electronics

MDPI

Article

Scene Text Detection with Polygon Offsetting and Border Augmentation

Thananop Kobchaisawat [1], Thanarat H. Chalidabhongse [1,2,3,*] and Shin'ichi Satoh [4]

1 Department of Computer Engineering, Chulalongkorn University, Bangkok 10330, Thailand; thananop.k@student.chula.ac.th
2 Center of Excellence in Infrastructure Management, Chulalongkorn University, Bangkok 10330, Thailand
3 Research Group on Applied Computer Engineering Technology for Medicine and Healthcare, Chulalongkorn University, Bangkok 10330, Thailand
4 National Institute of Informatics, Tokyo 101-8430, Japan; satoh@nii.ac.jp
* Correspondence: thanarat.c@chula.ac.th

Received: 1 November 2019; Accepted: 23 December 2019; Published: 8 January 2020

Abstract: Scene text localization is a very crucial step in the issue of scene text recognition. The major challenges—such as how there are various sizes, shapes, unpredictable orientations, a wide range of colors and styles, occlusion, and local and global illumination variations—make the problem different from generic object detection. Unlike existing scene text localization methods, here we present a segmentation-based text detector which can detect an arbitrary shaped scene text by using polygon offsetting, combined with the border augmentation. This technique better distinguishes contiguous and arbitrary shaped text instances from nearby non-text regions. The quantitative experimental results on public benchmarks, ICDAR2015, ICDAR2017-MLT, ICDAR2019-MLT, and Total-Text datasets demonstrate the performance and robustness of our proposed method, compared to previous approaches which have been proposed.

Keywords: scene text detection; curved text detection; convolutional neural networks

1. Introduction

Automatic scene text localization is a key part in many practical daily life applications, such as instant language translation, autonomous driving, image retrieval, scene text understanding, and scene parsing. Despite its similarity to the traditional OCR on scanned documents, scene text is much more challenging due to a large variation of text styles, sizes, orientations, and a wide range of complex backgrounds, where together with occlusions, it makes it challenging to locate scene texts from images. Therefore, accurate and robust scene text detection is still an interesting research challenge.

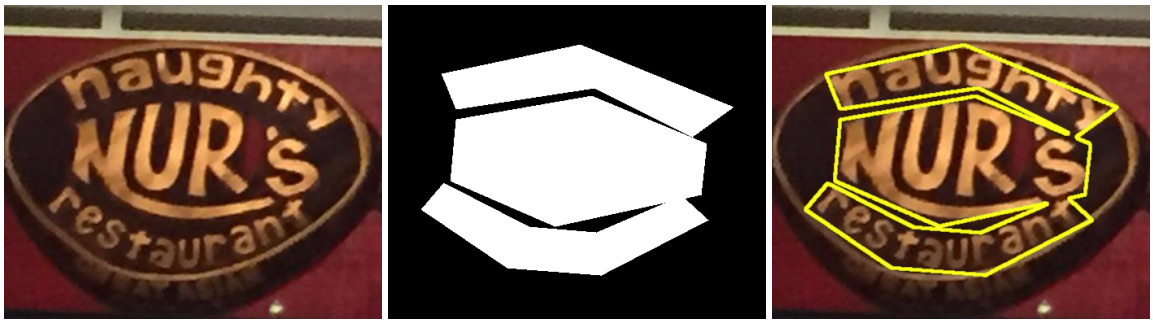
With the great success of convolutional neural networks (CNN) used in object detection, instance segmentation, and semantic segmentation problems, many scene text detectors based on object detection [1–7] and instance segmentation [8,9] have recently shown promising results. Unfortunately, some methods failed in some complex cases, such as in the case of arbitrarily shaped and curved texts, which is difficult to represent with a single rectangle or quadrangle used in generic object detectors, as shown in Figure 1.

As recent developments in pixel labelling problems have gained interest, in this paper we present a semantic segmentation-based text detector which can detect text in various shapes; however, semantic segmentation can be used to label the text regions, and it might not be able to distinguish text instances which are very close, thus resulting in a single merged text instance, as shown in Figure 2. To deal with this problem, in addition to representing the text instances using only text pixel masks, our proposed method also learns the text's outer border and offset masks. The text's outer border masks represent

each text instance boundary, while the offset masks represent the distance between the shrunk text instance polygon border and its original shape. Both can greatly help to separate the adjacent text instances.

(**a**) (**b**) (**c**) (**d**)

Figure 1. Different representations of text instances: (**a**) Axis-aligned bounding boxes; (**b**) oriented bounding box; (**c**) quadrangles; and (**d**) text polygons. As shown in the images, the polygon representation is able to precisely express the location, scale, and bending of the curved text, while the others cannot give accurate text instance locations.



(**a**) Original image. (**b**) Ground truth mask. (**c**) Inference text instance.

Figure 2. Merged detected text instance due to the connected segmentation map.

In this paper, we present a pipeline semantic segmentation-based text detector and extended text representation. We first used the ResNet-50 [10] combined with the feature pyramid network (FPN) [11] as a backbone to extract features from input images. Each scale feature was combined and up-sampled into the input image's original sizes. Instead of using single direct upsampling to the original size, we applied the consecutive upsampling modules, which improved the overall training stability and output segmentation results. The text instances were independently predicted on each scale using a simple connected component analysis. The text border information was also used to ensure a clear cut between each text instance. By using a simple polygon non-maximum suppression over the entire detected text instances, we obtained the final text locations. The experimental result is promising in terms of detection accuracy on the standard test benchmarks, including ICDAR2015 [12], ICDAR2017-MLT [13], ICDAR2019-MLT [14], and Total-Text [15].

The contribution of this paper can be summarized as follows:

1. In addition to the text pixel masks, we also employed the offset masks and text instances border to represent the text instances, which improves the distinguishing of contiguous text instances.
2. A post-processing pipeline to predict text instances location was proposed, which apparently yields higher accuracy while impacting slightly on inference time.
3. The experimental results show our proposed method that has a competitive accuracy on standard benchmarks.

The remainder of this paper is organized as follows: Section 2 discusses the previous text detection methods. In Section 3, the proposed method is described, including the text representation, network structure, loss function, and text instance inference details. Section 4 discusses the quantitative

experimental results on standard benchmark datasets and the effect of border augmentation. Section 5 draws final conclusions and directions for future work.

2. Related Works

Text detection is still a popular and active research area in the computer vision field. In this section, we introduce existing scene text detection methods, which can be categorized into three main categories: the connected component-based [16–18], detection-based [3,4,7,19,20], and semantic segmentation-based methods [9,21,22].

Connected component-based methods: Previous works in scene text detection have been dominated by bottom-up methods which are usually built on stroke or character detection. Individual character is detected based on the observation of scene text characteristics, such as colors, stroke width, enclosure contours, and geometric properties. These properties lead to classic text detection features, such as Stroke Width Transform [16] and Maximally Stable Extremal Regions (MSER) [17]. Then, the detected characters are grouped into words or text lines. The grouping methods usually adopt some defined heuristic or learned rules to remove false detections. Nevertheless, the connected component-based method might not be robust in complex scenarios due to the uncertain scene conditions, in terms of text distortion, orientation, occlusion, reflection, and noise.

Detection-based methods: Convolutional neural networks (CNN) have demonstrated strong capability in object detection problems. Many recent object detection frameworks, such as proposal-based detectors (Faster-RCNN [23], Mask-RCNN [24]) and regression-based detectors (Single Shot MultiBox Detector : SSD [25], You Only Look Once : YOLO [26]) have shown splendid performance in various practical applications. Both proposal and regression-based methods have been shown to produce impressive results in terms of speed and accuracy on many famous object-detection benchmark datasets. However, scene text has a different context when compared to generic objects. More specifically, text is significantly distinct from generic objects in many aspects, such as the various aspect ratios and non-axis-aligned orientation. These specific characteristics make it difficult to apply the existing object detection algorithms directly.

To handle multi-oriented scene text, R2CNN [3] employed rotatable anchors based on Faster-RCNN. TextBoxes++ [7] modified the convolution kernel shapes and SSD anchor boxes to effectively handle various text aspect ratios, especially long text. LOMO [19] and SPCNET [20] formulated the text detection problem into a instance segmentation problem by using Mask-RCNN as a base to generate both the axis-aligned bounding box and text segmentation mask, which was able to deal with arbitrary text shapes. EAST [4] directly applied regression from CNN features to form up-text quadrangles without using the anchor box mechanism.

Semantic segmention-based methods: Instead of detecting text in character- or word-bounding box levels, the intentions of the methods in this group are to label the text and non-text regions at a pixel level. PixelLink [9] proposed a method which represents the text instances in eight connected text pixel maps, and directly infers the word level boxes by using a simple connected component. TextSnake [21] presented the arbitrary shapes text detector by using the text region-based center line, together with geometry attribute representation. The text lines were reconstructed by a striding algorithm from central text line point lists. PSENet [22] utilized polygon offsetting and multi-scaled text segmentation maps to separate and detect text. These methods usually vary in the way they express text blobs and the method used to distinguish between each text instance.

3. Proposed Method

This section presents details of the proposed text detector, including the text representation, network structure, loss function, and text instance inference details of our method.

3.1. Text Representation

In many previous works, scene texts are typically represented by bounding boxes, which are 2D rectangles containing texts. Some use axis-aligned bounding boxes, which are aligned with the axes of the coordinate system, whereas some use oriented bounding boxes, which are arbitrarily oriented rectangles. To make the bounding box fit the text regions more accurately, quadrangles are used. However, in some difficult cases they are still not capable of precisely capturing text instances, such as the texts in Figure 2 which are aligned in a curved shape. To cope with this limitation, some methods have used text masks to represent arbitrarily shaped text instances. Nonetheless, we found that this might not be able to separate the very close text instances. Thus, in this work, instead of using only shrunk text masks, we combine the shrunk text masks and offset masks, which are offsetting polygons that can be either inward or outward. In addition, to make the network able to capture different text sizes, each original text polygon is offset into multiple scaled polygons based on its area and perimeter. If we consider the text instance t_i and polygon scaling factor α, the polygon offsetting ratio d_i can be calculated from the following equation:

$$d_i = \frac{\alpha * Area(t_i)}{Perimeter(t_i)} \tag{1}$$

The ground truth for each image consists of three components: text masks g_{tm}, which are filled offsetting polygons; offset masks g_{om}, where each polygon area is filled with the d_i; and outer border masks g_{bm}, which represent each text instance border. This text representation is illustrated in Figure 3.

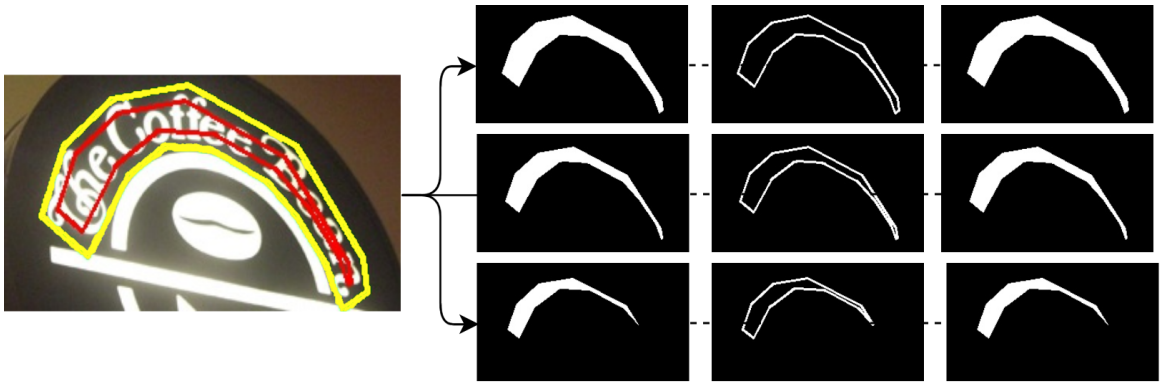


Figure 3. Proposed text representation: (**a**) Text polygon t_i (yellow) and its offset version to_i (red); (**b**) multi-scaled text masks; (**c**) border masks; and (**d**) offset masks.

3.2. Network Structure

In this work, a fully convolutional neural network based on ResNet-50 was used as our core network. To avoid loss of spatial information, we utilized the feature pyramid network (FPN) [11]. FPN has demonstrated a significant ability to handle multi-scaled semantic information in many recent works. The lateral connections are built between deep and shallow feature maps in order to generate high-quality feature maps from low-level and high semantic features. Since deconvolution causes the checkerboard pattern on the output text masks, we utilized bilinear interpolation to upsample feature maps to the desired input size. The output from the network contains three branches, text masks, offset masks, and border masks. The overall network structure is shown in Figure 4.

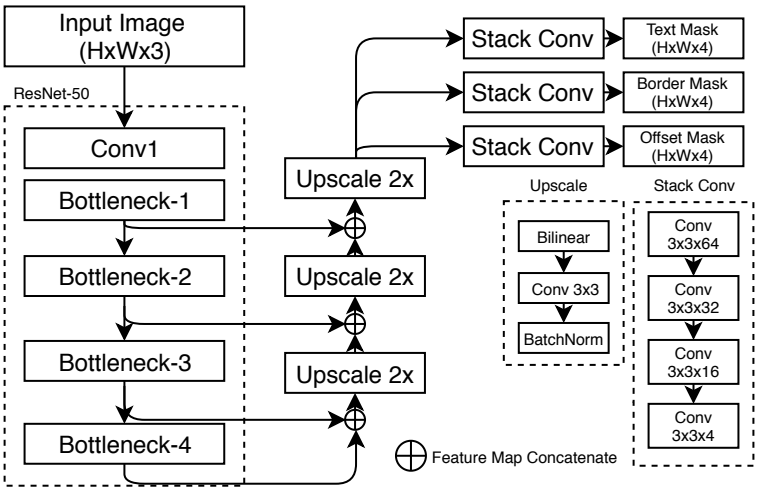


Figure 4. Network structure.

3.3. Loss Function

The output from the network consists of three components: text, offset, and border masks. For the text and border masks, since the ratio between text, non-text pixels, and especially border pixels in scene text images are greatly imbalanced, making the network tends to put more emphasis on non-text pixels, yielding false detections when using standard binary cross-entropy loss. Since our problem aims to maximize the overlapping regions between ground truth and the predicted mask, there are many region-based losses which can be applied to cope with this problem, such as weighted cross-entropy [27], tversky loss [28], and focal loss [29]. However, to maximize classification loss efficiency, the parameters need to be tuned. To address this problem, we utilized dice loss, which is non-parametic loss, for both the text mask L_{tm} and border masks L_{bm}, which can be formulated as follows:

$$L_{tm}(o_{tm}, g_{tm}) = \frac{2 * \sum o_{tm}(x,y) * g_{tm}(x,y)}{\sum o_{tm}(x,y) \sum g_{tm}(x,y)} \tag{2}$$

$$L_{bm}(o_{bm}, g_{bm}) = \frac{2 * \sum o_{bm}(x,y) * g_{bm}(x,y)}{\sum o_{bm}(x,y) \sum g_{bm}(x,y)}, \tag{3}$$

where $o_{tm}(x,y)$, $o_{bm}(x,y)$, $g_{tm}(x,y)$, and $g_{bm}(x,y)$ represent the pixel value at (x,y) on the output text masks, and border masks on their ground truth masks, respectively.

For the offset masks, to ensure good training stability, we employed smooth L_1 loss. The loss function for the offset mask can be formulated as:

$$L_{om}(o_{om}, g_{om}) = \sum SmoothL1(o_{om}(x,y) - g_{om}(x,y)). \tag{4}$$

In this work, we combined all losses into multi-task loss, L, which can be defined as:

$$L = \lambda_1 L_{tm} + \lambda_2 L_{bm} + \lambda_3 L_{om}, \tag{5}$$

where λ_1, λ_2, and λ_3 weigh the importance between text, border, and offset masks, respectively.

3.4. Text Instance Inference

After the forward pass, the network outputs are multi-scaled text masks, border, and offset masks. We employed thresholding on both text and border masks. To ensure a clear cut between each text instance, the border augmentation was used in combination with standard connected component analysis to detect and separate each component.

Border augmentation is a simple and fast operation between corresponding output text masks o_{tm} and border masks o_{bm}, which can be defined as.

$$o_{ba}(x,y) = \begin{cases} 1 & \text{if } o_{tm}(x,y) = 1 \text{ and } o_{bm}(x,y) = 0 \\ 0 & \text{otherwise,} \end{cases} \tag{6}$$

where o_{ba} represents the output text border augmented masks.

We then calculated the text instance score using polygon scoring, which can be defined as:

$$P(t_i) = \frac{1}{N} \sum_{(x,y) \in t_i} {}_{tm}(x,y), \tag{7}$$

where P and N represent polygon scoring and the number of pixels in text instance t_i, respectively.

Each text component is restored back to its original size by using the offset value $v(t_i)$, which can be calculated from output offset masks o_{om}, as follows:

$$v(t_i) = \underset{(x,y) \in t_i}{\text{median}}\{o_{om}(x,y)\}. \tag{8}$$

Given such text polygon candidates with their associated scoring probabilities, we performed polygon non-maximum suppression to discard the overlapping detections, thus obtaining the final set of text instances.

4. Experiments

In order to evaluate the performance of the proposed method, we conducted a quantitative test on standard benchmarks for scene text detection and compared with the existing methods.

4.1. Datasets

SynthText [30] is a large-scale, computer-generated dataset. This dataset contains about 800,000 images. The images were created by fusing natural background images with rendered text. In order to make the text look more realistic, artificial transformations were applied, such as random fonts, sizes, colors, and orientations. In this dataset, text instances were annotated in both word and character levels. For this work, we utilized this dataset to pre-train our model.

ICDAR2015 [12] first appeared in the 2015 incidental scene text detection robust reading competition. The images in this dataset were taken by Google Glasses without taking image quality and viewpoint into consideration. This dataset contained small, blurred, and multi-oriented text instances. There were 1500 images in total, which can be separated into 1000 training and 500 testing images. The text instances from this dataset were labeled in word-level quadrangles.

ICDAR2017-MLT [13] is a large, multi-lingual scene text dataset. This dataset included 7200 training, 1800 validation, and 9000 testing images, containing text from nine languages. The text instances from this dataset were annotated at word level by using four vertices quadrangles.

ICDAR2019-MLT [14] is the latest multi-lingual scene text dataset. This real-world dataset consisted of 10,000 training and 10,000 testing images containing text from 10 languages. The text instances from this dataset were annotated at word level by using four vertices quadrangles, as in ICDAR2017-MLT.

Total-Text [15] is a dataset which contains both horizontal and multi-oriented text instances. The dataset specially features curved text, which is occasionally presented in other benchmarks. The dataset is split into training and testing sets with 1255 and 300 images, respectively.

4.2. Implementation Details

We first trained our model on the SynthText dataset for 1 epoch, and continued to train and fine-tune on benchmark datasets until the model converged. The stochastic gradient descent (SGD)

with momentum was used by setting the momentum and weight decay to 0.9 and 5×10^{-4}. During the training on SynthText, the learning rate was initially set to 10^{-3}, which then decayed to 10^{-4} until the loss was stable. At the beginning, the batch size was set to 1, then increased to 4. We believed that the adaptive training batch size could slightly boost the model accuracy. From the experiment, the polygon scaling factor α was set as $[0.6, 0.75, 0.9, 1.25]$. The λ weighted the importance between text, border masks, and offset, which were set to 1, 1, and 0.1, respectively.

After we obtained the pre-trained weights from SynthText, the model was then fine-tuned on standard benchmarks, ICDAR2015, ICDAR2017-MLT, ICDAR2019-MLT, and Total-Text. The learning rate was initially set to 10^{-4} and decreased by a factor of 10 at every 250 epoches. We set the batch size as equal to 4, and trained the models on 4 $\times$ NVIDIA Tesla K-80. For the ICDAR datasets, the non-readable text regions, which were labeled as "###", were not used during the training. To correct the imbalance ratio between the number of text and non-text pixels, for each text scale mask we adopted Online Hard Negative Mining (OHEM) [31] with a ratio of 1:3. As data augmentation was crucial to increasing the robustness of the algorithm, the following augmentations were applied:

- Photometric distortion, as described in [32].
- Image rotation in range $[-30°, 30°]$, horizontal and vertical flip with a probability of 0.5.
- Image size re-scale in range [0.5, 3].
- Randomly cropping image to 512 $\times$ 512.
- Mean and standard deviation normalization.

In this work, we also introduced a mosaic data augmentation technique by randomly combining multiple image patches into a new training image. This technique can increase the amount of training data and algorithm robustness. The sample of input and output images is shown in Figure 5.

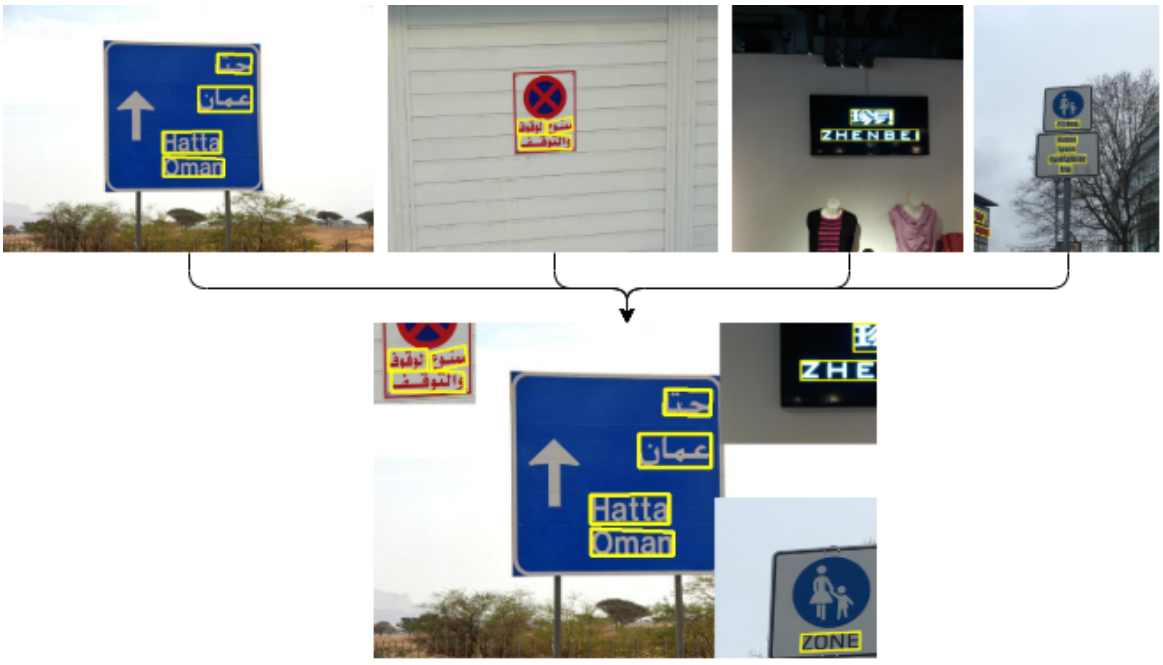


Figure 5. Result of a mosaic data augmentation example from four input images (best viewed in color).

The output results for each dataset depend on its ground truth representation. For the ICDAR datasets, which represented text instances in quadrangles, we calculated the minimal area rectangle by using the standard OpenCV function to acquire the four-point output. In the case of the arbitrary shape text dataset, the text polygons and masks were considered as outputs.

4.3. Results

The results were evaluated by standard evaluation, depending on the dataset corresponding protocol.

4.3.1. Multi-Oriented English Text

We compared our proposed method with previous works on the ICDAR2015 dataset. The model was fine-tuned by using a pre-trained weight from SynthText, and was further trained for 200 epoches. In the testing stage, we scaled the longer side of the image to 1280 pixels, while still preserving the image aspect ratio and using only single-scale testing. From the quantitative results list in Table 1, our method gives a competitive result in terms of the f-measure. The samples of several detection results are shown in Figure 6.

Table 1. Experimental results on standard benchmark datasets. P, R, F, and BA denote precision, recall, f-measure, and border augmentation, respectively. The best single-scale test results from each paper are reported.

Method	Dataset											
	ICDAR 2015			ICDAR2017			ICDAR2019			Total-Text		
	P	R	F	P	R	F	P	R	F	P	R	F
CTPN [1]	51.6	74.2	60.9	-	-	-	-	-	-	-	-	-
EAST [4]	80.5	72.8	76.4	-	-	-	-	-	-	-	-	-
SegLink [8]	73.1	76.8	75.0	-	-	-	-	-	-	-	-	-
TextBoxes++ [7]	87.2	76.7	81.7	-	-	-	-	-	-	-	-	-
R2CNN [3]	85.6	79.7	82.5	-	-	-	-	-	-	-	-	-
PixelLink [9]	85.5	82.5	83.7	-	-	-	-	-	-	-	-	-
TextSnake [21]	84.9	80.4	82.6	-	-	-	-	-	-	82.7	74.5	78.4
PSENet [22]	88.7	85.5	87.1	75.4	69.2	72.1	-	-	-	84.0	78.0	80.9
SPCNET [20]	88.7	85.8	87.2	73.4	66.9	70.0	-	-	-	83.0	82.8	82.9
Pixel-Anchor [33]	88.3	87.1	87.7	79.5	59.5	68.1	-	-	-	-	-	-
PMTD [34]	91.3	87.4	89.3	85.2	72.7	78.5	87.5	78.1	82.5	-	-	-
CRAFT [35]	89.8	84.3	86.9	80.6	68.2	73.9	81.4	62.7	70.9	87.6	79.9	83.6
LOMO [19]	91.2	83.5	87.2	78.8	60.6	68.5	87.7	79.8	83.6	87.6	79.3	83.3
Our Method (ResNet-50 without BA)	87.2	84.9	86.0	76.8	67.4	72.1	83.3	72.4	77.9	85.2	78.2	81.5
Our Method (ResNet-50 with BA)	89.8	86.8	88.1	78.7	69.8	73.4	86.1	75.7	80.9	88.2	79.9	83.5

4.3.2. Multi-Oriented and Multi-Language Text

To verify the robustness of our proposed method on multi-language scene text detection, we conducted the experiment on ICDAR2017-MLT and ICDAR2019-MLT datasets. The model weight from SynthText was fine-tuned for 300 epoches on the ICDAR2017-MLT training dataset, and 450 epoches on ICDAR2019-MLT. Since the image sizes in this dataset were not equal, we resized the longer side of the image to 1280 pixels, while still preserving the aspect ratio and test by using only single-scale. The experimental result on this dataset is shown in Table 1. Our method shows good and respectable performance compared to other state-of-the-art methods. Samples of the detection results on ICDAR2017-MLT and ICDAR2019-MLT datasets are shown in Figures 7 and 8, respectively.

4.3.3. Multi-Oriented and Curved English Text

We tested our method's ability to detect curved and arbitrary-oriented texts on the Total-Text dataset. Similar to the experiment on ICDAR2017-MLT and ICDAR2019-MLT, we started from the SynthText pre-trained weights and fine-tuned them on Total-Text for 150 epoches. The experimental results showed that our method surpassed other methods in terms of precision with respectable recall and f-measure. Detailed results are shown in Table 1. Figure 9 shows that our method can detect curved text in various styles, shapes, and orientations.

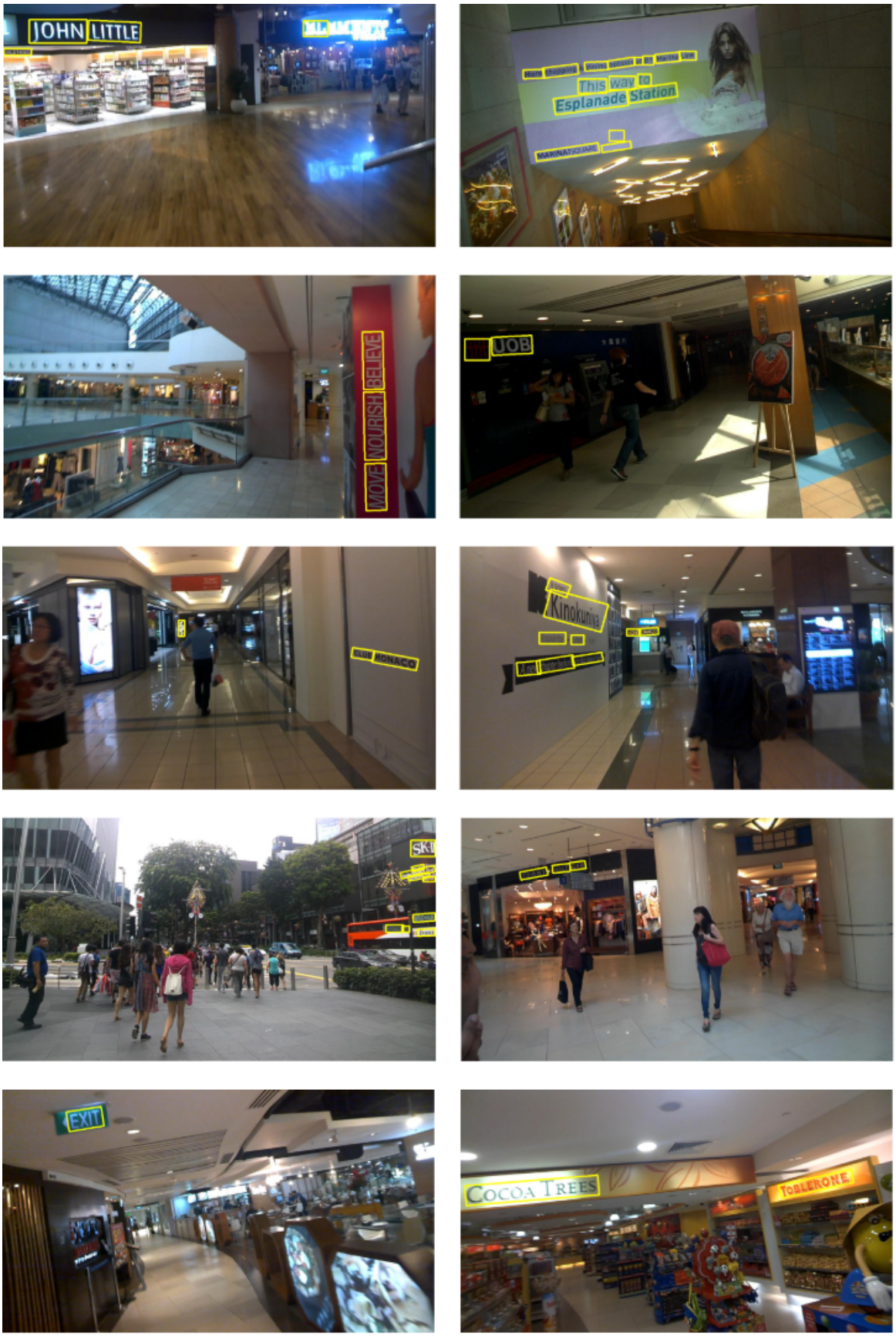


Figure 6. Some example results of our proposed method on ICDAR2015 benchmark datasets (best viewed in color).

Figure 7. Some example results of our proposed method on ICDAR2017 benchmark datasets (best viewed in color).

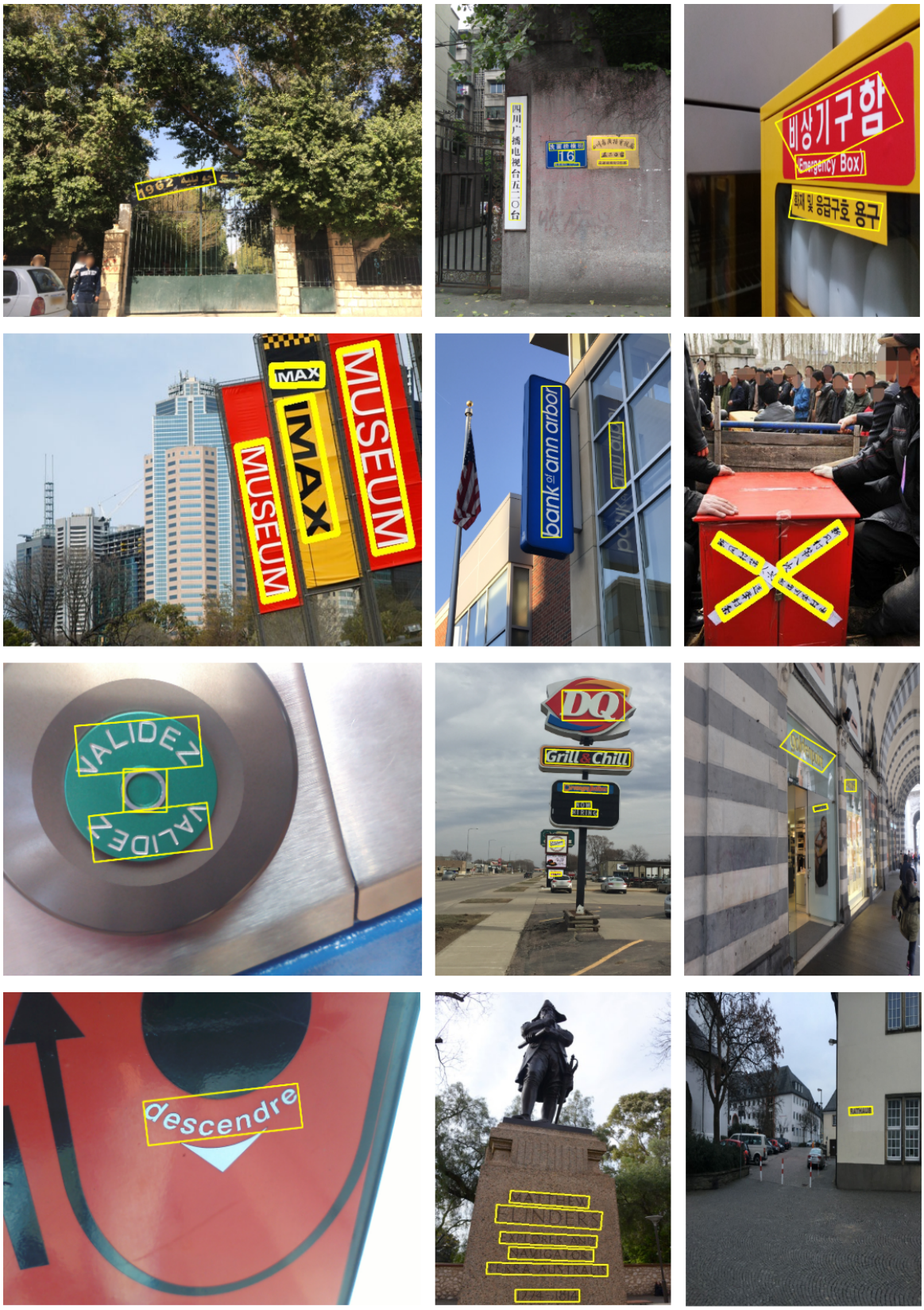


Figure 8. Some example results of our proposed method on ICDAR2019 benchmark datasets (best viewed in color).

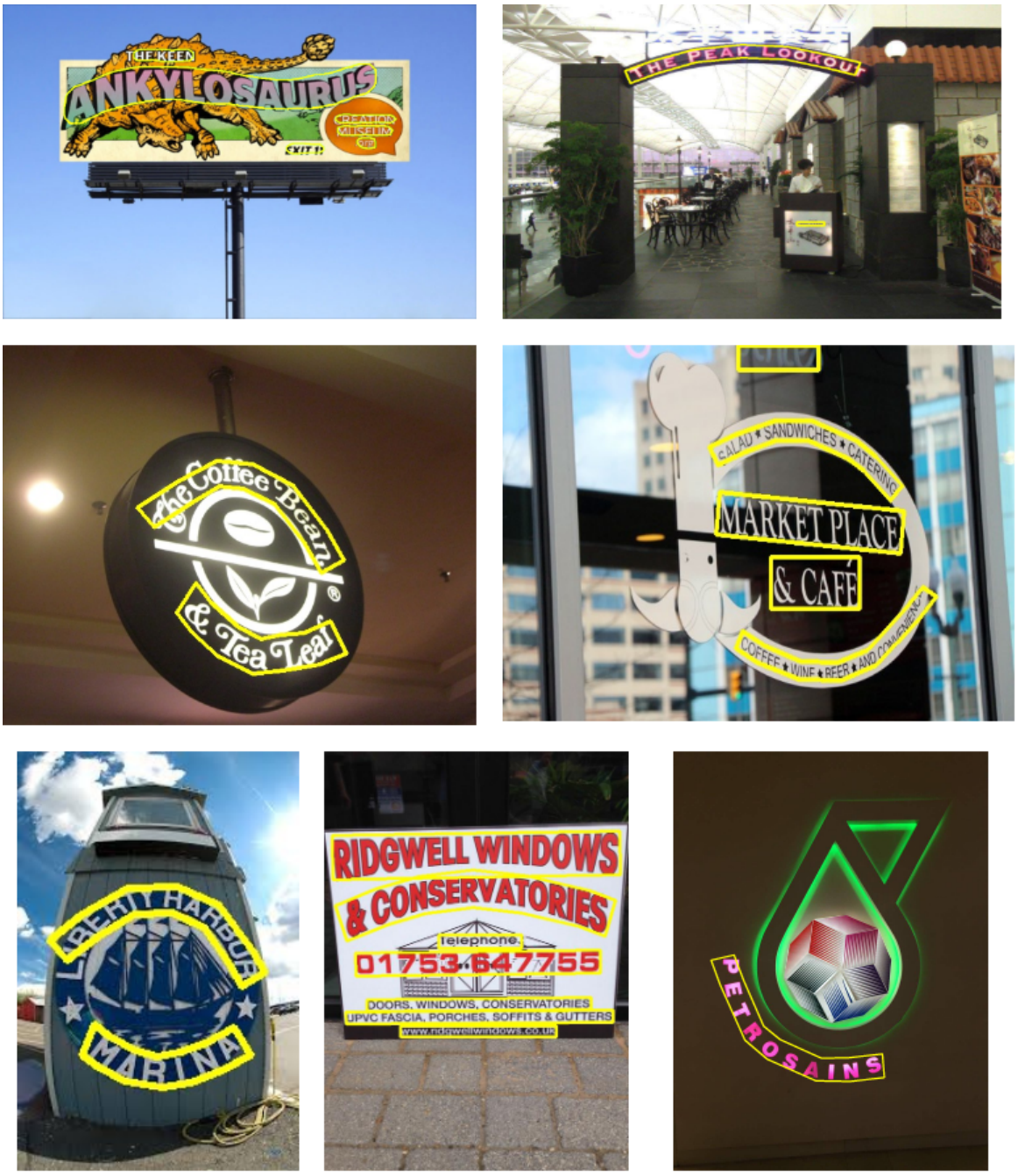


Figure 9. Some example results of our proposed method on Total-Text benchmark datasets (best viewed in color).

4.4. Speed Analysis

We also employed a comparative experiment in terms of text detection speed. All of the experiments were tested on NVIDIA GTX 1080 Ti and Intel i7-4770K.

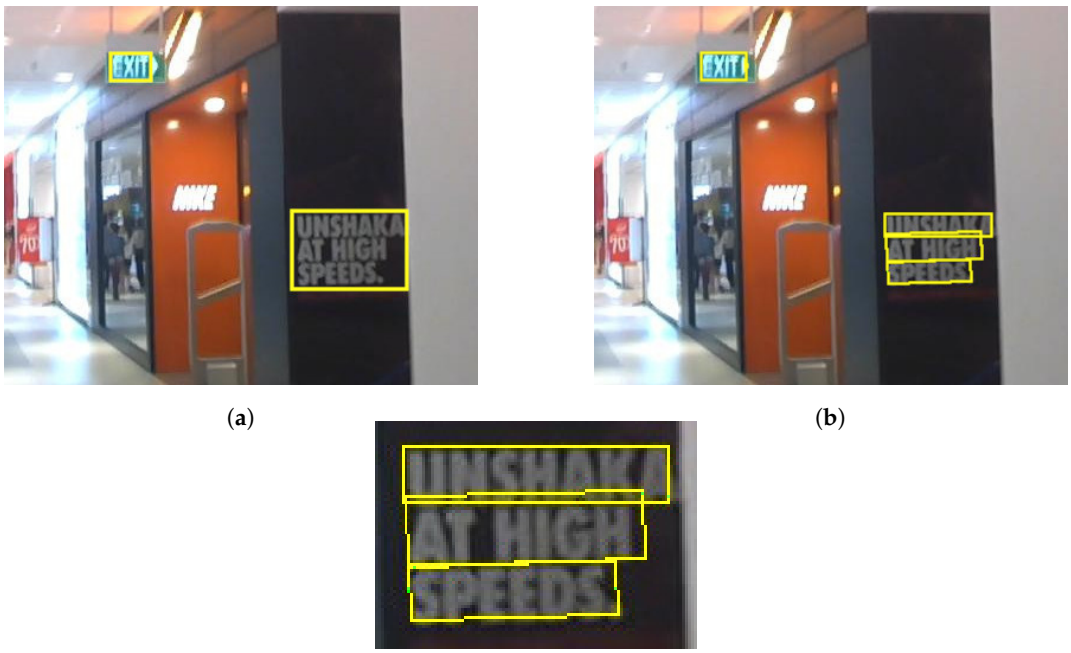
As shown in Table 2, our proposed method gives a good balance between detection speed and accuracy. In the experiment, ResNet-50 and ResNet-34 are considered as the feature extraction backbone to trade off the speed and accuracy. If we change the backbone to ResNet-34, our proposed method gives nearly real-time detection speed.

Table 2. Average text detection speed on standard benchmark datasets.

Method	Dataset and F-Measure results				FPS
	ICDAR2015	ICDAR2017	ICDAR2019	Total-Text	
CTPN [1]	60.9	-	-	-	7.5
EAST [4]	76.4	-	-	-	17.1
SegLink [8]	75.0	-	-	-	12.2
TextBoxes++ [7]	81.7	-	-	-	13.2
R2CNN [3]	82.5	-	-	-	-
PixelLink [9]	83.7	-	-	-	-
TextSnake [21]	82.6	-	-	78.4	12.7
PSENet [22]	87.1	72.1	-	80.9	9.6
SPCNET [20]	87.2	70.0	-	82.9	-
Pixel-Anchor [33]	87.7	68.1	-	-	-
PMTD [34]	89.3	78.5	82.5	-	-
CRAFT [35]	86.9	73.9	70.9	83.6	11.2
LOMO [19]	86.0	72.1	77.9	81.5	-
Our method (ResNet-34 without BA)	83.2	67.6	72.5	78.9	26.1
Our method (ResNet-34 with BA)	84.5	68.9	75.4	80.1	25.2
Our method (ResNet-50 without BA)	86.0	72.1	77.9	81.5	18.7
Our method (ResNet-50 with BA)	88.1	73.4	80.9	83.5	17.5

4.5. Border Augmentation

To analyze the adjacent text instance separation capability of our method, we removed the entire border augmentation part and conducted the experiment under the same configurations. As shown in Table 1, the border augmentation can improve the result on all datasets. The sample result of border augmentation is shown in Figure 10.



(**a**) (**b**)

(**c**)

Figure 10. The effect of border augmentation: (**a**) without border augmentation; (**b**) with border augmentation; (**c**) close-up of adjacent text instances. As shown in the images, the border augmentation is able to provide clear-cut and accurate text instances.

5. Conclusions

In this paper, we presented a method based on semantic segmentation which can be used to localize arbitrarily-oriented text in natural scene images. By using shared, multi-scaled convolution features to learn text and offset masks, we were able to effectively pinpoint the locations of exact text instances. The border augmentation mechanism also helps distinguish between adjacent text components. The numerical results on different standard scene text benchmarks show the advantages in terms of speed, while still preserving acceptable accuracy when compared to previously proposed text detectors.

In the future, we will investigate the causes of failed detection and the possibility to build a single and lightweight network for end-to-end scene text localization and recognition.

Author Contributions: Conceptualization, T.H.C.; Data curation, T.K.; Funding acquisition, T.H.C.; Investigation, T.K.; Methodology, T.K.; Resources, T.H.C. and S.S.; Software, T.K.; Supervision, T.H.C. and S.S.; Writing—original draft, T.K.; Writing—review & editing, T.H.C. and S.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research has been supported by The 100th Anniversary Chulalongkorn University Fund for Doctoral Scholarship.

Acknowledgments: This work was supported by the National Institute of Informatics international internship program and the scholarship from "The 100th Anniversary Chulalongkorn University Fund for Doctoral Scholarship". All authors have read and agreed to the published version of the manuscript

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Tian, Z.; Huang, W.; He, T.; He, P.; Qiao, Y. Detecting Text in Natural Image with Connectionist Text Proposal Network. In *Computer Vision—ECCV 2016*; Leibe, B., Matas, J., Sebe, N., Welling, M., Eds.; Springer International Publishing: Cham, Switzerland; pp. 56–72.
2. Liao, M.; Shi, B.; Bai, X.; Wang, X.; Liu, W. TextBoxes: A Fast Text Detector with a Single Deep Neural Network. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017.
3. Jiang, Y.; Zhu, X.; Wang, X.; Yang, S.; Li, W.; Wang, H.; Fu, P.; Luo, Z. R2CNN: Rotational Region CNN for Orientation Robust Scene Text Detection. *arXiv* **2017**, arXiv:1706.09579.
4. Zhou, X.; Yao, C.; Wen, H.; Wang, Y.; Zhou, S.; He, W.; Liang, J. EAST: An Efficient and Accurate Scene Text Detector. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
5. Ma, J.; Shao, W.; Ye, H.; Wang, L.; Wang, H.; Zheng, Y.; Xue, X. Arbitrary-Oriented Scene Text Detection via Rotation Proposals. *IEEE Trans. Multimed.* **2018**, *20*, 3111–3122. [CrossRef]
6. Liao, M.; Zhu, Z.; Shi, B.; Xia, G.S.; Bai, X. Rotation-Sensitive Regression for Oriented Scene Text Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018.
7. Liao, M.; Shi, B.; Bai, X. TextBoxes++: A Single-Shot Oriented Scene Text Detector. *IEEE Trans. Image Process.* **2018**, *27*, 3676–3690. doi:10.1109/TIP.2018.2825107. [CrossRef] [PubMed]
8. 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 (CVPR), Honolulu, HI, USA, 21–26 July 2017.
9. Deng, D.; Liu, H.; Li, X.; Cai, D. PixelLink: Detecting scene text via instance segmentation. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018.
10. He, K.; Zhang, X.; Ren, S.; Sun, J. Identity mappings in deep residual networks. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; pp. 630–645.
11. Lin, T.Y.; Dollar, 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 (CVPR), Honolulu, HI, USA, 21–26 July 2017.

12. Karatzas, D.; Gomez-Bigorda, L.; Nicolaou, A.; Ghosh, S.; Bagdanov, A.; Iwamura, M.; Matas, J.; Neumann, L.; Chandrasekhar, V.R.; Lu, S.; et al. ICDAR 2015 competition on Robust Reading. In Proceedings of the 2015 13th International Conference on Document Analysis and Recognition (ICDAR), Tunis, Tunisia, 23–26 August 2015; pp. 1156–1160; doi:10.1109/ICDAR.2015.7333942. [CrossRef]
13. Nayef, N.; Yin, F.; Bizid, I.; Choi, H.; Feng, Y.; Karatzas, D.; Luo, Z.; Pal, U.; Rigaud, C.; Chazalon, J.; et al. ICDAR 2017 Robust Reading Challenge on Multi-Lingual Scene Text Detection and Script Identification—RRC-MLT. In Proceedings of the 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), Kyoto, Japan, 9–15 November 2017; Volume 01, pp. 1454–1459; doi:10.1109/ICDAR.2017.237. [CrossRef]
14. Nayef, N.; Patel, Y.; Busta, M.; Chowdhury, P.N.; Karatzas, D.; Khlif, W.; Matas, J.; Pal, U.; Burie, J.C.; Liu, C.; et al. ICDAR2019 Robust Reading Challenge on Multi-lingual Scene Text Detection and Recognition—RRC-MLT-2019. *arXiv* **2019**, arXiv:1907.00945.
15. Ch'ng, C.K.; Chan, C.S. Total-Text: A Comprehensive Dataset for Scene Text Detection and Recognition. In Proceedings of the 14th IAPR International Conference on Document Analysis and Recognition ICDAR, Kyoto, Japan, 9–15 November 2017; pp. 935–942; doi:10.1109/ICDAR.2017.157. [CrossRef]
16. Epshtein, B.; Ofek, E.; Wexler, Y. Detecting text in natural scenes with stroke width transform. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 2963–2970; doi:10.1109/CVPR.2010.5540041. [CrossRef]
17. Neumann, L.; Matas, J. Real-time scene text localization and recognition. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 3538–3545; doi:10.1109/CVPR.2012.6248097. [CrossRef]
18. Cho, H.; Sung, M.; Jun, B. Canny Text Detector: Fast and Robust Scene Text Localization Algorithm. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 3566–3573; doi:10.1109/CVPR.2016.388. [CrossRef]
19. Zhang, C.; Liang, B.; Huang, Z.; En, M.; Han, J.; Ding, E.; Ding, X. Look More Than Once: An Accurate Detector for Text of Arbitrary Shapes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019.
20. Xie, E.; Zang, Y.; Shao, S.; Yu, G.; Yao, C.; Li, G. Scene Text Detection with Supervised Pyramid Context Network. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 9038–9045; doi:10.1609/aaai.v33i01.33019038. [CrossRef]
21. 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 The European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018.
22. Wang, W.; Xie, E.; Li, X.; Hou, W.; Lu, T.; Yu, G.; Shao, S. Shape Robust Text Detection With Progressive Scale Expansion Network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018.
23. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN : Towards Real-time Object Detection with Region Proposal Networks. In Proceedings of the 28th International Conference on Neural Information Processing Systems (NIPS'15), Montreal, QC, Canada, 7–12 December 2015; Volume 1, pp. 91–99.
24. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2980–2988; doi:10.1109/ICCV.2017.322. [CrossRef]
25. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018.
26. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
27. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015*; Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F., Eds.; Springer International Publishing: Cham, Switzerland, 2015; pp. 234–241.
28. Salehi, S.S.M.; Erdogmus, D.; Gholipour, A. Tversky Loss Function for Image Segmentation Using 3D Fully Convolutional Deep Networks. In *Machine Learning in Medical Imaging*; Wang, Q., Shi, Y., Suk, H.I., Suzuki, K., Eds.; Springer International Publishing: Cham, Switzerland, 2017; pp. 379–387.

29. Lin, T.; Goyal, P.; Girshick, R.; He, K.; Dollar, P. Focal loss for dense object detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *1*, doi:10.1109/TPAMI.2018.2858826. [CrossRef] [PubMed]
30. Gupta, A.; Vedaldi, A.; Zisserman, A. Synthetic Data for Text Localisation in Natural Images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016.
31. Shrivastava, A.; Gupta, A.; Girshick, R. Training Region-based Object Detectors with Online Hard Example Mining. *arXiv* **2016**, arXiv:1604.03540.
32. Howard, A.G. Some Improvements on Deep Convolutional Neural Network Based Image Classification. *arXiv* **2013**, arXiv:1312.5402.
33. Li, Y.; Yu, Y.; Li, Z.; Lin, Y.; Xu, M.; Li, J.; Zhou, X. Pixel-Anchor: A Fast Oriented Scene Text Detector with Combined Networks. *arXiv* **2013**, arXiv:1811.07432.
34. Liu, J.; Liu, X.; Sheng, J.; Liang, D.; Li, X.; Liu, Q. Pyramid Mask Text Detector. *arXiv* **2013**, arXiv:1903.11800.
35. Baek, Y.; Lee, B.; Han, D.; Yun, S.; Lee, H. Character Region Awareness for Text Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 9365–9374.