ResNet [2]. Then we fuse the intermediate outputs of FPN (i.e. $P_2, P_3, P_4, P_5$) by resize-concatenation operation "$\oplus$" into F, whose size is 1/4 of the input images. In our experiments, resize operation is implemented as bilinear interpolation layer. After merging, we apply two $3 \times 3$ convolution layers to reduce channels of F from 1024 to 256 and produce $f_k, f_t$ in Fig. 3, respectively. Consequently, $f_k, f_t$ are sent to TKAB unit and two $1 \times 1$ convolution-sigmoid layers for text and text kernel segmentation. Finally, we interpolate the segmentation maps to original size of input images by additional upsampling layers, obtaining high resolution segmentation results.

**Text Kernel Attention Block.** Lack of context information may cause misclassification. Inspired by CBAM [13], we introduce a text-context-aware module TKAB to solve this problem, which exploits their cross channel relationship through attention scheme. In this way we involves context information to make text segmentation more robust.

We describe the computation process of TKAB as depicted in Fig. 3. Given the feature maps $f_t, f_k \in R^{c \times H \times W}$ as inputs, Text Kernel Attention Block (TKAB) sequentially computes a 1D channel attention map $M_k^c \in R^{c \times 1 \times 1}$, and a 2D spatial attention map $M_k^s \in R^{1 \times H \times W}$ to refine the text kernel features by exploiting the inter-channel relationship of features. The attention process can be summarized as:

$$\begin{aligned} f_k^{'} &= M_k^c(f_k) \otimes f_k \\ f_k^{''} &= M_k^s(f_k^{'}) \otimes f_k^{'} \end{aligned} \tag{1}$$

where "$\otimes$" refers to element-wise multiplication. Next it concatenates the refined text kernel features $f_k^{''}$ with the text features $f_t$, and applies a $3 \times 3$ convolution layer to merge features. The feature fusion process can be summarized as:

$$f_t^{'} = conv(f_t || f_k^{''}) \tag{2}$$

The "||" refers concatenation operation along the channel dimension. Based on the merged feature map, TKAB further produces a 1D channel attention Map $M_t^c \in R^{2c \times 1 \times 1}$ to refine it, which can be represented as:

$$f_t^{''} = M_t^c(f_t^{'}) \otimes f_t^{'} \tag{3}$$

To generate channel attention maps $M_k^c$ and $M_t^c$, We utilize both average-pooled features and max-pooled features to aggregate spatial information, and use a shared Multi-Layer Perceptron (MLP) network with one hidden activation layer. The hidden activation layer for $M_k^c$ and $M_t^c$ are set to $R^{c/r \times 1 \times 1}$ and $R^{2c/r \times 1 \times 1}$, respectively, where "r" refers to the reduction ratio. To obtain a spatial attention map, TKAB applies average-pooling and max-pooling operations along the channel axis, then feeds the concatenated feature maps of them into a $3 \times 3$ convolution layer. After enhancing and merging features by TKAB, we take $f_t^{''}$, $f_k^{''}$ as the final refined features for subsequent text and text kernel segmentation, respectively.

### 3.2 Segmentation Map Generation

Based on the enhanced features $f_k^{''}$ and $f_t^{''}$ produced by TKAB, the text context aware network further generates text and text kernel segmentation map, respectively. The loss function can be written as a linear weighted sum of losses of two segmentation sub-tasks:

$$L = \lambda L_t + (1 - \lambda) L_k \tag{4}$$

where $L_t$ and $L_k$ refer to the loss for text and text kernel segmentation, respectively, and $\lambda$ is a weight to make a balance between two tasks.

**Text Segmentation Map Generation.** This branch generates score map indicating whether a pixel is text or background. Ground truths as shown in Fig. 4(a) and (b) are generated according to [9]. We label all pixels inside text bounding boxes as positive, otherwise negative. For text segmentation training, we adopt dice coefficient loss introduced in [8]. It is common that text regions are usually small in natural scene images and dice coefficient loss can help reduce the bias to non-text regions. $L_{text}$ can be calculated as:

$$L_{text} = 1 - 2\sum_{x,y}(P_{x,y} * G_{x,y}) / (\sum_{x,y} P_{x,y}^2 + \sum_{x,y} G_{x,y}^2) \tag{5}$$

where $P$ and $G$ refer to the prediction and ground truth respectively. Moreover, in natural scenes there are plenty of patterns that similar to texts. To better distinguish them, we adopt Online Hard Example Mining (OHEM) [1,10] which automatically selects hard samples based on positive samples to train. When there are S positive pixels, $r \times S$ pixels of largest loss are selected as hard samples, and r is a hyper-parameter fixed to 3 in our experiments.

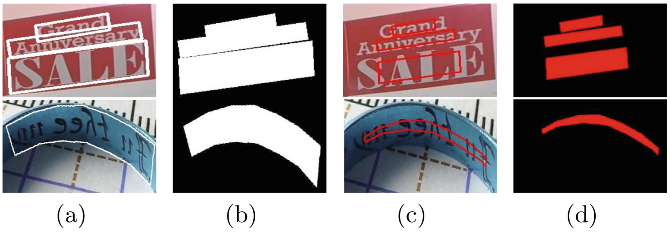

**Fig. 4.** Ground truth and label generation: (a) original image with corresponding bounding boxes for text region, (b) white masks are the corresponding ground truth of text region, (c) original image with bounding boxes for text kernel, (d) red masks are the ground truth of text kernel segmentation. (Color figure online)

**Text Kernel Segmentation Map Generation.** This branch aims to segment text kernels. Different from the "shrunk text lines" mentioned in [17] which is applied on quadrangular texts, our text kernels are defined as consistent representatives of multi-shaped scene text instances, which can well locate the central regions of both quadrangle and curved texts therefore guide the following clustering processes. The generation of text kernel ground truths and loss function are given in this section.

To obtain the ground truths in Fig. 4(c) and (d), we first shrink the original text annotation boxes to its center by $\triangle_{off}$ pixels with Vatti clipping algorithm. $\triangle_{off}$ refers to the margin between original bounding box and text kernel. We calculate it based on a hyper-parameter $\hat{r}$, which is the scale ratio between area of text kernel and area of whole text. The formulation of $\triangle_{off}$ is:

$$\triangle_{off} = (1 - \hat{r}^2) \times S/L \tag{6}$$

where $S$ and $L$ represent the area of bounding box and its perimeter, respectively. Next, we label pixels inside the shrunk bounding boxes as positive, otherwise negative. We use dice coefficient loss to train text kernel segmentation and $L_{kernel}$ can be calculated as:

$$L_{kernel} = 1 - 2\sum_{x,y}(P_{x,y} * G_{x,y})/(\sum_{x,y} P_{x,y}^2 + \sum_{x,y} G_{x,y}^2) \tag{7}$$

where $P$ represents the prediction map and $G$ is the ground truth.

### 3.3 Text Kernel Based Post-clustering

In order to detect and distinguish text lines, we propose a fast post-clustering processing method which is composed of two clustering module TIKC and TIC. Since text segmentation map cannot provide enough information to separate adjacent text instances on their own, while text kernel segmentation map is lack of sufficient information to point out the whole text instances, the combination of TIKC and TIC can bridge the advantages of both sides. TK-Text takes text kernels as instance representatives and use them to group text pixels and conduct the generation of whole text instances.

**Text Instance Kernel Clustering.** To obtain text instance kernels, we feed text kernel segmentation map to a clustering module called TIKC. It uses the connectivity information to gather positive pixels into non-overlap regions. Since text kernels do not overlap, they can be succinctly separated from each other.

In TIKC, we first set a threshold to transform the output probability map into a 0/1 binary map, where 1 refers to a positive pixel and 0 otherwise. Next, positive pixels are assembled to connected components. Each of them represents a text instance kernel, and pixels within it belong to same instance kernel.

**Text Instance Clustering.** Assuming that we obtain $N$ text pixels by text segmentation, and $K$ text instance kernels $\mu_1, \mu_2, \mu_3, ..., \mu_K$ predicted by TIKC, then the instance segmentation task can be transformed to divide $N$ text pixels into $K$ instance clusters. Regarding the $i_{th}$ text pixel as $x_i$, TK-Text assigns text pixels into text instance by minimizing the cost function $J$ as below:

$$\begin{gathered} J = \sum_{i=1}^{N}\sum_{t=1}^{K} T_{it} \cdot dist(x_i, \mu_t),\ where \\ dist(x_i, \mu_t) = \min_{y \in \mu_t} ||x_i - y||^2\ and\ T_{it} = \begin{cases} 1, & \text{if } x_i \text{ is assigned to } \mu_t \\ 0, & \text{otherwise} \end{cases} \end{gathered} \tag{8}$$

where $x$ is text pixel, and $\mu$ is a text instance kernel, $y$ is the text kernel pixel with smallest $L2$ distance between $x$ among all other kernel pixels.

After Text Instance Clustering step, we achieve results of connected components, which can be regarded as text instances. Based on requirements of different datasets, we utilize minimal area rectangles to generate quadrilateral bounding boxes, and Ramer-Douglas-Peucker algorithm for polygonal ones. In fact, we adopt nearly the same method to generate bounding boxes as PixelLink [1]. To remove false detections, we also implement several extra post-filtering methods proposed in [1]. A predicted box is selected only if it has proper side length, area, and an average confidence $\geq \epsilon$. $\epsilon$ is the selected threshold on confidence. We keep the selected boxes as the final results.

## 4 Experiment

In this section, we firstly introduce dataset. Then, we conduct experiments on quantity of public datasets are performed to prove the efficiency of TK-Text. Finally, we describe implementation details for readers' convenience.

### 4.1 Dataset

To show the effective and consistent performance of TK-Text on both curved and quadrilateral texts, we conduct experiments on three typical public benchmarks namely SCUT-CTW1500, ICDAR2015 and ICDAR 2017 MLT, where SCUT-CTW1500 is designed for arbitrarily curve text detection contains 1000 training

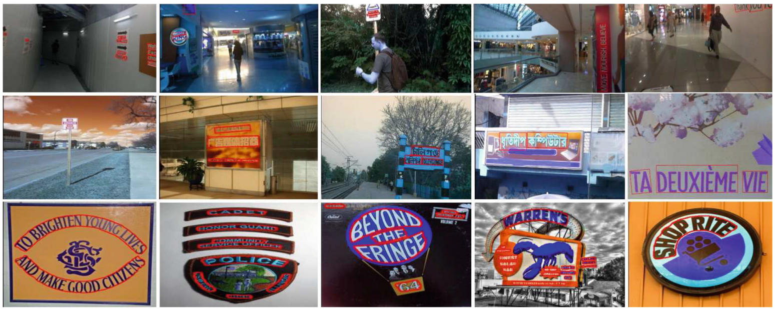

**Fig. 5.** Results on public benchmarks produced by TK-Text. The proposed method draws the bounding boxes with red lines. The detection examples of ICDAR 2015, ICDAR 2017 MLT and SCUT-CTW1500 are listed in the first, second and third row, respectively. (Color figure online)

**Table 1.** The single-scale results on SCUT-CTW1500.

| Method | Precision (%) | Recall (%) | F-measure (%) |
|---|---|---|---|
| CTD [15] | 74.3 | 65.2 | 69.5 |
| CTD + TLOC [15] | 77.4 | 69.8 | 73.4 |
| SLPR [18] | 80.1 | 70.1 | 74.8 |
| TextSnake [5] | 67.9 | **85.3** | 75.6 |
| **TK-Text** | **80.5** | 78.2 | **79.3** |

images and 500 test images, ICDAR 2015 is a commonly used dataset for incidental text detection and ICDAR 2017 MLT is a large scale multi-lingual text dataset proposed on ICDAR2017 Competition.

### 4.2 Results and Analysis

Several results on testing samples obtained by TK-Text are shown in Fig. 5, where we can notice TK-Text not only properly detects multi-shaped text instances, but also own the ability to clearly distinguish adjacent texts from natural scenes. All these sampling results prove that TK-Text can accurately detect both curved and quadrilateral texts from benefits of text kernel and TKAB.

We perform experiments on SCUT-CTW1500, ICDAR2015 and ICDAR2017 MLT datasets, where the former one and the latter two datasets are used to prove the effectiveness on detecting curve and oriented texts, respectively. Moreover, we conduct ablation experiment on ICDAR2015 to prove the usage on accurately detecting of different modules.

**Detecting Curve Text.** For SCUT-CTW1500 dataset, we adopt the same evaluation method proposed by [15] to perform single-scale comparison. We report the statics of experiments on SCUT-CTW1500 in Table 1, where F-measure achieved by TK-Text, i.e., 79.3, demonstrates the solid superiority of

**Table 2.** Comparison results on ICDAR 2015 Dataset.

| Method | Precision (%) | Recall (%) | F-measure (%) |
|---|---|---|---|
| MCLAB FCN [16] | 70.8 | 43.0 | 53.6 |
| CTPN [11] | 74.2 | 51.5 | 60.9 |
| Yao et al. [14] | 72.3 | 58.7 | 64.8 |
| SegLink [9] | 73.1 | 76,8 | 75.0 |
| EAST+PVANET2s RBOX [17] | 83.6 | 73.5 | 78.2 |
| PixelLink [1] | 85.5 | 82.0 | 83.7 |
| Lyu et al. [6] | **94.1** | 70.7 | 80.7 |
| **TK-Text** | 86.1 | **83.0** | **84.5** |

**Table 3.** Comparison results on ICDAR 2017 MLT Dataset.

| Method | Precision (%) | Recall (%) | F-measure (%) |
|---|---|---|---|
| YY AI OCR Group [7] | 64.8 | 44.3 | 52.6 |
| SARI-FDU-RRPN-v0 [7] | 67.1 | 55.4 | 60.7 |
| SARI-FDU-RRPN-v1 [7] | 71.2 | 55.5 | 62.4 |
| **TK-Text** | **76.8** | **55.9** | **64.7** |

the proposed method for detecting curved texts. In fact, text kernel based post-clustering module can greatly refine the rough text instances with accurate text boundaries. With the hyperparameters obtained by grid search on the training set, our method achieved a more balanced trade-off between precision and recall, when comparing with TextSnake [5] which clearly trades recall with precision. Results on SCUT-CTW1500 proves the effectiveness of TK-Text in detecting scene texts of irregular shapes.

**Detecting Oriented Text.** Comparisons with other methods on ICDAR datasets are shown in Tables 2 and 3, where we can notice that TK-Text achieves comparable F-measure results of 84.5% and 64.7% on ICDAR 2015 and ICDAR 2017 MLT, respectively. It's noted that Lyu [6] achieves 94.1% in precision, which is much higher than 86.1% achieved by TK-Text. The reason of higher precision lies in the fact that Lyu [6] pre-trains their model with a quite large dataset, i.e., 800000 synthetic images and carefully finetunes on ICDAR 2015 to pursue best precision result. Meanwhile, TK-Text achieves balance performance between precision and recall with less training data. The competitive results on the ICDAR datasets indicate the proposed method can obtain consistent performance on quadrilateral texts.

**Ablation Experiment.** Results of several contrast control experiments on ICDAR 2015 are reported in Table 4, which is designed to show the specific effect of text kernel, TKAB and post processing. Specifically, we directly locate text instances on text segmentation map without text kernel, TKAB unit and

**Table 4.** Ablation experiment with different settings on ICDAR 2015 Dataset.

| Configuration | Precision (%) | Recall (%) | F-measure (%) |
|---|---|---|---|
| Without text kernel | 68.0 | 63.0 | 65.4 |
| Without post processing | 74.1 | **86.4** | 79.8 |
| Without TKAB | 85.9 | 81.7 | 83.7 |
| TK-Text | **86.1** | 83.0 | **84.5** |

post processing module to perform three group of ablation experiment, respectively. We can observe a significant decline in both precision and recall achieved by "Without text kernel", which proves text kernel could help improve both precision and recall. Meanwhile, we can find that disable post processing module could result in a sharp decline in precision, which proves that post filtering can effectively reduce false detections. Last but not least, TKAB makes the predicted bounding boxes more accurate, thus improving performance with 0.8 on f-measure.

### 4.3 Implementation Details

We adopt a data augmentation method to help build the proposed TK-Text, where we rotate images by a random degree in the range of [−10, 10], resize them with a random ratio in the range of [0.5, 1.0, 2.0, 3.0] and uniformly sample a 640 × 640 patch from each image to generate more training samples.

Our method uses ResNet-101 [2] pretrained on ImageNet dataset as backbone, and all networks are trained by SGD. On ICDAR2017 MLT, we train our models for 300 epochs with initial learning rate $1e^{-3}$, which is divided by 10 at 100 and 200 epoch. On ICDAR2015 and SCUT-CTW1500, our model is first pretrained on ICDAR2017 MLT then fine-tuned for 400 epochs, the batch size and learning rate settings are same with [12]. In experiments, reduction ratio of TKAB, $\hat{r}$ of text kernel, and the confidence threshold $\epsilon$ is set to 16, 0.4 and 0.8, respectively. All hyper-parameters are tuned by grid search on training set.

## 5 Conclusion and Future Work

In this paper, we propose a concise method implementing easy instance segmentation to detect multi-shaped text instances in natural scenes. The design of text kernel and TKAB makes TK-Text robust to shapes, which better distinguish text instances. In the future we intend to extend the proposed scene text detection framework to an efficient end-to-end text recognition system.

**Acknowledgment.** This work is supported by the Natural Science Foundation of China under Grant 61672273 and Grant 61832008, and Scientific Foundation of State Grid Corporation of China (Research on Ice-wind Disaster Feature Recognition and Prediction by Few-shot Machine Learning in Transmission Lines), and National Key

R&D Program of China under Grant 2018YFC0407901, the Natural Science Foundation of China under Grant 61702160, the Science Foundation of Jiangsu under Grant BK20170892.

## References

1. Deng, D., Liu, H., Li, X., Cai, D.: PixelLink: detecting scene text via instance segmentation. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)
2. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
3. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2117–2125 (2017)
4. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3431–3440 (2015)
5. Long, S., Ruan, J., Zhang, W., He, X., Wu, W., Yao, C.: TextSnake: a flexible representation for detecting text of arbitrary shapes. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 20–36 (2018)
6. Lyu, P., Yao, C., Wu, W., Yan, S., Bai, X.: Multi-oriented scene text detection via corner localization and region segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7553–7563 (2018)
7. Nayef, N., et al.: ICDAR 2017 robust reading challenge on multi-lingual scene text detection and script identification - RRC-MLT. https://rrc.cvc.uab.es/?ch=8&com=evaluation&task=1
8. Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28
9. Shi, B., Bai, X., Belongie, S.: Detecting oriented text in natural images by linking segments. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2550–2558 (2017)
10. Shrivastava, A., Gupta, A., Girshick, R.: Training region-based object detectors with online hard example mining. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 761–769 (2016)
11. Tian, Z., Huang, W., He, T., He, P., Qiao, Y.: Detecting text in natural image with connectionist text proposal network. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9912, pp. 56–72. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46484-8_4
12. Wang, W., et al.: Shape robust text detection with progressive scale expansion network. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2019
13. Woo, S., Park, J., Lee, J.-Y., Kweon, I.S.: CBAM: convolutional block attention module. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11211, pp. 3–19. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01234-2_1
14. Yao, C., Bai, X., Sang, N., Zhou, X., Zhou, S., Cao, Z.: Scene text detection via holistic, multi-channel prediction. arXiv preprint arXiv:1606.09002 (2016)

15. Yuliang, L., Lianwen, J., Shuaitao, Z., Sheng, Z.: Detecting curve text in the wild: new dataset and new solution. arXiv preprint arXiv:1712.02170 (2017)
16. Zhang, Z., Zhang, C., Shen, W., Yao, C., Liu, W., Bai, X.: Multi-oriented text detection with fully convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4159–4167 (2016)
17. Zhou, X., et al.: East: an efficient and accurate scene text detector. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pp. 5551–5560 (2017)
18. Zhu, Y., Du, J.: Sliding line point regression for shape robust scene text detection. In: 2018 24th International Conference on Pattern Recognition (ICPR), pp. 3735–3740. IEEE (2018)