International Journal Of
Industrial Electronics Control and Optimization
Print ISSN: 3517-2645
Online ISSN: 3568-2645
Homepage: https://ieco.usb.ac.ir/

Multi-Oriented Scene Text Detection at the Character Level

Mahdi Kazeminia[1] | Hamed Shahraki [2] | Mehran Tamjidi [3]

Faculty of Engineering, Velayat University, Iranshahr, Iran.[1, 2, 3]

Corresponding author's email: M.Kazeminia@velayat.ac.ir

Article Info

Article type:
Research Article

Article history:
Received: 18-April-2023
Received in revised form: 14-Aug-2023
Accepted: 16-Aug-2023
Published online:20-Sep-2023

Keywords:
Deep Learning,
Scene Text Detection,
CRAFT.

ABSTRACT

Recent scene text detection methods perform superior on benchmark datasets using deep-learning frameworks. In this paper, we re-implement the state-of-the-art text detection method, character region awareness for text detection (CRAFT), which can detect individual characters of scene text images. CRAFT is a character-based detection method with many advantages in detecting complex text by detecting character units and estimating the area between characters, capable of detecting texts of any shape. We also improve the detection performance of the baseline method, CRAFT, by some modifications in its architecture and proposing a training scheme that takes benefit of the advanced optimizer. The performance improvements of CRAFT are validated on three benchmark datasets: ICDAR2013, ICDAR2015, and COCO-Text. By applying the pre-trained models on COCO-Text, CRAFT shows that it cannot generalize without fine-tuning. We also improve the ICDAR2015 model and evaluate it on benchmark datasets. The evaluation results show improved precision performance compared to the original pre-trained model with fewer iterations and higher accuracy.

I. Introduction

One of the most popular research topics in computer vision is reading text from wild images [1, 2, and 3], which is the base of numerous practical applications such as multi-language translation, image OCR, and image retrieval. Reading text from scene images is divided into two categories: (1) text detection, which aims at the localization of text in the image, and (2) text recognition, which aims at converting the localized text or cropped word image into a string. This paper focuses on the detection task, which is more challenging than recognition because of the complicated background and significant variance of text shape.

Before the deep learning era, the methods typically identified character or text component candidates using connected component-based or sliding window-based methods, which used hand-crafted features like MSER [4, 5] or SWT [6] as essential components. However, these methods have several significant drawbacks: (1) they are built only to detect individual characters or components, making it challenging for regional context information identification and leading to low recall performance. (2) These methods require multiple post-processing steps for detecting text. (3) They only work on the horizontal type of text and fail on the multi-oriented text.
Therefore, the performance of these classical machine learning-based methods is still far from satisfactory. Recent deep learning-based methods have proved that they can detect more challenging text in scene images. These methods usually adopt general frameworks of object detection methods such as SSD [7], YOLO [8], Faster R-CNN [9], or segmentation frameworks like FCN [10] and Mask R-CNN [11]. Most deep learning-based text detectors that detect text at the word level have difficulty finding curved, extremely long, or highly deformed words with a single bounding box. Character region awareness for text detection (CRAFT) [12] is a character-based detection method with many advantages in detecting complex text by detecting character units and estimating the area between characters, capable of detecting texts of any shape. Fig. 1 illustrates the CRAFT detection performance for

Fig. 1. The capability of the text detection results in any shape in CRAFT. (a) The polygon bounding box of the detected characters, (b) Gaussian heat-map of individual characters of the text.

different forms of texts. Most existing datasets do not provide character-level annotations and getting character ground truths (GTs) is too expensive. CRAFT localizes the individual character regions and concatenates them into text instances to tackle these problems. Moreover, CRAFT proposes a weakly-supervised learning framework for estimating character-based GTs from word-based annotation datasets to solve the problem relating to the lack of character-based annotations. CRAFT is a segmentation method that outputs a character region score localizing individual characters in an image and a character affinity score grouping characters into a single instance.

In this paper, we improve the detection performance of the baseline method, CRAFT [12], by some modifications in its architecture and proposing a training scheme that takes benefit of the advanced optimizer.

This article is organized as follows: The most significant text detectors are presented in Section II. In Section III, a comprehensive analysis is provided to understand the mechanism of the proposed method. To evaluate the proposed model, some tests are performed in Section IV, and finally, the conclusion is provided in Section V.

II. Background

New deep learning methods see scene text detection as an object detection problem in which words, characters, or text lines are treated as a detection target. For this reason, most text detection methods are based on advanced object detection methods categorized as follows:

A. Regression-Based Text Detectors

These types of methods [13, 14] adapt general Object Detection frameworks such as SSD [7] and Faster R-CNN [9] for text Detection; in these methods, the text is considered as an object, and candidate bounding boxes of text are predicted directly. For example, the single-shot descriptor (SSD) [7] is modified by using long default anchors and filters in TextBoxes [14]. Therefore, it handles the significant variation of aspect ratios of text instances to detect the various type of text shapes. Unlike TextBoxes, the Deep matching prior Network (DMPNet) presented in [15] introduces quadrilateral sliding windows to deal with multi-orientation text. Many regression-based methods [16, 17, and 18] tried to solve the detection challenges of rotated, curved, and arbitrary shape text. For instance, in a recent work, Li et al. [18] proposed a regression-based method that applies only NMS as post-processing to detect arbitrary shape text instances. However, due to structural limitations, it is not very easy to recognize the text of all possible shapes in this way. It would help if we could create anchors that cover all forms and determine the number of proposed boxes, but there is a trade-off between execution time and accuracy.

B. *Segmentation-based Text Detectors*

These methods [19, 20, and 21] classify text regions of images at the pixel level and provide word-level or character-level detection. They usually modified the segmentation framework like FCN [10] and Mask R-CNN [11]. Zhang et al. [22] deployed FCN to predict the salient map of text regions in an oriented scene text. An attention procedure is used in a single-shot text detector (SSTD) [19] to boost the text regions of an image and reduce the effect of background interference on the feature level. FCN is also deployed in TextSnake [21] as a base detector. The aim is to extract text instances by detecting and assembling local components. Liao et al. [23] proposed a simple segmentation network that utilizes differentiable binarization in recent work. It can real-time detect text in arbitrary shapes.

C. *End-to-End Text Detectors*

In these methods [24, 25] the detection and recognition mechanisms are trained simultaneously. End-to-end text detector methods improve the performance through the recognition module. FOTS [24] proposed the RoIRotate to effectively detect horizontal and multi-oriented text of an image by sharing convolutional features between detection and recognition modules. In [25], a single-shot text spotter is presented to make a direct mapping between an input image and character sequences by learning. The performance of detection is high in end-to-end methods because they deploy the recognition results to improve the text detection performance. In [26], an end-to-end segmentation-based

method is presented to predict a character-level probability map. It uses the Mask R-CNN architecture [11] for detecting and recognizing text instances in arbitrary shapes.

The previously mentioned methods aim to detect words in the input image. However, it is challenging to use a word as the basic unit of detection, leading to some problems. These problems are: (1) It is difficult to cope with arbitrary shape text: In previous methods, the training process was performed with rigid word-level bounding boxes, and limitations are exhibited to represent the text region in an arbitrary shape [27], (2) Defining the word is not easy: Although, in most methods, the text is detected with words as its unit, determining the extent of a word is essential for detection. Because words are distinguished by different bases, such as spaces or color, some letters do not have spaces, (3) it is evident that character is the basic unit in handling various text types: scene text images are organized hierarchically for visual components, such as characters, text lines, and words. As most languages are established based on characters, performing text detection methods on character detectors seems reasonable [28].

D. *Character-Level Text Detectors*

Zhang et al. [29] utilized MSER [4] to detect text at the character level. However, this method performs poorly on the low contrast images and curve shape texts. In [30], text maps obtained by segmentation use character-level annotations to generate multi-oriented text bounding boxes. Seglink [31] searches for small text elements (segments) in the image and links these segments to create word boxes with additional post-processing. CRAFT improved the idea of WordSup [28] by using a weakly supervised framework to detect individual characters in arbitrary shape text, enabling it to achieve state-of-the-art performance on public benchmark datasets.

III. Methodology

A. *Architecture*

Fig. 2 shows the network architecture of CRAFT. The overall architecture is similar to U-Net [32], a standard segmentation model based on FCN [10]. It adopts VGG-16 [33] as a backbone in the encoder and generates a Region Score/affinity score map by applying up-sampling and skip-connection of the U-Net structure. In simple words, VGG16 is essentially the feature-extracting architecture that is used to encode the network's input into a certain feature representation. The decoding segment of the CRAFT has to skip connections that aggregate low-level features. It predicts two scores for each character:

Region Score: As the name suggests, it gives the region of the character. It localizes the character.

Affinity Score: 'Affinity' is the degree to which a substance tends to combine with another. So, an affinity score merges characters into a single instance (a word). CRAFT generates

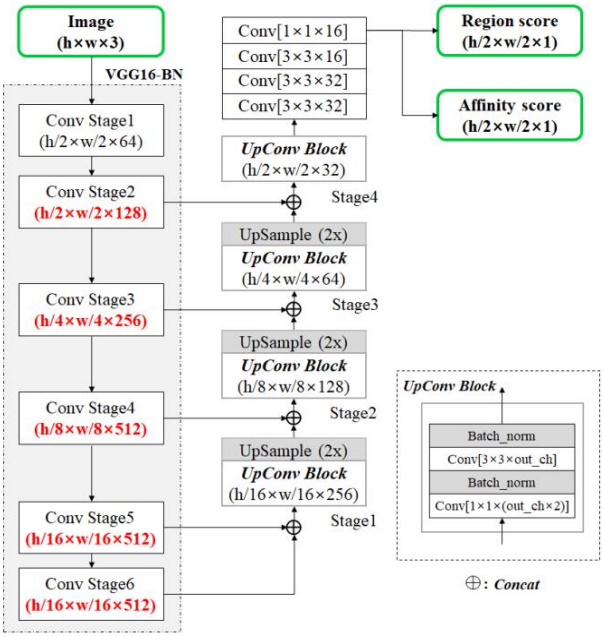

Fig. 2. The network architecture of CRAFT [1]. The size of the feature layers has been miscalculated, which is corrected in this figure with red.

two maps as output: Region Level Map and Affinity Map. The areas where the characters are present are marked in the Region Map. The Affinity Map is a pictorially that represents the related character. Finally, the affinity and region scores are combined to give the bounding box of each word. The coordinates are in order:

(left-top), (right-top) (right-bottom), (left-bottom), where each coordinate is an (x, y) pair.

B. *Training*

Ground Truth Label Generation**:** A GT label is created for each learning image's region and affinity score with character bounding boxes. Region scores and affinity scores are encoded as Gaussian heat-map. The heat map representation is highly flexible when dealing with GT without strict limitations. The proposed GT definition makes detecting large and long text possible even if the model's receptive fields are small. Character-by-character detection allows the convolution filter to focus on the character-to-character relationship instead of the entire text instance.

Weakly-Supervised Learning: Unlike Synthetic image datasets, real datasets usually provide only word-by-word annotations. CRAFT creates a text box from a word-by-word annotation weakly supervised. The trained model generates a character bounding box by predicting a region score from an image in which a word unit annotation image is provided. To show the reliability of the intermediate model prediction, the reliability map for each word box is calculated in proportion to the number of detected characters divided by the number of

TABLE I
TEXT DETECTION DATASETS USED IN THE CRAFT[12]

Dataset	Language	Year	# Images			# Text instance			Text Shape		Annotation level		
			Total	Train	Test	Total	Train	Test	AQ	MO	Char	Word	Text-Line
MSRA-TD500 [34]	EN/CN	2012	500	300	200	-	-	-	✓	–	–	–	✓
ICDAR2013 [35]	EN	2013	462	229	233	1944	849	1095	–	–	✓	✓	–
ICDAR2015 [36]	EN	2015	1500	1000	500	17548	122318	5230	✓	–	–	✓	–
SynthText [37]	EN	2016	800k	-	-	6M	-	-	✓	–	–	✓	–
Total-Text [38]	EN	2017	1525	1225	300	9330	-	-	✓	✓	–	✓	✓
CTW-1500 [39]	EN/CN	2017	1500	1000	500	10751	-	-	✓	✓	–	✓	✓
MLT-2017 [40]	ML	2017	18000	7200	10800	-	-	-	✓	–	–	✓	–
COCO-Text [41]	EN	2014	63686	43686	20000	145859	118309	27550	✓	–	–	✓	–

Note: H: Horizontal, MO: Multi-Oriented, C: Curved, EN: English, CN: Chinese, ML: Multi-Language, D: Detection, R: Recognition, AQ: Arbitrary-Quadrilateral.

characters in the GT. This is used as a learning weight learning. The real datasets can be trained as follows:

- According to the label of the real-world data, the text line is cropped out.
- Run the network and get the resulting map.
- Split a single text based on the graph output from the network.
- Generate a label based on the results of the previous step.

These labels are not necessarily accurate, so training with these data does not necessarily make the model better, especially when the model is not accurate at first. Therefore, the paper uses an existing annotation result to give confidence that the text length is used. If the length of the text is the same as the result of the model, then the confidence is high. Otherwise, the confidence is low (the confidence is directly multiplied by the loss).

For generating pseudo-GT, the confidence is calculated from the length information of the text to determine whether the data is reliable, and the confidence is used to update the loss. The confidence of each word is calculated as follows:

$$S_{conf}(w) = \frac{l(w) - \min\left(l(w), |l(w) - l^c(w)|\right)}{l(w)} \quad (1)$$

Where l(w): word length of the given word-level annotation. $l^c(w)$: is the length of the word (number of bounding boxes) obtained from the score map. $0 < S_{conf}(w) < 1$: when the actual number of boxes equals the estimated number of boxes, close to 0 when different.

$$S_c(p) = \begin{cases} S_{conf}(w) & p \in R(w) \\ 1 & otherwise \end{cases} \quad (2)$$

Where, p: each pixel, R(w): area occupied by word w, Sc(p): is a confidence value for each pixel, and the Sc map has a confidence value of 0 to 1 on a pixel-by-word basis and the loss function is computed as follow:

$$L = \sum_p S_c(p)\left(\|S_r(p) - S_r^*(p)\|_2^2 + \|S_a(p) - S_a^*(p)\|_2^2\right) \quad (3)$$

Where, $S_r^*(p)$ and $S_a^*(p)$ are pseudo-ground truth region scores and affinity map, respectively. $S_r(p)$ and $S_a(p)$ are predicted region scores and affinity scores. They are used to localize the individual characters in the image, and group each character into a text instance, respectively. As learning progresses, the CRAFT model predicts letters more accurately, and the S_{conf} also gradually increases.

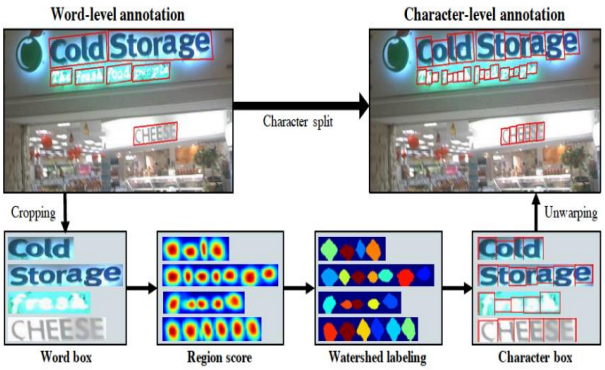

Fig. 3. Character-level annotation procedure of CRAFT [12].

C. Post Processing

In the post-processing stage, the output of the final stage can be varied in terms of word boxes, text boxes, and polygons.

After the network outputs the score map, the pixel-level labels are combined into a box. For this purpose, first, the threshold map is used to filter the score map, and a connected domain analysis is performed (connected component labeling), then draw the final quadrilateral bounding box through the connected domain.

D. Improvement

CRAFT has two significant drawbacks. Firstly, it is slow during training, especially for fine-tuning real-word datasets producing many post-processing stages for character-level annotation and using these labels for training. Secondly, the recall performance of detection does not improve fast. To address the first issue, we change the learning rate hyperparameters. To address the second issue, we modify the Gaussian heat map and watershed labeling thresholds in the post-processing framework of Fig. 3.

IV. Experiments

A. Datasets

Table I shows the benchmark datasets used in CRAFT [12]. We trained our model on ICDAR2015 and COCO-Text and evaluated ICDAR13, ICDAR15, and COCO-Text.

1) ICDAR2013: This dataset was introduced in the ICDAR 2013 Robust Reading Competition for focused scene text detection [35]. This dataset is annotated at the word level using rectangular boxes containing 229 and 233 images for training and testing, respectively.

2) ICDAR2015: This dataset was introduced in the ICDAR2015 for incidental scene text detection, which contains 1000 images for training and 500 image testing. The annotations of this dataset are at the word level using quadrilateral boxes.

3) ICDAR2017: This dataset is one of the enormous multi-lingual text datasets, including scene images in nine languages. ICDAR17 Multi-language dataset has 18000 images with 7200 training images, 1800 validation images, and 9000 testing images [40].

4) COCO-Text: This dataset introduced in [41] is the largest and most challenging text detection dataset, consisting of 43,686 annotated images used for the training and 10,000 images for training [42, 30]. The images are annotated with rectangle bounding boxes in this dataset at the word level.

5) Synth Text: In the Wild dataset [43], the SynthText contains 858,750 synthetic scene images with 7,266,866 word-instances and 28,971,487 characters. Moreover, most text instances in the data set have multi-oriented and annotated features with word and character-level rotated bounding boxes and text sequences. These instances are created by blending natural images with text rendered with random fonts, sizes, orientations, and colors.

B. Training Procedure

In summary, the training procedure is (1) an image-net pre-trained model, (2) pre-train on SynthText, and (3) fie-tune on real datasets. We perform all of our experiments on two NVIDIA Tesla T4 GPUs with 8GB RAM. We use Adam [44] as an optimizer and set the initial learning rate to 3.278e - 5. We multiply the learning rate by 0.8 at 5k iterations. We train our model for about three epochs, use a batch size of 8, and resize the input images to 768 × 768. We re-implemented the training code of CRAFT, and our models are trained from scratch. First, we trained CRAFT on the Synthetic dataset for 50k iterations and weakly supervised the COCO-Text and ICDAR2015 datasets for 75 and 175 epochs, respectively. For SynthText and COCO-Text, we follow the [12] when settings the hyperparameters, but for the ICDAR2015 model, we modified the learning rate and the post-processing threshold parameters to improve the performance of the paper.

C. Experimental Results

This dataset is one of the enormous multi-lingual text datasets, including scene images in nine languages. ICDAR17 Multi-language dataset has 18000 images with 7200 training images, 1800 validation images, and 9000 testing images [40].

$$H - mean = 2\frac{P \times R}{P + R} \quad (4)$$

We compare the effects of the ICDAR2013 and ICDAR2015 datasets that have been used in the [12]. Besides, a new dataset, namely COCO-Text, is used for evaluation, which has not been used in the main paper. Table II compares the IOU results of the pre-trained models.

Pre-trained models of the paper: In the official Github1 page of the CRAFT [12], researchers provide two models; The first model, Git_Model_MLT, is fine-tuned on a combination of ICDAR2013 [35] and ICDAR17-MLT [40] training images, and the model Git_Model_IC15 is fine-tuned only on the ICDAR2015 [36] dataset. Using the model Git_Model_IC15, H-mean declined by about 5% for ICDAR2015 and COCO-Text compared to model Git Model MLT and about 15% for the ICDAR2013 dataset. Thus, CRAFT requires more images for training to achieve better results. Therefore, to tackle this problem, we can use more data for training the craft. Fig. 4 compares the output results of two models on ICDAR2013 and ICDAR2015. As seen in Table II, CRAFT's performance for the two models declined significantly on the COCO-Text dataset. These models do not have generalization ability without fine-tuning on unseen datasets. Fig. 5 shows the output results of the two models on the COCO-Text.

Our trained models: As shown in Table II, we provide three models for CRAFT that are trained from scratch. The first model, Syndata, is trained on 800k train images of the SynthText [37] dataset, and for the training of this model, we follow the same structure and hyperparameters used in the CRAFT. Although this dataset has many images for training, this model performs poorly on real datasets because the text in real datasets is different from synthetic datasets.

For creating the Fine_Coco model, we fine-tuned the pre-trained Syndata model on the COCO-text dataset with many image and text instances for training. This model shows excellent precision performance in evaluation, especially in the COCO-text datasets compared to the Git_Model_IC15 of CRAFT. Still, the recall performance of this model was low because this model only trained for 75 epochs and needs optimization of hyperparameters.

The major limitation of the CRAFT framework is that it is too slow during fine-tuning of real datasets because it has much post-processing for the detection of individual characters in the image and generating word bounding boxes. For example, the authors of CRAFT used 4 GPU and 500 epochs to fine-tune the Git_Model_IC15 model. However, we fine-tuned the Fine_IC15 model for 175 epochs and two GPUs and achieved better results in comparison to the model of CRAFT (Table II).

For this purpose, we used a learning rate of 3.2768e-5 and decay of 0.8 for every 5k iterations compared to 1e-4 and 0.8 decay for 10k iterations used in the main paper. We improved

TABLE II

THE BASELINE AND OUR TRAINED MODEL ARE COMPARED USING IOU [36] EVALUATION METRIC. THE BEST PERFORMANCES ARE SHOWN IN BOLD. PROPOSED MODELS PERFORM BETTER THAN THE BASELINE MODELS (AUTHORS) FOR THE DATASETS IN THE STUDY.

	Model	ICDAR2013*			ICDAR2015			COCO-Text		
		Precision	Recall	H-mean	Precision	Recall	H-mean	Precision	Recall	H-mean
Authors	Paper	97.4	93.1	95.2	89.8	84.3	86.9	–	–	–
	Git_Model_MLT	92.2	90.4	91.3	88.9	80	84.2	64.6	58.2	61.2
	Git_Model_IC15	72.7	77.6	75.2	82.2	77.8	79.9	56.7	55.9	56.3
Ours	Syndata	75.8	66.8	71	70.8	38.5	49.9	46.8	22.5	30.4
	Fine_Coco	80.8	**77.5**	79.2	82.1	62.2	70.8	**70.0**	50.8	**58.9**
	Fine_IC15	**83.7**	76.7	**80.1**	**85.5**	**76.3**	**80.6**	62.0	**53.9**	57.7

*: This dataset is evaluated on the DetEval [35] evaluation metric in the paper, while here I used the IOU [36] metric.

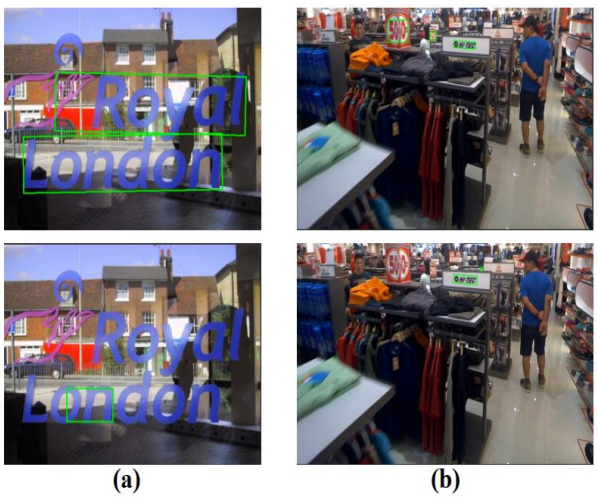

Fig. 4. The detected text of model Git_Model_MLT (above) and model Git_Model_IC15 (below) on (a) ICDAR2013, and (b) ICDAR2015 datasets. The model Git_Model_MLT trained on more images performs better than the model Git_Model_IC15.

Fig. 5. The output result of our final model on the COCO-Text dataset. The baseline CRAFT model performed poorly on the occluded and low-resolution text of these images. These results are obtained from the best model of CRAFT (Git Model MLT).

precision performance by about 11%, 3%, and 4% in the ICDAR2013, ICDAR2015, and COCO-Text datasets.

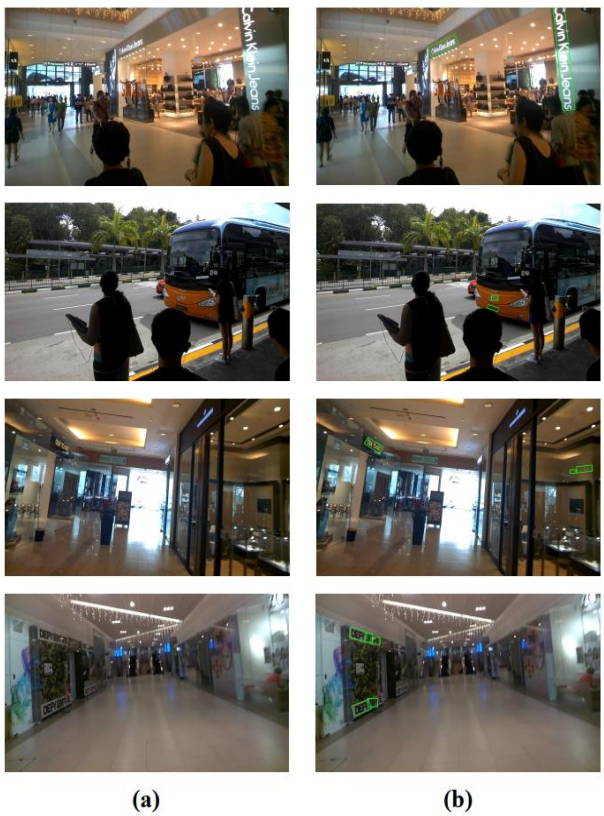

Fig. 6. Comparison of some selected images of the ICDAR2015 dataset: (a) The detection result of the Git_Model_IC15 model from the main paper, which failed to detect any text in these images, and (b) the detection results of our trained model, which performed better.

We also trained CRAFT on the COCO-Text dataset, in which we achieved a 14% precision improvement compared to the Git_Model_IC15 of the CRAFT. Fig. 6(a) shows some selected images from the ICDAR2015 dataset in which the pre-trained model (Git_Model_IC15) failed to detect text in them, and Fig. 6(b) shows the detection result of our trained model that performs better in these images.

We also compare our model with several previous [12, 13, 20, 45] and recent state-of-the-art [23, 46, 47] scene text

TABLE III
COMPARISON OF OUR MODEL WITH SEVERAL WELL-KNOWN PREVIOUS AND RECENT STATE-OF-THE-ART TEXT DETECTION METHODS.

Method	ICDAR13			ICDAR15			COCOText		
	Precision	Recall	H-mean	Precision	Recall	H-mean	Precision	Recall	H-mean
EAST [13]	**84.8**	74.24	79.2	83.6	73.5	78.2	55.4	32.8	41.3
Pixellink [20]	62.2	62.5	62.3	82.8	81.6	82.2	61.0	33.4	43.2
PSENET [45]	81.0	62.4	70.5	84.6	77.5	80.9	60.5	49.3	54.4
DBNet [23]	--	--	--	86.8	78.4	82.3	--	--	--
DRRG [46]	--	--	--	88.5	84.7	86.6	--	--	--
TextDCT[47]	--	--	--	**88.9**	**84.8**	**86.8**	--	--	--
Baseline [12]	72.7	77.6	75.1	82.2	77.8	82.2	56.7	**55.9**	56.3
Our_Model	83.7	**76.7**	**80.1**	85.5	76.3	80.6	**62.0**	53.9	**57.7**

detection methods in Table III on three benchmark datasets [35, 36, and 41].

For a fair comparison, we follow a similar procedure described in [48] for comparing our model with the previous methods in [12, 13, 20, 45]. Table III shows that our model outperformed these approaches regarding H-mean performance in all three benchmarks. Compared to the baseline [12] model with a similar setup of training and parameters, our model also obtained a higher margin of H-mean performance. These performances are evident in ICDAR13 (~5% higher) and ICDAR15 (~3% higher). For the unseen dataset, COCO-Text also achieved the best results in terms of precision and H-mean compared to the baseline model.

To further enhance our model's performance in scent text detection, we plan to explore various cutting-edge techniques in natural language processing and computer vision. This includes investigating the latest advancements in deep learning architectures, such as transformer-based models and attention mechanisms, which have shown great promise in tackling complex language tasks. Additionally, we will focus on incorporating domain-specific knowledge and contextual information to better understand scent-related text patterns and improve the model's ability to discern relevant information. Furthermore, data augmentation techniques will be utilized to create a more diverse and comprehensive training dataset, enabling our model to generalize better to real-world scenarios. By continuously refining and optimizing the model's architecture, we are confident in bridging the performance gap and establishing our model as a competitive solution for scent text detection tasks.

V. Conclusions

In this paper, we have re-implemented CRAFT. This text detection method uses a weakly supervised learning method to detect the character of scene text images CRAFT achieved state-of-the-art performance in most public benchmark datasets. By applying the pre-trained model provided by CRAFT researchers, we showed that CRAFT does not have the generalization capability on unseen datasets without fine-tuning. We also made some modifications to the model, which was fine-tuned on the ICDAR2015 dataset and improved CRAFT's performance with fewer iterations and higher accuracy compared to the model provided by the authors.

References

[1] H. Lin, P. Yang, and F. Zhang, "Review of scene text detection and recognition," *Archives of Computational Methods in Engineering*, pp. 433–454, 2020.

[2] S. Long, Y. Guan, B. Wang, K. Bian, and C. Yao, "Rethinking Irregular Scene Text Recognition," *arXiv.org*, Nov. 11, 2019. https://arxiv.org/abs/1908.11834.

[3] T. Diep, "State-of-the-art in action: unconstrained text detection, " in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2019, pp. 0-0.

[4] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image and Vision Computing*, vol. 22, no. 10, pp. 761–767, Sep, 2004.

[5] L. Neumann and J. Matas, "A method for text localization and recognition in real-world images," in *Asian Conference On Computer Vision, Springer*, 2010, pp. 770-783.

[6] B. Epshtein, E. Ofek, and Y. Wexler, "Detecting text in natural scenes with stroke width transform, " in *Proc. IEEE Conf. on Comp. Vision and Pattern Recognit*, 2010, pp. 2963–2970.

[7] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu, and A. Berg, "SSD: single shot multibox detector, " in *European Conference on Computer Vision, Springer*, Oct. 2016, pp. 21–37.

[8] 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.

[9] S. Ren, K. He, R. Girshick, and J. Sun. "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing*

Systems, 2015, pp. 91–99.
[10] 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.
[11] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," in *Proc. Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2961–2969.
[12] Y. Baek, et al. "Character region awareness for text detection" in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9365-9374.
[13] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, and J. Liang, "East: an efficient and accurate scene text detector," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5551–5560.
[14] M. Liao, B. Shi, X. Bai, X. Wang, and W. Liu, "Textboxes: A fast text detector with a single deep neural network," in *Thirty-First AAAI Conference on Artificial Intelligence*, Feb. 2017.
[15] Y. Liu and L. Jin, "Deep matching prior network: Toward tighter multi-oriented text detection, " in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1962–1969.
[16] M. Liao, B. Shi, and X. Bai, "Textboxes++: a single-shot oriented scene text detector," *IEEE Transactions on Image Processing*, Vol. 27, No. 8, pp. 3676–3690, Apr. 2018.
[17] J. Ma, W. Shao, H. Ye, L. Wang, H. Wang, Y. Zheng, and X. Xue, "Arbitrary-oriented scene text detection via rotation proposals, " *IEEE Transactions on Multimedia*, Vol. 20, No. 11, pp.3111–3122, 2018.
[18] X. Li, J. Liu, S. Zhang, and G. Zhang, "Learning to predict more accurate text instances for scene text detection," *arXiv preprint arXiv: 1911.07423*, 2019.
[19] P. He, W. Huang, T. He, Q. Zhu, Y. Qiao, and X. Li, "Single shot text detector with regional attention," in Proceedings of the IEEE international conference on computer vision, 2017, pp. 3047–3055.
[20] D. Deng, H. Liu, X. Li, and D. Cai, "Pixellink: Detecting scene text via instance segmentation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32, No. 1. 2018.
[21] S. Long, et al. "Textsnake: A flexible representation for detecting text of arbitrary shapes," In *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 20–36.
[22] 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.
[23] M. Liao, Z. Wan, C. Yao, K. Chen, and X. Bai, "Real-time scene text detection with differentiable binarization," In *AAAI Conf. on Artificial Intelligence*, 2020, pages 11474–11481.
[24] X. Liu, D. Liang, S. Yan, D. Chen, Y. Qiao, and J. Yan, "Fots: Fast oriented text spotting with a unified network," In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5676–5685.
[25] T. He, Z. Tian, W. Huang, C. Shen, Y. Qiao, and C. Sun, "An end-to-end text spotter with explicit alignment and attention," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5020–5029.
[26] P. Lyu, M. Liao, C. Yao, W. Wu, and X. Bai, "Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes," *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 67-83.
[27] Q. Wang, Y. Zheng, and M. Betke, "Sa-text: Simple but accurate detector for text of arbitrary shapes," arXiv preprint arXiv:1911.07046, 2019.
[28] H. Hu, C. Zhang, Y. Luo, Y. Wang, J. Han, and E. Ding, "Wordsup: Exploiting word annotations for character based text detection," *in Proceedings of the IEEE international conference on computer vision*, 2017, pp. 4940–4949.
[29] S. Zhang, M. Lin, T. Chen, L. Jin, and L. Lin, "Character proposal network for robust text extraction," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2016, pp. 2633–2637.
[30] C. Yao, X. Bai, N. Sang, X. Zhou, S. Zhou, and Z. Cao, "Scene text detection via holistic, multi-channel prediction," arXiv preprint arXiv: 1606.09002, 2016.
[31] B. Shi, X. Bai, and S. Belongie, "Detecting oriented text in natural images by linking segments," In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2550–2558.
[32] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, Springer, 2015, pp. 234–241.
[33] K. Simonyan and A. Zisserman, "Detecting oriented text in natural images by linking segments," CoRR, abs/1409.1556, 2014.
[34] C. Yao, X. Bai, W. Liu, Y. Ma, and Z. Tu, "Detecting texts of arbitrary orientations in natural images," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2012, pp. 1083–1090.
[35] D. Karatzas, et al. " Icdar 2013 robust reading competition, " in *12th international conference on document analysis and recognition*, 2013, pp. 1484–1493.
[36] D. Karatzas, et al. "Icdar 2015 competition on robust reading," in *13th international conference on document analysis and recognition (ICDAR, 2015)*, pp. 1156–1160.
[37] A. Gupta, A. Vedaldi, and A. Zisserman, "Synthetic data for text localisation in natural images, " In *Proc. IEEE Conf. on Comp. Vision and Pattern Recognit*, 2016, pp. 2315–2324.
[38] C.K. Ch'ng and C.S. Chan, "Total-text: A comprehensive dataset for scene text detection and recognition," In *Proc. IAPR Int. Conf. on Document Anal. and Recognit.* (ICDAR), Vol. 1, 2017, pp. 935–942.
[39] L. Yuliang, J. Lianwen, Z. Shuaitao, and Z. Sheng, "Detecting curve text in the wild: New dataset and new solution," In arXiv preprint arXiv: 1712.02170, 2017.
[40] M. Iwamura, N. Morimoto, K. Tainaka, D. Bazazian, L. Gomez, and D. Karatzas, "Icdar2017 robust reading challenge on omnidirectional video," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, Vol. 1, 2017, pp. 1448–1453.
[41] T. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision, Springer*, 2014, pp. 740–755.
[42] A. Veit, T. Matera, L. Neumann, J. Matas, and S. Belongie, "Coco-text: Dataset and benchmark for text detection and recognition in natural images," arXiv preprint arXiv:1601.07140, 2016.

[43] Z. Raisi, M.A. Naiel, P. Fieguth, S. Wardell, and John Zelek. "Text detection and recognition in the wild: A review, " arXiv preprint arXiv: 2006.04305, 2020.
[44] D.P. Kingma and J. Ba, "Adam: A method for stochastic optimization, " arXiv preprint arXiv:1412.6980, 2014.
[45] W. Wang, E. Xie, X. Li, W. Hou, T. Lu, G. Yu, and S. Shao, "Shape robust text detection with progressive scale expansion network," arXiv preprint arXiv:1903.12473, 2019.
[46] S.X. Zhang, et al. "Deep relational reasoning graph network for arbitrary shape text detection," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9699–9708.
[47] Y. Su, Z. Shao, Y. Zhou, F. Meng, H. Zhu, B. Liu, and R. Yao, "Textdct: Arbitrary-shaped text detection via discrete cosine transform mask", *IEEE Transactions on Multimedia*, 2022.
[48] Z. Raisi, et al. "Smart Text Reader System for People who are Blind Using Machine and Deep Learning," *Machine Learning Algorithms for Signal and Image Processing*, pp. 161-200, 2022.

Mahdi Kazeminia received the B.Sc., the M.Sc., and his Ph.D degree in telecommunication engineering from University of Sistan and Baluchestan, Iran, Zahedan in 2010, 2012, and 2019, respectively.
From 2017 to 2018, he was visiting researcher at the Univesity of Padova, Padova, Italy. Currently, he is an assistant professor in the Department of Electronics, Velayat University, Iranshahr, Iran. His research interests include resource allocation optimization, IoT networks, and D2D communication.

Hamed Shahraki was born in Zabol, Iran in 1984. He received BSc in Electrical Engineering from University of Sistan and Baluchestan, Zahedan, Iran in 2008, MS in Electrical Engineering from Shiraz University of Technology, Shiraz, Iran in 2012, and Ph.D. degree from Shahid Bahonar University of Kerman, Kerman, Iran in 2018. He is currently an Assistant Professor with the Electrical Engineering Department, Velayat University of Iranshahr, Iranshahr, Iran. His main areas of research interest are MTM and RF/micro wave circuits design.

Mehran Tamjidi is currently an Assistant Professor in the Department of Industrial Engineering at the University of Velayat. He received a B.S. in Industrial Engineering from the Azad University, Zahedan branch in 2005, and a M.S. in Manufacturing Engineering from University Malaya (UM) in 2012, and continued on to receive the Ph.D. in Manufacturing Systems Engineering from University Putra Malaysia (UPM) in 2017. His research interests include the machining process and optimization technics.