



Article

An Algorithm for Scene Text Detection Using Multibox and Semantic Segmentation

Hongbo Qin ¹, Haodi Zhang ¹, Hai Wang ^{2,*}, Yujin Yan ¹, Min Zhang ² and Wei Zhao ¹

¹ Key Laboratory of Electronic Equipment Structure Design, Ministry of Education, Xidian University, Xi'an 710071, China; qhb0920qhb@xidian.edu.cn (H.Q.); haodiz@stu.xidian.edu.cn (H.Z.); yanyujin.xidian@gmail.com (Y.Y.); weizhao@xidian.edu.cn (W.Z.)

² School of Aerospace Science and Technology, Xidian University, Xi'an 710071, China; minzhanghk@gmail.com

* Correspondence: wanghai@mail.xidian.edu.cn; Tel.: +86-029-8820-3115

Received: 28 January 2019; Accepted: 8 March 2019; Published: 13 March 2019



Abstract: An outside mutual correction (OMC) algorithm for natural scene text detection using multibox and semantic segmentation was developed. In the OMC algorithm, semantic segmentation and multibox were processed in parallel, and the text detection results were mutually corrected. The mutual correction process was divided into two steps: (1) The semantic segmentation results were employed in the bounding box enhancement module (BEM) to correct the multibox results. (2) The semantic bounding box module (SBM) was used to optimize the adhesion text boundary of the semantic segmentation results. Non-maximum suppression (NMS) was adopted to merge the SBM and BEM results. Our algorithm was evaluated on the ICDAR2013 and SVT datasets. The experimental results show that the developed algorithm had a maximum increase of 13.62% in the F-measure score and the highest F-measure score was 81.38%.

Keywords: scene text detection; multibox detector; semantic segmentation

1. Introduction

Scene text detection is an important and challenging task in computer vision [1,2]. The goal is to accurately locate the text area in a scene image. This technology has a wide range of applications in image retrieval, scene analysis, blind navigation, and other fields [3]. Text found in natural scenes have different fonts, styles, and sizes. It is usually accompanied by geometric distortion, a complex background, and uncontrolled lighting. Therefore, natural scene text detection is still a very open research challenge.

Early mainstream approaches focused on various heuristics that help detect characters or character components. The two most famous approaches are: (1) the maximum stable extremal region (MSER) [4–6]; and (2) the stroke width transformation (SWT) [7,8]. MSER extracts character regions with a similar intensity whereas SWT assumes that the text component has a comparable stroke width. Both MSER and SWT must be combined with additional post-processing to produce reasonable text candidates.

In recent works, various convolutional neural network (CNN)-based methods have been proposed to detect scene text [9–14]. These efforts focus on reducing the number of handcrafted features or artificial rules in text detection. Tian et al. [15] propose a text flow method using a minimum cost flow network to sort character CNN candidate detection, erroneous character deletion, text line extraction, and text line verification. Cho et al. [16] propose a canny text detector using the maximum stable region, edge similarity, text line tracking, and heuristic rule grouping to calculate candidate characters.

One of the most important CNN-based methods is a fully convolutional network (FCN). Zhang et al. [17] suggest using a FCN to obtain text block candidates, a character-centroid FCN to generate auxiliary text lines, and a set of heuristic rules based on intensity and geometric consistency to reject the wrong candidates. Gupta et al. [18] proposes a fully convolutional regression network (FCRN) that efficiently performs text detection and bounding-box regression at all locations across multiple scales in an image based on a FCN. All of these algorithms use FCN to generate text semantic segmentation detection results containing semantic segmentation information.

Another method called multibox uses multiple default candidate boxes to calculate the position of text in an image. Since general object detection based on CNN has achieved remarkable results in recent years, scene text detection has been greatly improved by treating text words or lines as objects. High-performance methods for object detection such as a faster region-based convolutional neural network (R-CNN) [19], single shot multibox detector (SSD) [20], and you only look once (YOLO) [21] have been modified to detect horizontal scene text [10,14,22,23] and have been greatly improved.

In this paper, the outside mutual correction (OMC) algorithm is proposed. Existing fusion approaches use semantic segmentation as a module for extracting features inside a multibox detector, this is referred to as inside feature extraction (IFE). During IFE processing, feature maps extracted by semantic segmentation are enlarged first and then reduced, which usually introduces noise and reduces the accuracy of detection. In our proposed algorithm, semantic segmentation and multibox are processed in parallel, and the text detection results are mutually corrected. Thus, the bounding box enhancement module (BEM) and the semantic bounding box module (SBM) were designed. The proposed algorithm inherits the advantages of these methods and obtained more accurate text detection results through OMC.

The rest of the paper is organized as follows: In Section 2, we provide a brief review of the related theories, including single shot multibox and the FCN. In Section 3, we describe the details of our proposed algorithm. In Section 4, we present the experimental results on benchmarks and comparisons to other scene text detection systems. In Section 5, we provide the conclusions of this paper.

2. Related Work

2.1. Multibox Text Detector

The multibox text detector extracts feature maps using convolutional layers. The detector draws multiple default bounding boxes on feature maps of different resolutions. After the convolution process, the targets in the original image will decrease in size. Since the default box has a fixed shape, the target will be captured at the right size. SSD is a representative of multibox text detectors. SSD has several advantages, including: (1) SSD discretizes the output space of bounding boxes into a set of default boxes over different aspect ratios and scales per a feature map location. (2) During the prediction procedure, the network generates scores for the presence of each object category in each default box and adjusts the box to better match the object shape. (3) Additionally, the network combines predictions from multiple feature maps with different resolutions to naturally handle objects of various sizes [20]. Figure 1 shows the SSD network architecture.

He et al. [24] proposed an improved SSD-based network named single shot text detector (SSTD). The inception structure and semantic segmentation are integrated in the SSD. The inception structure shows better performance for extraction convolution features. A new module named the text attention module is used to fuse semantic segmentation information and convolution features. The whole semantic segmentation fusing process has three steps: (1) The feature maps are enlarged and scored during semantic segmentation using a deconvolution process. (2) In order to fuse with the original feature map, the semantic segmentation results need to be reduced to smaller sizes through convolution processing. (3) Finally, the text attention module combines semantic segmentation information and convolution features. In this process, the feature map will be enlarged first and then reduced,

introducing noise. This problem is caused by the desire to combine semantic segmentation within the multibox. Thus, this method to integrate semantic segmentation has limited effect.

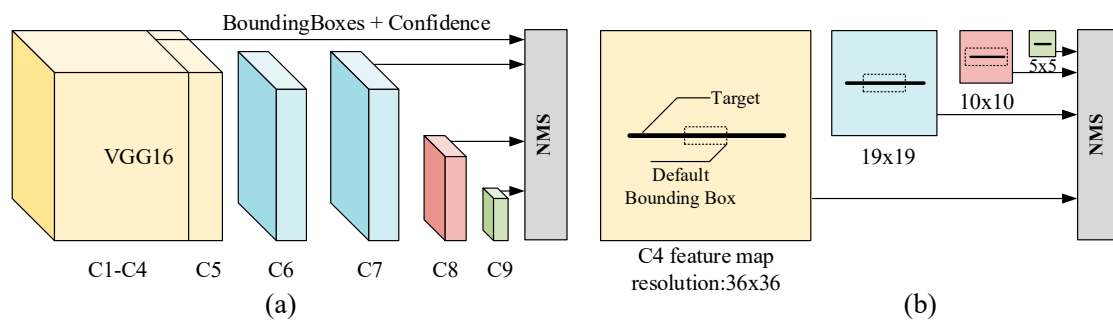


Figure 1. (a) Single shot multibox detector (SSD) architecture. From the convolutional layers C1 to C9, a number of bounding boxes and their corresponding confidence are inferred. (b) Fixed size default bounding box on different resolution feature maps. The fixed size default box captures different-sized targets in different resolution feature maps. This architecture allows the SSD to capture both large and small targets in one shot.

2.2. Semantic Segmentation

Semantic segmentation information can be acquired from the fully convolutional network (FCN) [25]. Typically, the convolutional neural network has several fully connected layers after the convolution layers. The feature maps generated from convolution layers are mapped to feature vectors by fully connected layers. For example, the 1000-dimensional vector output by the ImageNet model in [26] indicated the probability that the input image belongs to each class. The FCN classifies the object's class at the pixel level. The FCN uses the deconvolution layer to upsample the feature map and restore it to the input image size. After the deconvolution process, a prediction is made for each pixel classification and the spatial information in the input image is preserved. The FCN fused features from different coarseness layers to refine the segmentation using spatial information. Finally, the loss of the softmax classification was calculated pixel by pixel, which was equivalent to one training sample per pixel. The FCN architecture is shown in Figure 2.

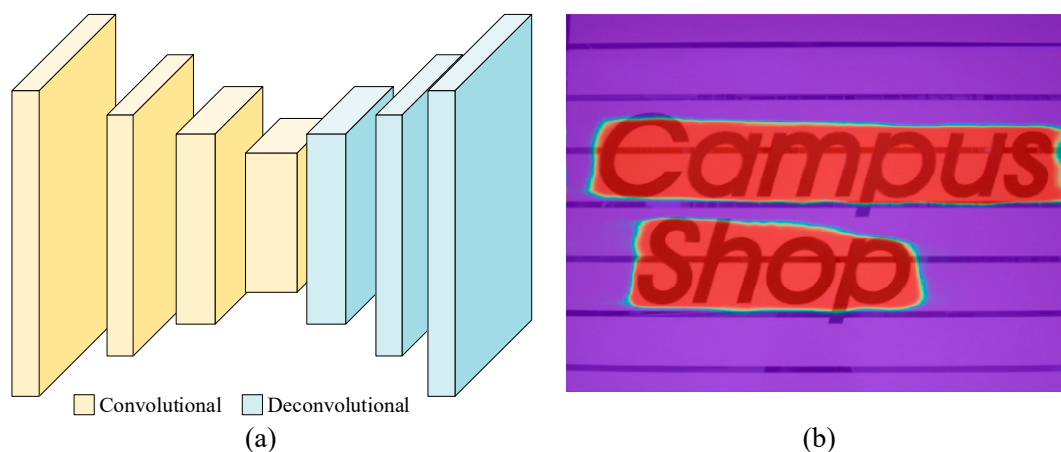


Figure 2. (a) Fully convolutional network (FCN) architecture. Including convolutional and deconvolution layers. (b) Overlay of the input image and FCN processing results.

Based on the FCN, DeepLab-V2 [27], containing atrous spatial pyramid pooling (ASPP), was proposed. In ASPP, parallel atrous convolution with different rates were applied in the input feature map and fused together. As objects of the same class may have different scales in the image, ASPP helps to account for different object scales that can improve the accuracy.

Usually semantic segmentation is treated as a special feature extraction process integrated into the convolutional neural network. The feature map will first be enlarged and then reduced during the inside feature extraction process as shown in Figure 3. During this process, noise is introduced and the detection results are affected. Therefore, the proposed algorithm uses the outside mutual correction (OMC) algorithm to fuse semantic segmentation outside the multibox. Through the proposed algorithm, the pixel-level classification of semantic segmentation can be more effectively utilized to improve the detection accuracy. We used SSD and SSTD as the basic multibox text detectors. DeepLab-V2 was used to generate semantic segmentation information.

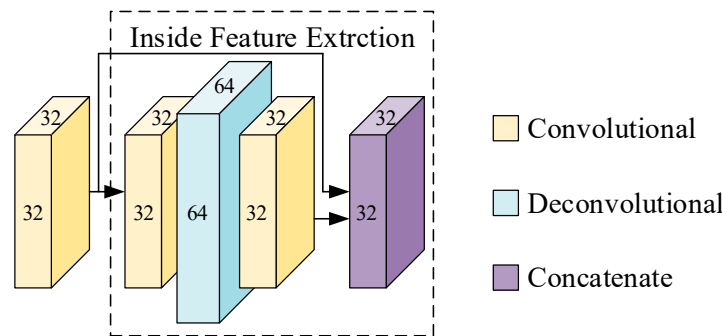


Figure 3. Inside feature extraction (IFE) process. The feature map will be enlarged first (32×32 to 64×64) and then reduced (64×64 to 32×32).

3. Proposed Algorithm

3.1. Overall Framework

The proposed algorithm is shown in Figure 4. Plenty of text candidate bounding boxes were obtained from the multibox processing. Meanwhile, the text semantic segmentation result was obtained from semantic segmentation processing. A softmax layer was added to the output layer of the semantic segmentation process to obtain the classification probability of each pixel. The text candidate bounding boxes and semantic segmentation results are merged in the bounding box enhancement module (BEM) to eliminate the false results of multibox processing. The semantic segmentation result enter the semantic bounding box module (SBM). After the CRF algorithm optimizes the text boundaries, the text semantic bounding box is computed. Finally, the outputs of the BEM and SBM are sent to the non-maximum suppression (NMS) to remove duplicate bounding boxes.

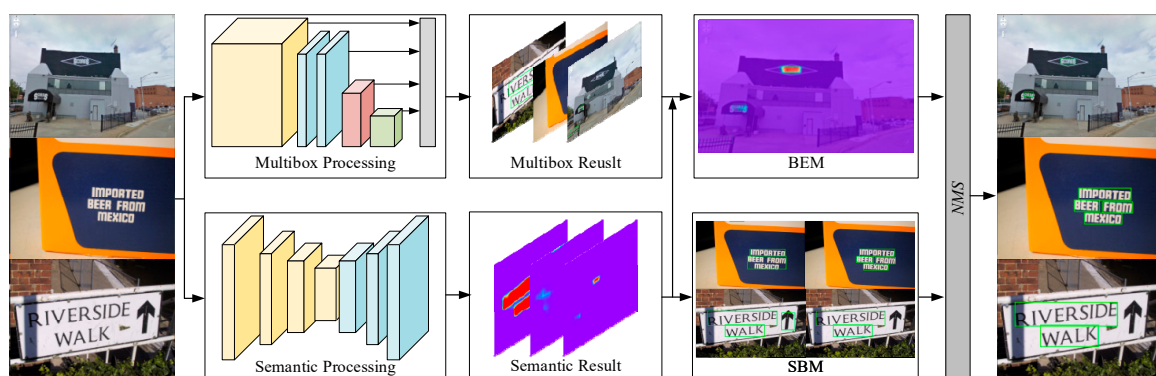


Figure 4. Framework of scene text detection algorithm using multibox and semantic segmentation.

3.2. Bounding Box Enhancement Module

The BEM merges the multibox results and semantic segmentation result to eliminate the false bounding boxes. The regional median probability of all bounding boxes was calculated based on the

semantic segmentation result. The bounding box will be removed if its regional median probability is less than the threshold. The detailed steps of the BEM are shown in Algorithm 1.

Algorithm 1. Bounding box enhancement module (BEM)

- Step 1. Acquire a multibox result: $Rec_i = ((x_1, y_1), (x_2, y_2))_i$. i refers to the i -th result. (x_1, y_1) is the coordinates of the upper left corner of the text bounding box. (x_2, y_2) is the coordinates of the right bottom corner of the text bounding box.
- Step 2. Get the rectangular area $Area_{Rec}$ of Rec_i in the semantic segmentation result.
- Step 3. Calculate the regional median probability:

$$P_{Area} = Median(\sum_{ij \text{ in Area}} P_{ij})$$
- Step 4. Compare P_{Area} to the threshold T :
If $P_{Area} < T$: Delete the Rec_i in the multibox results.
Else: continue.
- Step 5. Repeat steps 1–4 until all multibox results have been calculated.

The BEM process is shown in Figure 5. It can be seen from the figure that region (a) is the text and the region (b) is the background. These two blocks are detected as text in multibox processing. From the heatmap of the semantic processing results, we found that region (a) was bright while region (b) was dark. After the processing of the bounding box enhancement module, region (a) was reserved and region (b) was discarded.

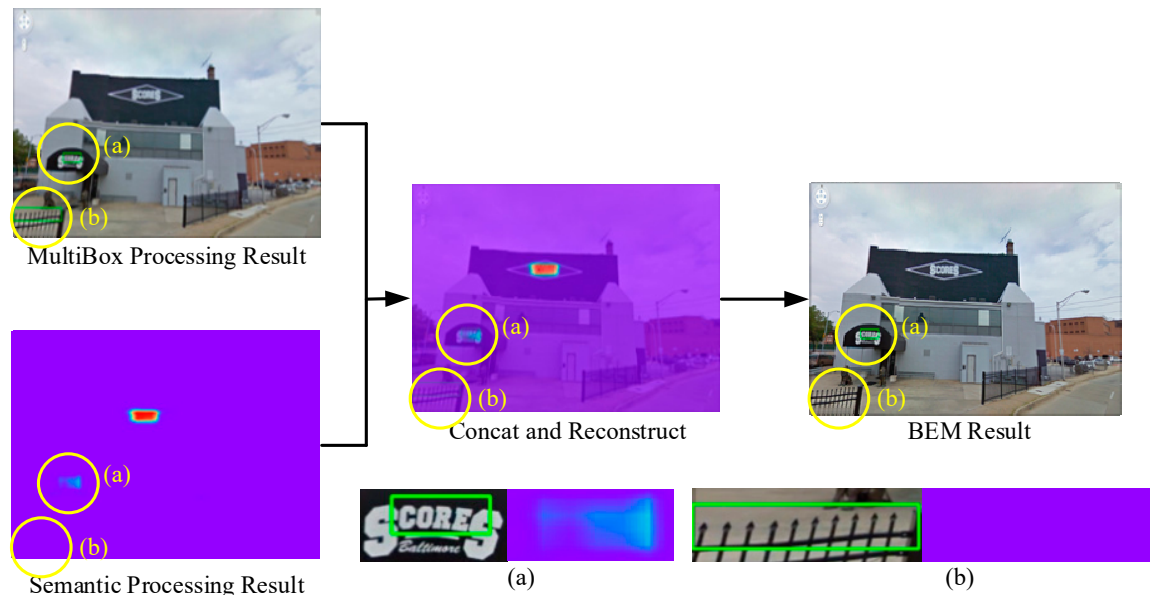


Figure 5. Flow chart of the BEM.

3.3. Semantic Bounding Box Module

The SBM contains two modules: CRF processing and bounding box search. The CRF processing solves the problem of boundary blur and stickiness in text semantic segmentation. The bounding box search obtains the optimal text semantic segmentation bounding box according to the CRF processing result. The SBM process is shown in Figure 6.

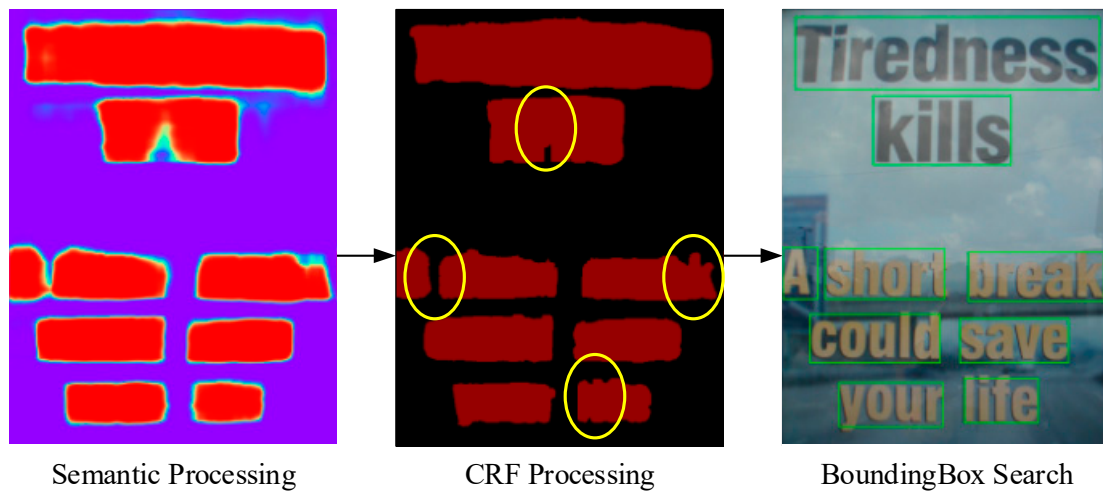


Figure 6. Semantic bounding box module flow chart. The yellow circle areas in the figure are clearer in the text boundary portion after the CRF processing.

The semantic segmentation result has a defect, which is that it is easy to cause adhesion when words are close. The CRF algorithm is used to correct the pixel-level prediction of the semantic segmentation result. Text edges are sharper and less sticky after CRF processing. CRF has been employed to smooth noisy segmentation maps [28,29]. These methods use short-range CRF to couple neighboring nodes, favoring same-label assignments to spatially proximal pixels. In this work, the goal should be to recover the detailed local structure rather than smooth it further. Therefore, the fully connected CRF model [30] is integrated into our network. The CRF model employs the energy function:

$$E(x) = \sum_i \theta_i(x_i) + \sum_{ij} \theta_{ij}(x_i, x_j), \tag{1}$$

where the x is the label assignment for pixels and the unary potential is defined as follows:

$$\theta_i(x_i) = -\log P(x_i), \tag{2}$$

where $P(x_i)$ is the label assignment probability of pixel i computed by semantic processing.

The pairwise potential has a form that allows for efficient inference while using a fully-connected graph, i.e., connecting all pairs of image pixels, i, j . In particular, as in [30], the following expression is used:

$$\theta_{ij}(x_i, x_j) = \mu(x_i, x_j) \left[\omega_1 \exp\left(-\frac{\|p_i - p_j\|^2}{2\sigma_\alpha^2} - \frac{\|I_i - I_j\|^2}{2\sigma_\beta^2}\right) + \omega_2 \exp\left(-\frac{\|p_i - p_j\|^2}{2\sigma_\gamma^2}\right) \right], \tag{3}$$

where $\mu(x_i, x_j) = 1$ if $x_i \neq x_j$, and zero otherwise. The remaining expression uses two Gaussian kernels in different feature spaces; the first, bilateral kernel depends on both pixel positions p and RGB color I , while the second kernel depends on pixel positions. The hyperparameters σ_α , σ_β and σ_γ control the scale of Gaussian kernels. The first kernel has similar tags for pixels with similar colors and positions, whereas the second kernel considers spatial proximity while smoothing the boundaries. Figure 7 shows the effectiveness of the CRF processing.



Figure 7. Examples of CRF processing effects. (a) Two lines of text are placed in one bounding box. After CRF processing, each word has a separate bounding box. (b) The arrow is erroneously detected as text and the arrow bounding box is removed after CRF processing.

4. Experimental Results

4.1. Datasets

- ICDAR2013

The ICDAR 2013 [31] consists of 229 training images and 233 testing images, with word-level annotations provided. It is the standard benchmark for evaluating near-horizontal text detection. Some examples of the ICDAR2013 dataset are shown in Figure 8.

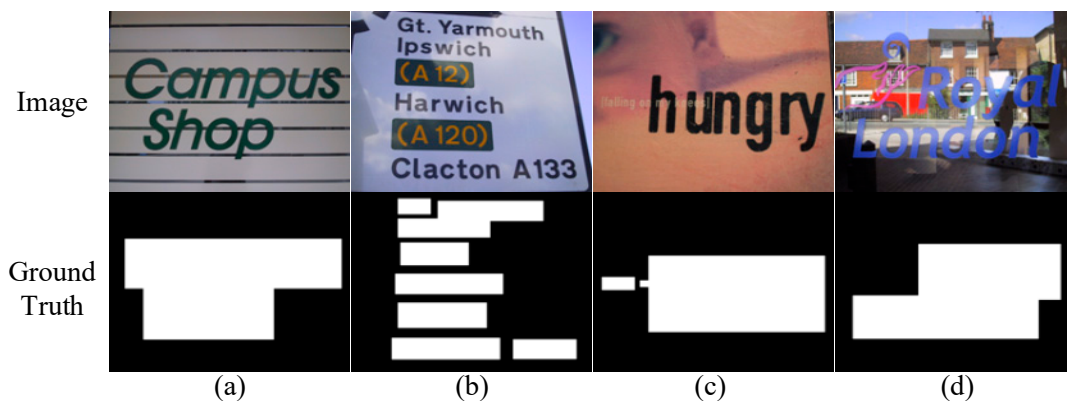


Figure 8. Several examples of difficulties in text detection in the ICDAR2013 dataset. (a) Background destruction font structure. (b) Perspective transformation and uneven illumination. (c) Text is too small and low contrast. (d) Complex background and low contrast.

- Street View Text (SVT)

The SVT dataset is harvested from Google Street View. Image text in this data exhibits high variability and often has low resolution [32]. In autonomous driving, the text in these Street View images helps the system confirm its position. Compared to the ICDAR2013 dataset, text in the SVT dataset has lower resolution, more complex light conversion, and relatively smaller text, which is more challenging. Figure 9 shows some examples of the SVT dataset.

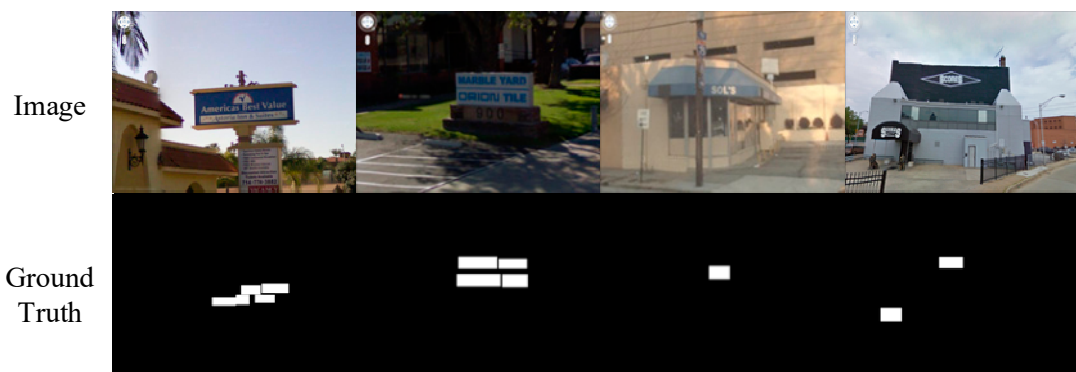


Figure 9. Examples of the Street View Text (SVT) dataset. Text with lower resolution, complex light conversion, and smaller text.

4.2. Exploration Study

Inside feature extraction (IFE) and outside mutual correction (OMC) algorithms were tested. Since the IFE in SSTD cannot be removed alone, the performance without IFE cannot be tested. Therefore, only IFE and OMC algorithms were tested. The training and testing environments were consistent. All methods used the same training datasets, the same number of training epochs, and the same set parameters. We used two standard evaluation protocols: the IC13 standard and the DetEval standard [33]. The proposed method was implemented with Caffe and Matlab, running on a computer with an 8-core CPU, 32G RAM, TitanXP GPU, and Ubuntu 16.04. The complete test results are shown in Tables 1 and 2.

Table 1. Comparison of the IFE and outside mutual correction (OMC) algorithms (on ICDAR2013).

Network	IC13 Standard			DetEval Standard		
	R ¹	P ²	F ³	R	P	F
SSD	52.20%	87.76%	65.18%	53.06%	87.24%	65.98%
SSD-IFE	49.17%	82.79%	61.69%	49.88%	83.29%	62.39%
SSD-OMC	69.30%	81.21%	74.78%	70.15%	85.10%	76.91%
SSTD-IFE	74.54%	83.65%	78.83%	75.39%	84.07%	79.50%
SSTD-OMC	80.51%	82.27%	81.38%	80.43%	87.60%	83.86%

¹ Recall. ² Precision. ³ F-measure.

Table 2. Comparison of IFE and OMC algorithms (on SVT).

Network	IC13 Standard			DetEval Standard		
	R	P	F	R	P	F
SSD	48.61%	73.24%	58.43%	48.17%	75.12%	58.70%
SSD-IFE	50.34%	79.91%	61.77%	50.34%	79.91%	61.77%
SSD-OMC	69.91%	73.57%	71.69%	66.01%	77.97%	71.48%
SSTD-IFE	78.60%	73.35%	75.89%	78.60%	73.35%	75.89%
SSTD-OMC	85.13%	71.68%	77.83%	81.65%	79.78%	80.71%

‘SSD’ refers to the original SSD algorithm without IFE and OMC. ‘SSD-IFE’ refers to the SSD added IFE algorithm. ‘SSD-OMC’ refers to the SSD added OMC algorithm. ‘SSTD-IFE’ refers to the SSTD added IFE algorithm, which was the original SSTD. ‘SSTD-OMC’ refers to the SSTD added OMC algorithm.

The experimental results show that the IFE algorithm reduces the accuracy of text detection on the ICDAR2013 dataset. For the SVT dataset, the IFE algorithm slightly improved the accuracy of the text detection. Compared to the IFE algorithm, the OMC algorithm significantly improved the F-measure score.

4.3. Experimental Results

Five methods were tested on the ICDAR2013 and SVT datasets: FCN, SSD, SSD-OMC, SSTD, and SSTD-OMC. The SSD-OMC and the SSTD-OMC use the proposed algorithm to combine the semantic segmentation with SSD and SSTD, respectively.

Table 3 shows the tested results of five methods on the ICDAR2013 dataset using two standard evaluation protocols. It can be seen from the test results that the SSD-OMC and SSTD-OMC algorithms showed an increase of 17.10% (IC13), 17.09% (DetEval) and 5.97% (IC13), 5.04% (DetEval) in the recall rate relative to the SSD and SSTD algorithms, which means that more texts missed by the multibox processing was detected. The SSD-OMC method was 9.60% (IC13) and 10.93% (DetEval) higher than the SSD method in the F-measure score, and the SSTD-OMC method was 2.55% (IC13) and 4.36% (DetEval) higher compared to the SSTD method in the F-measure score, meaning that the multibox processing optimized by our algorithm had a better detection accuracy.

Table 3. Improved algorithm compared with original algorithm (on ICDAR2013).

Network	IC13 Standard			DetEval Standard		
	R ¹	P ²	F ³	R	P	F
FCN	62.54%	60.80%	61.66%	66.97	62.05%	64.42%
SSD	52.20%	87.76%	65.18%	53.06%	87.24%	65.98%
SSD-OMC	69.30%	81.21%	74.78%	70.15%	85.10%	76.91%
SSTD	74.54%	83.65%	78.83%	75.39%	84.07%	79.50%
SSTD-OMC	80.51%	82.27%	81.38%	80.43%	87.60%	83.86%

¹ Recall. ² Precision. ³ F-measure.

Table 4 shows the results of the SSTD-OMC algorithm and four other advanced text detection algorithms tested on the ICDAR2013 dataset. As can be seen in the table, SSTD-OMC has a higher f-measure score, indicating that SSTD-OMC had the better detection accuracy among these five algorithms. Some detection results are shown in Figure 10.

Table 4. Improved algorithm compared with other advanced algorithms (on ICDAR2013).

Network	IC13 Standard			DetEval Standard		
	R	P	F	R	P	F
Yin [34]	0.66	0.88	0.76	0.69	0.89	0.78
Neumann [35]	0.72	0.82	0.77	-	-	-
Zhang [36]	0.74	0.88	0.80	0.76	0.88	0.82
Textboxes [22]	0.74	0.86	0.80	0.74	0.88	0.81
SSTD-OMC	0.80	0.82	0.81	0.80	0.80	0.83

Table 5 shows the results of five methods tested on the SVT dataset. The SSD-OMC method showed an improvement of 13.62% (IC13) and 12.78% (DetEval) on the F-measure score compared to the SSD method, while the SSTD-OMC method improved 1.94% (IC13) and 4.82% (DetEval) on the F-measure score compared to the SSTD method. Figure 11 shows some detection results from the SVT dataset.



Figure 10. Some detection results and their ground truth from the ICDAR2013 dataset.

Table 5. Improved algorithm compared with original algorithm (on SVT).

Network	IC13 Standard			DetEval Standard		
	R	P	F	R	P	F
FCN	50.78%	54.94%	52.78%	55.13%	54.94%	55.03%
SSD	48.61%	73.24%	58.43%	48.17%	75.12%	58.70%
SSD-OMC	69.91%	73.57%	71.69%	66.01%	77.97%	71.48%
SSTD	78.60%	73.35%	75.89%	78.60%	73.35%	75.89%
SSTD-OMC	85.13%	71.68%	77.83%	81.65%	79.78%	80.71%



Figure 11. Some detection results and their ground truth from the SVT dataset.

5. Conclusions

Our work provided an OMC algorithm to fuse multibox with semantic segmentation. In the OMC algorithm, semantic segmentation and multibox were processed in parallel, and the text detection results were mutually corrected. The mutual correction process had two stages. In the first stage, the pixel-level classification results of the semantic segmentation were adopted to correct the multibox

bounding boxes. In the second stage, the CRF algorithm was used to precisely adjust the boundaries of the semantic segmentation results. Then the NMS was introduced to merge the text bounding boxes generated by multibox and semantic segmentation. The experimental results showed that the proposed OMC algorithm had better performance than the original IFE algorithm. The F-measure score increased by a maximum of 13.62% and the highest F-measure score was 81.38%. Future work will focus on more powerful and faster detection structures, as well as on rotating text detection research.

Author Contributions: Conceptualization: H.Q. and H.Z.; formal analysis: H.W. and Y.Y.; investigation: H.Q. and H.Z.; methodology: H.Q. and M.Z.; Writing—Original Draft: H.W. and W.Z.; Writing—Review & Editing: H.Q. and H.Z.

Funding: This research was funded by National Natural Science Foundation of China, grant number 61801357.

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

References

1. Fletcher, L.A.; Kasturi, R. A robust algorithm for text string separation from mixed text/graphics images. *IEEE Trans. Pattern Anal. Mach. Intell.* **1988**, *10*, 910–918. [[CrossRef](#)]
2. Ye, Q.; Doermann, D. Text detection and recognition in imagery: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1480–1500. [[CrossRef](#)] [[PubMed](#)]
3. Wang, F.; Zhao, L.; Li, X.; Wang, X.; Tao, D. Geometry-aware scene text detection with instance transformation network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; IEEE: Piscataway, NJ, USA, 2018.
4. Chen, H.; Tsai, S.S.; Schroth, G.; Chen, D.M.; Grzeszczuk, R.; Girod, B. Robust text detection in natural images with edge-enhanced maximally stable extremal regions. In Proceedings of the 18th IEEE International Conference on Image Processing (ICIP), Brussels, Belgium, 11–14 September 2011; IEEE: Piscataway, NJ, USA, 2011.
5. Neumann, L.; Matas, J. Real-time scene text localization and recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, 16–21 June 2012; IEEE: Piscataway, NJ, USA, 2012.
6. Shi, C.; Wang, C.; Xiao, B.; Zhang, Y.; Gao, S. Scene text detection using graph model built upon maximally stable extremal regions. *Pattern Recognit. Lett.* **2013**, *34*, 107–116. [[CrossRef](#)]
7. Epshtein, B.; Ofek, E.; Wexler, Y. Detecting text in natural scenes with stroke width transform. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), San Francisco, CA, USA, 13–18 June 2010; IEEE: Piscataway, NJ, USA, 2010.
8. Mosleh, A.; Bouguila, N.; Hamza, A.B. Image text detection using a bandlet-based edge detector and stroke width transform. In *BMVC*; *BMVC*: Newcastle, UK, 2012.
9. He, W.; Zhang, X.Y.; Yin, F.; Liu, C.L. Deep direct regression for multi-oriented scene text detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; IEEE: Piscataway, NJ, USA, 2017.
10. Liu, Y.; Jin, L. Deep matching prior network: Toward tighter multi-oriented text detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; IEEE: Piscataway, NJ, USA, 2017.
11. Ma, J.; Shao, W.; Ye, H.; Wang, L.; Wang, H.; Zheng, Y.; Xue, X. Arbitrary-oriented scene text detection via rotation proposals. *IEEE Trans. Multimedia* **2018**, *20*, 3111–3122. [[CrossRef](#)]
12. Xiang, D.; Guo, Q.; Xia, Y. Robust text detection with vertically-regressed proposal network. In *European Conference on Computer Vision*; Springer: Berlin, Germany, 2016.
13. 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, Honolulu, HI, USA, 21–26 July 2017; IEEE: Piscataway, NJ, USA, 2017.
14. Tian, Z.; Huang, W.; He, T.; He, P.; Qiao, Y. Detecting text in natural image with connectionist text proposal network. In *European Conference on Computer Vision*; Springer: Berlin, Germany, 2016.

15. Tian, S.; Pan, Y.; Huang, C.; Lu, S.; Yu, K.; Lim Tan, C. Text flow: A unified text detection system in natural scene images. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; IEEE: Piscataway, NJ, USA, 2015.
16. Cho, H.; Sung, M.; Jun, B. Canny text detector: Fast and robust scene text localization algorithm. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; IEEE: Piscataway, NJ, USA, 2016.
17. 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, Las Vegas, NV, USA, 27–30 June 2016; IEEE: Piscataway, NJ, USA, 2016.
18. 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; IEEE: Piscataway, NJ, USA, 2016.
19. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [[CrossRef](#)] [[PubMed](#)]
20. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single Shot Multibox Detector. In *European Conference on Computer Vision*; Springer: Berlin, Germany, 2016.
21. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; IEEE: Piscataway, NJ, USA, 2016.
22. Liao, M.; Shi, B.; Bai, X.; Wang, X.; Liu, W. *TextBoxes: A Fast Text Detector with a Single Deep Neural Network*; AAAI: Menlo Park, CA, USA, 2017.
23. 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, Honolulu, HI, USA, 21–26 July 2017; IEEE: Piscataway, NJ, USA, 2017.
24. He, P.; Huang, W.; He, T.; Zhu, Q.; Qiao, Y.; Li, X. Single shot text detector with regional attention. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; IEEE: Piscataway, NJ, USA, 2017.
25. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*; IEEE: Piscataway, NJ, USA, 2015.
26. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *ACM* **2012**, *60*, 84–90. [[CrossRef](#)]
27. Chen, L.-C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 834–848. [[CrossRef](#)] [[PubMed](#)]
28. Rother, C.; Kolmogorov, V.; Blake, A. Grabcut: Interactive foreground extraction using iterated graph cuts. In *ACM Transactions on Graphics (TOG)*; ACM: New York, NY, USA, 2004.
29. Kohli, P.; Torr, P.H. Robust higher order potentials for enforcing label consistency. *Int. J. Comput. Vis.* **2009**, *82*, 302–324. [[CrossRef](#)]
30. Krähenbühl, P.; Koltun, V. Efficient inference in fully connected crfs with gaussian edge potentials. *Adv. Neural Inf. Process. Syst.* **2011**, *24*, 109–117.
31. Karatzas, D.; Shafait, F.; Uchida, S.; Iwamura, M.; Bigorda, L.G.; Mestre, S.R.; Mas, J.; Mota, D.F.; Almazan, J.A.; de Las Heras, L.P. ICDAR 2013 robust reading competition. In Proceedings of the 12th International Conference on Document Analysis and Recognition (ICDAR), Washington, DC, USA, 25–28 August 2013; IEEE: Piscataway, NJ, USA, 2013.
32. Wang, K.; Belongie, S. Word spotting in the wild. In *European Conference on Computer Vision*; Springer: Berlin, Germany, 2010.
33. Lucas, S.M.; Panaretos, A.; Sosa, L.; Tang, A.; Wong, S.; Young, R.; Ashida, K.; Nagai, H.; Okamoto, M.; Yamamoto, H. ICDAR 2003 robust reading competitions: entries, results, and future directions. *Int. J. Doc. Anal. Recognit.* **2005**, *7*, 105–122. [[CrossRef](#)]
34. Yin, X.-C.; Yin, X.; Huang, K.; Hao, H.W. Robust text detection in natural scene images. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *36*, 970–983. [[PubMed](#)]

35. Neumann, L.; Matas, J. Efficient scene text localization and recognition with local character refinement. In Proceedings of the 13th International Conference on Document Analysis and Recognition (ICDAR), Tunis, Tunisia, 23–26 August 2015; IEEE: Piscataway, NJ, USA, 2015.
36. Zhang, Z.; Shen, W.; Yao, C.; Bai, X. Symmetry-based text line detection in natural scenes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; IEEE: Piscataway, NJ, USA, 2015.



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).