6. Appendix

6.1. Details of Sampling

To learn a reliable arbitrary-shape text contour generation network, we not only employ the positive training samples, but also utilize the negative training samples. As shown in Fig. 5, we take the minimum bounding box (red) of the text as the positive training sample. To generate the negative training sample, we first place the contour of each arbitrary-shape scene text on the image with the stride of 8 to generate candidate contours (cyan dashed line) in a copy-move manner. Then, we calculate the overlaps between the axis-aligned circumscribed bounding boxes of these candidate contours and those of all positive contours (cyan solid line). After that, the candidate contours are assigned to different bins {(0, 0.05], (0.05, 0.1], (0.1, 0.15], (0.15, 0.2], (0.2, 0.25], (0.25, 0.3]} based on the overlaps. Next, we randomly choose a contour from the bin of the lowest interval that contains candidate contours. Finally, the minimum bounding box (yellow) of the selected contour is regarded as the negative sample. Additionally, the generation of the minimum bounding box follows the strategy that utilizes $cv2.minAreaRect$ and $cv2.boxPoints$ to obtain corner points. We further reorder the generated corner points from the top-left point anticlockwise, as the Arabic number labeled in Fig. 5.

6.2. More Qualitative Detection Results

To further demonstrate the effectiveness of our proposed method, more qualitative detection results are displayed in Fig. 6. We observe that our model can not only localize the arbitrary-shape text in complex scenarios, but also detect texts with extremely small sizes. Additionally, our method can also localize long Chinese or English texts well via the progressive regression strategy.

6.3. Intermediate Results

As shown in Fig. 7, we report the quantitative results of intermediate outputs and visualize the qualitative results of several samples. In Fig. 7 (c), the axis-aligned bounding boxes generated by the horizontal text proposal generation module can not surround the oriented or curved texts well, thus resulting in poor performances, *e.g.*, *Recall* of 63.1%, *Precision* of 70.1%, and *F-measure* of 66.7%. Meanwhile, the oriented bounding boxes yielded by the oriented text proposal generation module also can not enclose the curved texts well, as shown in Fig. 7 (d), but it promotes the *F-measure* from 66.7% to 72.2%. After we evolve the oriented text proposals to arbitrary-shape text contours for one time, the *F-measure* increases by 8.4%. It indicates that the localized text contours are much better than the oriented bounding boxes, as shown in Fig. 7 (e). When we further evolve the localized text contours, the text contours are closer to the ground-truth, as illustrated in Fig. 7 (f). It boosts 2.7%, 2.1%, and 2.5% in *Recall*, *Precision*, and *F-measure*, respectively. Moreover, Fig. 7 (g) reveals that the reliable contour localization mechanism can achieve more accurate text contours with the high confidence, which especially brings an improvement of 1.1% in *Precision*.

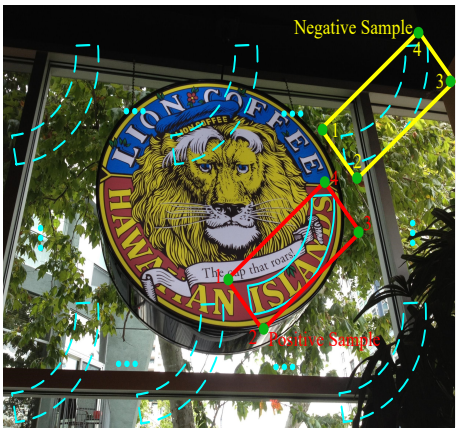


Figure 5: Illustration of sampling. The cyan dashed line denotes the negative candidate contours generated by the contour (cyan solid line) of the scene text in a copy-move manner. The red and yellow minimum bounding boxes of contours indicate the positive training sample and the negative training sample respectively. The labeled Arabic numbers represent the orders of reordered corner points.

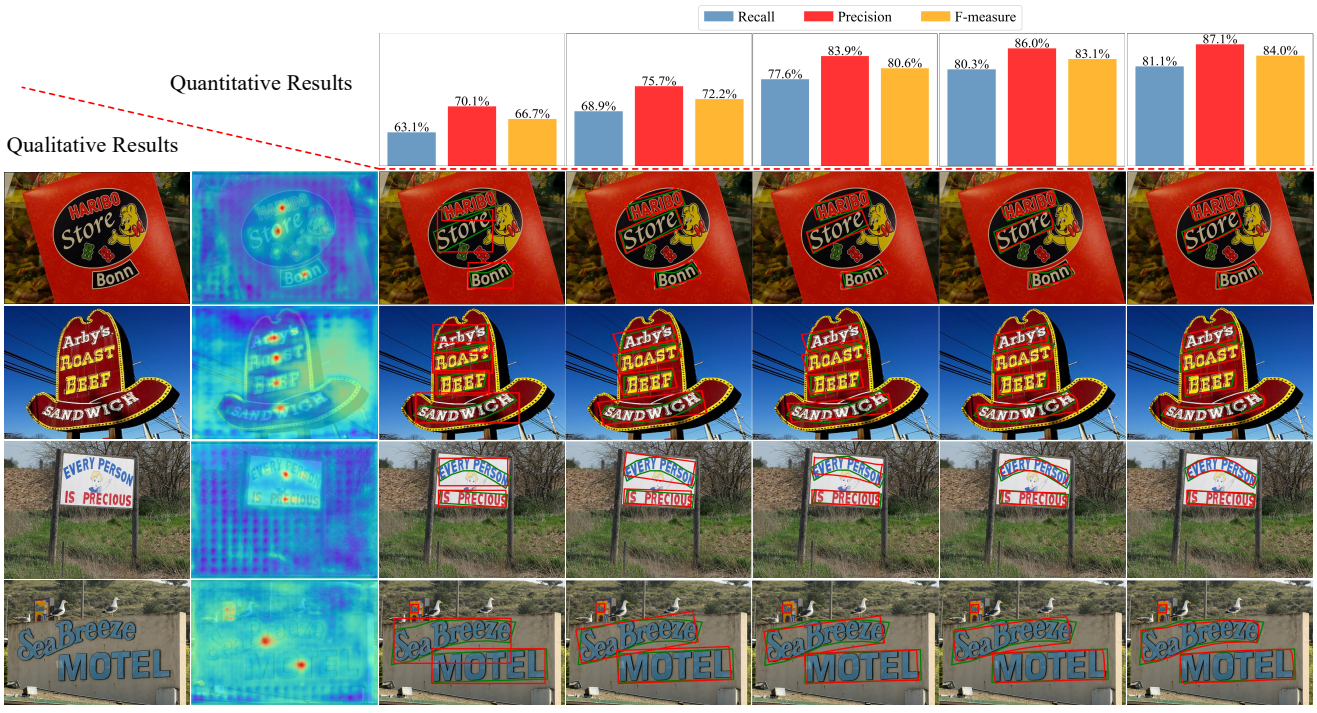
6.4. Qualitative Analyses of Ablation Study

To further illustrate the effectiveness of the oriented text proposals generation (OTPG) module, the contour information aggregation (CIA) technique, and the reliable contour localization mechanism (RCLM), we present the qualitative detection results in Fig. 8. When our proposed model does not utilize CIA, the localized text contours are not smooth enough and surround more backgrounds, as displayed in Fig. 8 (c). When we do not integrate CRLM, Fig. 8 (d) reveals that it localizes the curved text well, but generates some false and missing detections. Once the OTPG module is not employed, as shown in Fig. 8 (f), it would generate some self-intersection text contours, compared with those in Fig. 8 (e).

6.5. Runtimes Analyses

In the inference stage, the time consumption of our proposed method mainly consists of two components: the network inference time and the post-processing time. According to Table 8, the testing scale can distinctly influence the network inference time. In effect, we have fixed the shorter side of the testing image to 416, 512, 640, and 640 for CTW1500, Total-Text, ArT, and TD500, respectively, while resized the longer side to keep the original aspect ratio. Ad-

Figure 6: Qualitative detection results of our method on CTW1500, Total-Text, ArT, and TD500. Red bounding boxes denote detection results. Green bounding boxes are the ground-truth. It is worth noting that the ground-truth of ArT is not available.



(a) Input (b) Center Map (c) HTPG (d) OTPG (e) CLM-1 (f) CLM-2 (g) RCLM

Figure 7: Illustration of quantitative and qualitative results for intermediate outputs. (a) denotes the input image. (b) is the predicted center heatmap that is placed on the input image. (c) and (d) indicate the outputs of the horizontal text proposal genreation (HTPG) module and the oriented text propogal generation (OTPG) module. (e) and (f) mean the outputs of the first and second contour localization mechanism (CLM) module. (g) is the output of the reliable contour localization mechanism (RCLM). Red means the detected results. Green is the ground-truth. The experiments are conducted on CTW1500.

ditionally, the number of text contour detections, generated by the network, would mainly affect the speed of the polygonal NMS [5] in the post-processing. It is because the number of text contour detections could influence the calculation of the overlaps between text contours. To accelerate the polygonal NMS operation, we utilize *cython* and *c* lan-

(a) Input (b) Ours (c) w/o CIA (d) w/o CRLM (e) w/o CIA+CRLM (f) w/o CIA+CRLM+OTPG

Figure 8: Qualitative detection results in the ablation study. Red bounding boxes denote detection results. Green bounding boxes represent the ground-truth.

Table 8: Runtimes of our method. 'F' means the *F-measure*.

Dataset	F (%)	Testing Scale	Time Consumption (ms)		FPS
			Network Inference	Post Processing	
CTW1500	84.7	416	49.8	0.52	19.9
Total-Text	85.2	512	53.8	0.50	18.4
ArT	74.0	640	60.8	0.63	16.3
TD500	87.0	640	59.6	0.25	16.7

guage, instead of pure *python* used in the body. Compared with the speed reported in Table 2 of the body, the accelerated NMS can increase the total runtime of our model by about 8 FPS for CTW1500. The reported runtime in Table 8 is the average time per image over three runs for each dataset, based on a workstation with one 4.0 GHz CPU and a single NVIDIA GTX 2080Ti GPU.

6.6. Limitations

According to previous experiments, our proposed method works well in most challenging scenarios, but it still fails for some difficult cases. Firstly, when the characters of scene texts contain artistic fonts, these texts could not be detected or only a part of the texts are localized, as shown in Fig. 9 (a). The reason is that the feature representations of artistic characters are more similar to general objects. It is easy to be regarded as the non-text class, thus resulting in the inaccurate predictions of text centers and sizes. Secondly, Fig. 9 (b) reveals that our model is incapable of detecting the overlapped texts. It can be ascribed to two reasons: (i) the centers of overlapped texts may be the same, which only generates one prediction; (ii) the polygonal NMS in our

(a) Artistic fonts (b) Overlapped texts (c) Large character space

Figure 9: Failure samples. These samples come from CTW1500, Total-Text, and TD500. Red bounding boxes denote detection results. Green bounding boxes are the ground-truth.

model utilizes the maximum intersection [5] to calculate the overlap, which easily results in only keeping one text when the detected text is largely contained by the other. Thirdly, the large character space in the texts can lead these texts to be undetected, or some individual characters of the texts to be detected, as displayed in Fig. 9 (c). It is because the large character space may induce our model to generate uncertain text centers and sizes. Meanwhile, the space between characters may be regarded as the background to break off the entire text.