2018 24th International Conference on Pattern Recognition (ICPR)
Beijing, China, August 20-24, 2018

Sliding Line Point Regression for Shape Robust Scene Text Detection

Yixing Zhu and Jun Du
National Engineering Laboratory for Speech and Language Information Processing
University of Science and Technology of China
Hefei, Anhui, China
zyxsa@mail.ustc.edu.cn, jundu@ustc.edu.cn

***Abstract*—Traditional text detection methods mostly focus on quadrangle text. In this study we propose a novel method named sliding line point regression (SLPR) in order to detect arbitrary-shape text in natural scene. SLPR regresses multiple points on the edge of text line and then utilizes these points to sketch the outlines of the text. The proposed SLPR can be adapted to many object detection architectures such as Faster R-CNN and R-FCN. Specifically, we first generate the smallest rectangular box including the text with region proposal network (RPN), then isometrically regress the points on the edge of text by using the vertically and horizontally sliding lines. To make full use of information and reduce redundancy, we calculate x-coordinate or y-coordinate of target point by the rectangular box position, and just regress the remaining y-coordinate or x-coordinate. Accordingly we can not only reduce the parameters of system, but also restrain the points which will generate more regular polygon. Our approach achieved competitive results on traditional ICDAR2015 Incidental Scene Text benchmark and curve text detection dataset CTW1500.**

I. Introduction

Text detection is important in our daily life as it can be applied in many areas, such as digitization of text, text translation, etc. In this study, we focus on scene text detection. Some of the previous methods [1] [2] [3] have obtained good results on many horizontal scene texts dataset based on Faster R-CNN [4] or SSD [5]. Some methods [6] [7] [8] [9] [10] [11] [12] also tried to solve arbitrary-oriented text detection problem. [10] and [12] regressed first a horizontal rectangle and then a quadrilateral. [13] aimed to generate an irregular polygon after regressing a rectangle. The methods mentioned above mostly treated a text line as a quadrilateral which can be completely represented by four points. However, besides the quadrilateral shape, there are many other various shapes of text line in natural scene. Therefore, recent research [14] [15] have begun to explore curve text line detection, and some papers also explored curve text recognition such as [16]. In this paper we explore both arbitrary-oriented and curve text detection. Our method named sliding line point regression (SLPR) is based on 2-step object detection methods using Faster R-CNN or R-FCN. Firstly we propose some interesting rectangular regions with region proposal network (RPN), then regress the points on the edge of text. We generate some rules to determine which points should be regressed so that there will be relevance between points. Different from [14] which directly regressed both x-coordinate and y-coordinate of fixed annotated points and employed RNN [17], [18] to learn their relevance, we introduce some rules to vertically and horizontally slide lines along text and then regress the intersection points of sliding lines and text lines as illustrated in Fig. 1. In this way, we can only regress x-coordinate or y-coordinate of these points, then calculate other coordinates with the position of rectangle, yielding reduction of unnecessary computation and improvement of performance.

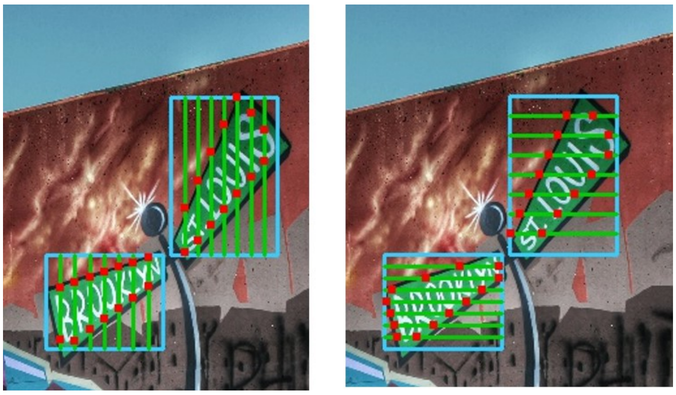

Fig. 1: Illustration of ground truth points generated by horizontally and vertically sliding lines.

The contributions of this paper are as follows:

1. We explore regressing multiple points on the border of text line, try to handle arbitrary-oriented and curve text detection based on Faster R-CNN and R-FCN.

2. We introduce a sliding line method to determine the ground truth points for the regression, and we make full use of the relevance of these points to generate more regular polygon.

II. Related Work

In recent years, scene text detection and recognition has drawn more and more attention. But scene text detection remains a difficult problem due to its complicated orientation and background. All the methods can be divided into three categories: character based methods, word based methods and segmentation based methods. Character based methods often need synthetic datasets because labeling characters in text

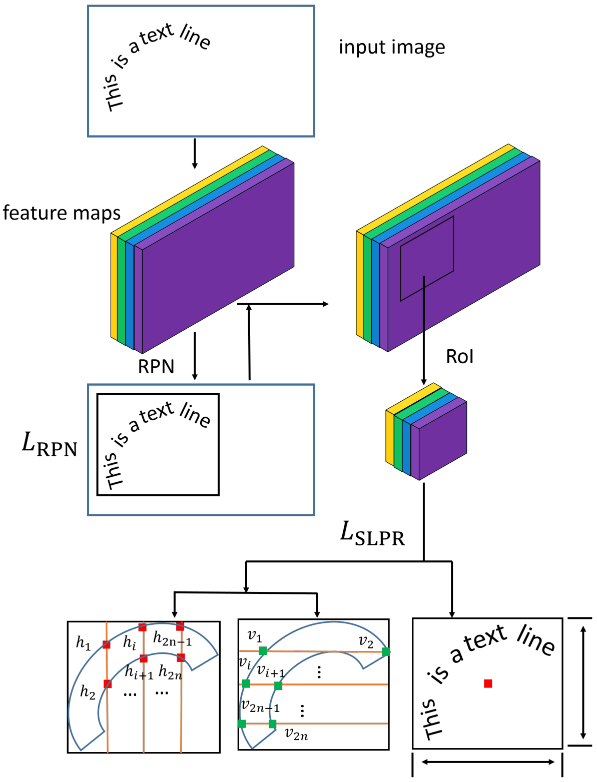

Fig. 2: The SLPR architecture.

lines requires additional efforts. However, the generated data is greatly deviated from the real data, which can not make the trained model to achieve the state-of-the-art results on real dataset such as the popular ICDAR2015 Incidental Scene Text benchmark. In order to solve this problem, [19] used semi-supervised method to finetune model on real data and obtained good results.

The segmentation based methods have also been used in text detection recently. [20] trained a fully convolutional network (FCN) [21], [22] to predict the salient map of text regions, then traced the text line by combining the salient map and character components. [23] added the border class to separate text from their neighbor. [11] and [9] generated text maps and regress the size and angle of the corresponding quadrilateral, or coordinates of four vertexes at the same time. Compared with the traditional segmentation methods, they made a huge breakthrough on ICDAR2015 Incidental Scene Text benchmark.

Many methods of object detection can be applied to text detection, e.g., Faster R-CNN [24], SSD [5], R-FCN [25] and YOLO [26]. [2] used irregular 1×5 convolutional filters instead of the standard 3×3 convolutional filters to make the network more suitable for long text detection. [27] used the attention map to remove background noise. Recently more and more researchers proposed 2-step methods based on Faster R-CNN or R-FCN. [12] firstly generated axis-aligned bounding boxes and then regressed the text quadrangle. They used multi-scale pool operations on the roipool layer. [10] tried to segment and detect text simultaneously. Considering the particularity of text line, [28] appended different angle anchors which are suitable for arbitrary-oriented text line. More recently, [15] considered the polygon case and labeled a new dataset of curve text. [14] also constructed a curve text dataset named CTW1500, and they proposed a new structure named curve text detector (CTD) to solve curve text detection problem.

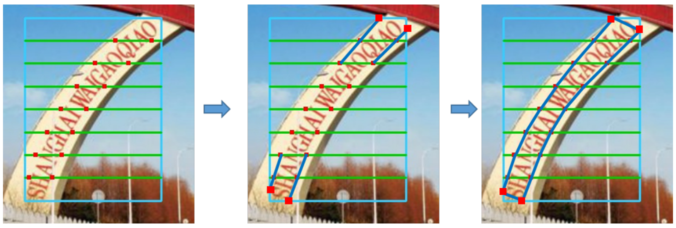

Fig. 3: Restoration of the polygon by using intersection points on sliding lines along the long side.

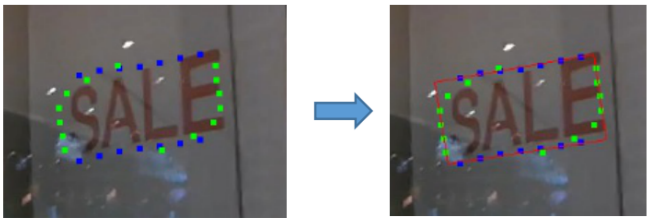

Fig. 4: Restoration of the polygon by using all intersection points from SLPR.

III. Method

Our model can be applied to any 2-step object detection framework such as Faster R-CNN and R-FCN. Our system simultaneously regresses the minimum rectangle including text line and the coordinates of some specific points on the boundary of text line. More specifically, take Faster R-CNN as an example, we first get some interesting regions using the RPN, then we not only regress the position of the rectangle, but also regress the coordinates of the points on the edge of the text line, finally we can get arbitrary shape text area.

A. *Which points should be regressed?*

Obviously, how to determine the point set for restoring the polygon is quite important. We believe the simpler the rules, the easier the neural net learns. We do not regress the fixed points such as vertexes on the polygon because there are a large variety of shapes and angles in natural scene and it is difficult to define the order of fixed feature points for all shapes. Although for quadrilateral, we can perfectly restore it by regressing the corresponding four vertices, the determination of the order of four vertices requires a complicated rule which is difficult for the neural net to learn. Alternatively, as shown in Fig. 2, we introduce some rules to vertically and horizontally slide the lines (we use equidistant sliding in our experiment) on text line and then regress the intersection of sliding lines and text line border. On the other hand, the correlation exists among the coordinates of different intersection points due to the constraints of the sliding lines. It is not necessary to regress both x-coordinate and y-coordinate of all points simultaneously. If it is horizontal sliding, the x-coordinate of the point on the text boundary can be calculated by the coordinates of the rectangle, so we only need to regress the y-coordinate of these points. Similarly, if it is vertical sliding, we only need to regress the x-coordinate of these points. This method not only reduces the computational complexity of the network, but also adds restraints to the regressed points as the prior knowledge which can prevent generating polygons with weird shapes and further improve the accuracy. As for the number of sliding lines, we observe that this parameter is not sensitive to quadrangle text line. But in order to restore other shape text line well, after balancing the performance and network complexity, seven sliding lines are used for we decided to for vertical and horizontal directions, respectively. Accordingly a total of 14 lines with 28 intersection points are generated.

B. The multi-task learning

To optimize the neural network parameters, as illustrated in Fig. 2, we adopt the multi-task learning to define the loss function L as:

$$L = L_{\text{RPN}} + L_{\text{SLPR}} \tag{1}$$

$$L_{\text{RPN}} = L_{\text{RCLS}} + \lambda_{\text{R}} L_{\text{RB}} \tag{2}$$

$$L_{\text{SLPR}} = L_{\text{CLS}} + \lambda_{\text{B}} L_{\text{B}} + \lambda_{\text{S}} L_{\text{SLPRB}} \tag{3}$$

where L_{RPN} is the region proposal loss, L_{RCLS} is region proposal classification loss, L_{RB} is box regression loss. L_{SLPR} is the loss for the second step after RPN. Similarly, the first two items L_{CLS} and L_{B} are respectively classification loss and box regression loss. λ_{R}, λ_{B} and λ_{S} are the related weighting factors, which are all set to 1 in this study. L_{SLPRB} is the proposed new loss item for SLPR:

$$L_{\text{SLPRB}} = \frac{1}{4n}\left[\sum_{j=1}^{2n} L_{\text{Reg}}(x_{v_j}, x^*_{v_j}) + \sum_{i=1}^{2n} L_{\text{Reg}}(y_{h_i}, y^*_{h_i})\right] \tag{4}$$

L_{Reg} is the smooth L1 loss for the box regression task:

$$L_{\text{Reg}}(z, z^*) = \begin{cases} 0.5(z - z^*)^2 & \text{if } |z - z^*| < 1 \\ |z - z^*| - 0.5 & \text{otherwise} \end{cases} \tag{5}$$

In Eq. (4), n represents the number of sliding lines in one direction and we set $n = 7$ in our experiments. In general, each line has two intersection points with the text line border. If there are more than two intersection points, we take the smallest and the largest coordinates. x_{v_j} is x-coordinate of the intersection point v_j of vertically sliding lines and text line border while y_{h_i} is y-coordinate of the intersection point h_i of horizontally sliding lines and text line border. $x^*_{v_j}$ and $y^*_{h_i}$ are the corresponding estimated points from neural net outputs. For horizontally sliding lines, we only regress the y-coordinate of its intersection point. For vertically sliding lines, we only regress the x-coordinate of its intersection point. The other coordinates can be restored through the coordinates of the rectangle:

$$y_{v_j} = y_{\min} + (y_{\max} - y_{\min})\frac{\lfloor (j-1)/2 \rfloor + 1}{(n+1)} \tag{6}$$

$$x_{h_i} = x_{\min} + (x_{\max} - x_{\min})\frac{\lfloor (i-1)/2 \rfloor + 1}{(n+1)} \tag{7}$$

$x_{\min}$ and $y_{\min}$ represent the minimum x-coordinate and y-coordinate of the rectangular border while $x_{\max}$ and $y_{\max}$ represent the maximum x-coordinate and y-coordinate of the rectangular border. $\lfloor \cdot \rfloor$ is the floor function. In a word, in order to regress the coordinates of polygon, 32 parameters should be considered including 4 parameters for the rectangle and 28 parameters to represent x and y coordinates of intersection points on text line border.

C. Restoration of polygon

Through the above SLPR method, we can obtain multiple points from the output of neural nets. To restore the final quadrilateral or polygon, the following two approaches are adopted and compared:

1) Only Using Points in Long Side (PLS): The text line always extends to the long side, and the lines that slide along the long side can better reflect the shape of the text. In fact we can restore the polygon by only scanning the long side, as shown in Fig. 3. Specifically, we firstly judge whether the text line is horizontal or vertical through the regressed rectangle, and then restore the polygon through points in the corresponding direction. Taking the vertical direction as an example in Fig. 3, since we do not regress the intersection point on the rectangular border, we firstly extend the four lines near the border to find four intersection points with the rectangle, then we connect four new points and other intersection points to generate polygon.

2) Using Both of Horizontal and Vertical Points (BHVP): In fact, if we use both horizontal and vertical points to restore polygon, we can calculate a polygon or quadrangle that passes through these points roughly as shown in Fig.4 by using the method in [29]. In this way we can obtain dense enough points in both horizontal and vertical direction and we do not need to calculate the intersection with the rectangle as in PLS method. However, we observe BHVP is not as effective as PLS for the polygon case. So we use this method only on the quadrilateral dataset (ICDAR2015 Incidental Scene Text).

D. Polygonal non-maximum suppression

Non-maximum suppression (NMS) is a basic method commonly used in the object detection, and its purpose is to remove duplicate boxes. The traditional NMS method is based on rectangular boxes, which is not the best choice for other

TABLE I: The performance comparison of SLPR method with different settings on ICDAR2015 Incidental Scene Text dataset.

Scales	NMS	Restoration	Precision (%)	Recall (%)	Hmean (%)
(850)	PNMS	BHVP	86.1	81.6	83.8
(850)	NMS	BHVP	86.8	80.1	83.3
(850)	PNMS	PLS	85.6	81.5	83.5
(850,1000)	PNMS	BHVP	85.5	83.6	84.5
(850,1000)	NMS	BHVP	86.2	82.7	84.4
(850,1000)	PNMS	PLS	84.9	83.6	84.3

TABLE II: The comparison with the-state-of-the-art on ICDAR2015 Incidental Scene Text dataset.

Methods	Precision (%)	Recall (%)	Hmean (%)
HUST [30]	44.0	37.8	40.7
Zhang et al. [20]	70.8	43.1	53.6
RRPN [28]	82.2	73.2	77.4
WordSup [19]	79.3	77.0	78.2
EAST [11]	83.3	78.3	80.7
Deep direct regression [9]	82.0	80.0	81.0
R^{2}CNN [12]	85.6	79.7	82.5
FSTN [10]	**88.6**	80.0	84.1
SLPR (Ours)	85.5	**83.6**	**84.5**

shapes. In recent years, other NMS approaches were investigated, e.g., locality-aware NMS [11], inclined NMS [12], Mask-NMS [10] and polygonal NMS (PNMS) [14]. As we consider the polygon in this study, both NMS and PNMS are compared in our experiment.

IV. Experiments

A. Datasets

1) ICDAR2015 Incidental Scene Text.: ICDAR2015 Incidental Scene Text dataset [30] is commonly used benchmark for detecting arbitrary-angle quadrangular text lines. It contains 1000 images for training, 500 images for testing. Some words which are too short, or unclear is annotated as don't cared samples.

2) CTW1500: Curve text dataset (CTW1500) is constructed by Yuliang et al. [14]. Different from traditional text datasets, a text line is labelled by a polygon with 14 points.

B. Implementation Details

Since our proposed SLPR can be applied to any 2-step object detection framework. We adopted Faster R-CNN in ICDAR2015 Incidental Scene Text. And because [14] also proposed a 2-step framework based on R-FCN while presenting the CTW1500 dataset, to perform a fair comparison, we directly used the network in [14] from https://github.com/Yuliang-Liu/Curve-Text-Detector. All experiments were implemented in Caffe [31] by using the NVIDIA GTX 1080Ti GPU.

1) ICDAR2015 Incidental Scene Text.: For Faster R-CNN structure, we used an additional 64^2 anchor and replaced RoIPool with RoIAlign [32] because the text line is smaller than other objects. We set anchor scales as $[64^2, 128^2, 256^2, 512^2]$ and set ratios as [0.5, 1, 2]. The base network is VGG16 [33], which is initialized by the pre-trained model on ImageNet database. We used stochastic gradient descent (SGD) with back-propagation and the maximum iteration was 20×10^4. Learning rates started from 10^{-3}, decays to one-tenth every 5×10^4 iterations. We set weight decay as 0.0005, and momentum as 0.9. We used 1000 training incidental images in ICDAR2015 Incidental Scene Text [30] and the 229 training images from ICDAR 2013 to train our network. In order to prevent over-fitting we employed data augmentation. Specifically, we randomly resized the images to $[720, 850, 960, 1200, 1400]$ where the numbers represent the length of the short side, and randomly rotated the images among $[0°, 15°, 30°, ..., 360°]$.

2) CTW1500: We used the curve text detector (CTD) network which is based R-FCN from [14]. [14] also added LSTM units named transverse and longitudinal offset connection (TLOC) to learn the correlation of points. But we removed it. As we only used PLS to restore polygon, Eq. (4) was modified as:

$$L_{\text{SLPRB}} = \frac{1}{4n}\left[\lambda_{hw} I(\frac{h}{w} > k) \sum_{j=1}^{2n} L_{\text{Reg}}(x_{v_j}, x^*_{v_j}) + I(\frac{h}{w} < \frac{1}{k}) \sum_{i=1}^{2n} L_{\text{Reg}}(y_{h_i}, y^*_{h_i})\right]$$

$I(z)$ equals to 1 when z is true, otherwise 0. h and w are the height and width of the rectangle. Because most of the texts in this dataset are horizontal text line, to solve the imbalance between horizontal and vertical samples, we added $\lambda_{hw} = 4$ to balance the losses between them. And when h is close to w , the text line may be judged as horizontal or vertical, so we set k as 0.8. The base network is ResNet-50 [34], which is initialized by the pre-trained model on ImageNet database. The max iteration was 8×10^4. The learning rate in this experiment was always 10^{-3}. We set weight decay as 0.0005, and momentum as 0.9. To conduct a fair comparison, we only used the training set in CTW1500 to train our network and did not use data augmentation.

C. Results

1) ICDAR2015 Incidental Scene Text: Table I shows the results of SLPR system with different settings. First, for the restoration of the quadrangle for the text region, BHVP using all the points can achieve better results than PLS using only the long-side points. Second, even we aim to detect the quadrangle in this dataset, PNMS still outperforms NMS. Finally, the use of multi-scale is one way to improve detection performance

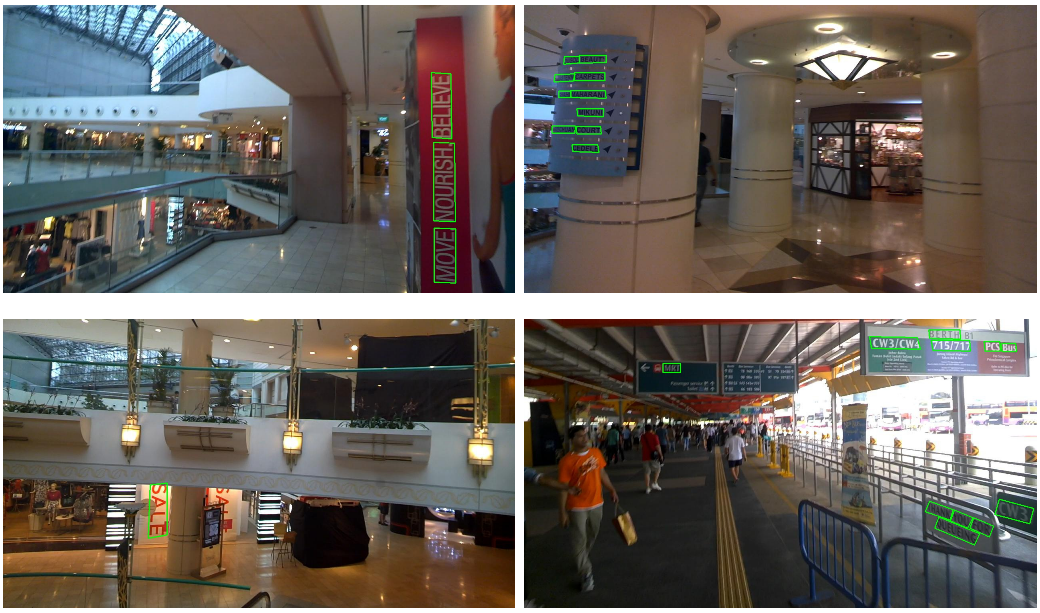

Fig. 5: The detection results on ICDAR2015 Incidental Scene Text dataset.

TABLE III: The performance comparison of SLPR method with different NMS settings on CTW1500 dataset.

NMS Method	Precision (%)	Recall (%)	Hmean (%)
NMS0.1	81.2	64.3	71.8
NMS0.2	81.0	68.7	74.3
NMS0.3	80.1	70.1	**74.8**
PNMS0.1	80.5	69.5	74.6
PNMS0.2	78.8	70.4	74.4
PNMS0.3	75.6	71.3	73.4

TABLE IV: The comparison with CTD on CTW1500 dataset.

Method	Precision (%)	Recall (%)	Hmean (%)
CTD+TLOC [14]	77.4	69.8	73.4
CTD [14]	74.3	65.2	69.5
SLPR (ours)	80.1	70.1	**74.8**

on different target sizes. We also test the multi-scale results of our system at (850, 1000), which yields about 1% absolute improvement of Hmean measure. Fig. 5 lists several challenging examples of detection results on ICDAR2015 Incidental Scene Text dataset. Table II gives the comparison of SLPR with state-of-the-art results on ICDAR2015 Incidental Scene Text. We can observe that our method achieved the competitive results on this dataset.

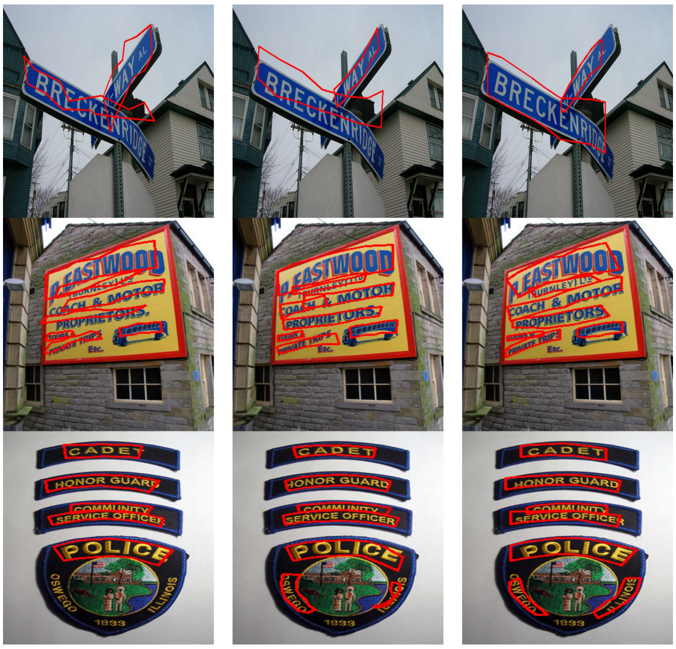

Fig. 6: The detection results on CTW1500 dataset. From left to right: CTD, CTD+TLOC and SLPR.

2) CTW1500: Table III shows the results of our method with different NMS settings. Different from the observation in ICDAR2015 Incidental Scene Text, our method achieved the best result on NMS0.3, namely the traditional NMS method with the threshold 0.3 for calculating the IoU (Intersection-over-Union). Table IV lists the results of our method compared with CTD and CTD+TLOC. We removed TLOC from [14] as our base network which is the same as CTD. Clearly, the Hmean performance of our SLPR method could be increased by 5.3% over the CTD method, demonstrating the effectiveness of our simple rules to set the regression points. Even compared with the CTD+TLOC method with an additional LSTM network, SLPR still achieved 1.4% improvement of Hmean performance. Fig. 6 gives several examples of the detection results of CTD, CTD+TLOC and our SLPR. We can observe that our method generated smoother regions and better detection results compared with CTD, which implied that the proposed SLPR can better handle the arbitrary-oriented case due to the novel design of the horizontally and vertically symmetrical scanning using sliding lines.

V. Conclusion

In this study, we propose a novel SLPR method for the text detection in arbitrary-shape case. Compared with the curve text detection method CTD+TLOC [14], SLPR is more concise without using LSTM and obtains better performance. In the traditional quadrangle dataset (ICDAR2015 Incidental Scene Text), SLPR also achieves the state-of-the-art performance.

Acknowledgment

This work was supported in part by the National Key R&D Program of China under contract No. 2017YFB1002202, the National Natural Science Foundation of China under Grants No. 61671422 and U1613211, the Key Science and Technology Project of Anhui Province under Grant No. 17030901005, and MOE-Microsoft Key Laboratory of USTC.

References

[1] Z. Zhong, L. Jin, S. Zhang, and Z. Feng, "Deeptext: A unified framework for text proposal generation and text detection in natural images," *arXiv preprint arXiv:1605.07314*, 2016.

[2] M. Liao, B. Shi, X. Bai, X. Wang, and W. Liu, "Textboxes: A fast text detector with a single deep neural network." in *AAAI*, 2017, pp. 4161–4167.

[3] Z. Zhong, L. Sun, and Q. Huo, "Improved localization accuracy by locnet for faster r-cnn based text detection," in *Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on*, vol. 1. IEEE, 2017, pp. 923–928.

[4] A. Gupta, A. Vedaldi, and A. Zisserman, "Synthetic data for text localisation in natural images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2315–2324.

[5] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.

[6] X.-C. Yin, W.-Y. Pei, J. Zhang, and H.-W. Hao, "Multi-orientation scene text detection with adaptive clustering," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 9, pp. 1930–1937, 2015.

[7] L. Kang, Y. Li, and D. Doermann, "Orientation robust text line detection in natural images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 4034–4041.

[8] Y. Liu and L. Jin, "Deep matching prior network: Toward tighter multi-oriented text detection," *arXiv preprint arXiv:1703.01425*, 2017.

[9] W. He, X.-Y. Zhang, F. Yin, and C.-L. Liu, "Deep direct regression for multi-oriented scene text detection," *arXiv preprint arXiv:1703.08289*, 2017.

[10] Y. Dai, Z. Huang, Y. Gao, and K. Chen, "Fused text segmentation networks for multi-oriented scene text detection," *arXiv preprint arXiv:1709.03272*, 2017.

[11] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, and J. Liang, "East: An efficient and accurate scene text detector," *arXiv preprint arXiv:1704.03155*, 2017.

[12] Y. Jiang, X. Zhu, X. Wang, S. Yang, W. Li, H. Wang, P. Fu, and Z. Luo, "R2cnn: Rotational region cnn for orientation robust scene text detection," *arXiv preprint arXiv:1706.09579*, 2017.

[13] Z. He, Y. Zhou, Y. Wang, and Z. Tang, "Sren: Shape regression network for comic storyboard extraction." in *AAAI*, 2017, pp. 4937–4938.

[14] L. Yuliang, J. Lianwen, Z. Shuaitao, and Z. Sheng, "Detecting curve text in the wild: New dataset and new solution," *arXiv preprint arXiv:1712.02170*, 2017.

[15] C. Kheng Chng and C. S. Chan, "Total-text: A comprehensive dataset for scene text detection and recognition," *arXiv preprint arXiv:1710.10400*, 2017.

[16] U. Pal, P. P. Roy, N. Tripathy, and J. Lladós, "Multi-oriented bangla and devnagari text recognition," *Pattern Recognition*, vol. 43, no. 12, pp. 4124–4136, 2010.

[17] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*. IEEE, 2013, pp. 6645–6649.

[18] J. Zhang, J. Du, and L. Dai, "A gru-based encoder-decoder approach with attention for online handwritten mathematical expression recognition," *arXiv preprint arXiv:1712.03991*, 2017.

[19] H. Hu, C. Zhang, Y. Luo, Y. Wang, J. Han, and E. Ding, "Wordsup: Exploiting word annotations for character based text detection," *arXiv preprint arXiv:1708.06720*, 2017.

[20] Z. Zhang, C. Zhang, W. Shen, C. Yao, W. Liu, and X. Bai, "Multi-oriented text detection with fully convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4159–4167.

[21] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440.

[22] J. Zhang, J. Du, S. Zhang, D. Liu, Y. Hu, J. Hu, S. Wei, and L. Dai, "Watch, attend and parse: An end-to-end neural network based approach to handwritten mathematical expression recognition," *Pattern Recognition*, 2017.

[23] Y. Wu and P. Natarajan, "Self-organized text detection with minimal post-processing via border learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5000–5009.

[24] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[25] J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: Object detection via region-based fully convolutional networks," in *Advances in neural information processing systems*, 2016, pp. 379–387.

[26] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779–788.

[27] P. He, W. Huang, T. He, Q. Zhu, Y. Qiao, and X. Li, "Single shot text detector with regional attention," *arXiv preprint arXiv:1709.00138*, 2017.

[28] J. Ma, W. Shao, H. Ye, L. Wang, H. Wang, Y. Zheng, and X. Xue, "Arbitrary-oriented scene text detection via rotation proposals," *arXiv preprint arXiv:1703.01086*, 2017.

[29] C. Tensmeyer, B. Davis, C. Wigington, I. Lee, and B. Barrett, "Pagenet: Page boundary extraction in historical handwritten documents," *arXiv preprint arXiv:1709.01618*, 2017.

[30] 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 *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*. IEEE, 2015, pp. 1156–1160.

[31] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 2014, pp. 675–678.

[32] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," *arXiv preprint arXiv:1703.06870*, 2017.

[33] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[34] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.