Re-Attention Is All You Need: Memory-Efficient Scene Text Detection via Re-Attention on Uncertain Regions

Hsiang-Chun Chang[1], Hung-Jen Chen[1], Yu-Chia Shen[1], Hong-Han Shuai[1] and Wen-Huang Cheng[2]

***Abstract*— Scene text detection plays an important role on vision-based robot navigation to many potential landmarks such as nameplates, information signs, floor button in the elevators. Recently, scene text detection with segmentation-based methods has been receiving more and more attention. The segmentation results can be used to efficiently predict scene text of various shapes, such as irregular text in most scene text images. However, two kinds of texts remain unsolved: 1) tiny and 2) blurry instances. Moreover, the annotations for tiny/blurry texts are usually ignored during training, while tiny/blurry texts can still offer visual auxiliaries for robots to understand the world. Therefore, in this paper, we propose a new approach to effectively detect both clear and blurry texts. Specifically, we propose a re-attention module without increasing the learnable parameters, which first predicts the region of texts as the candidate region and leverages the same network to detect the candidate region again for reducing the required memory. Moreover, to avoid the errors from the first detection propagating to the re-attended area, we propose a new fusion module that learns to integrate the results of the re-attended regions and the first prediction. Experimental results manifest that the proposed method outperforms state-of-the-art methods on four challenging datasets.**

I. Introduction

Automatic scene text detection draws a lot of attention due to the various applications of machine vision systems [24], [32], [14], such as autonomous robot navigation, visual SLAM, street address detection, and blind auxiliary. It has been served as the preprocessing block for Optical Character Recognition (OCR), which is also important from the semantic mapping perspective [3]. The main goal of scene text detection is to localize the bounding box or the area of each text instance. Early deep learning-based works use the regression-based methods [17], [27], [36], [37] to predict the bounding boxes of text instances. For instance, TextBoxes++ [17] applies quadrilaterals regression that is able to detect texts with different orientations. Moreover, DeRPN [36] proposes a dimension decomposition region proposal network to replace the bounding boxes of RPN with flexible anchor strings, decoupling width and height.

Although the detection accuracy is high for regular texts, the performance significantly drops for irregular texts, e.g., curved/arbitrary-shaped texts. To deal with the challenging texts, the segmentation-based methods [1], [15], [16], [18] have attracted a lot of attention since they are able to describe different shapes of texts according to the pixel-wise prediction. For instance, Mask TextSpotter [16] detects arbitrary-shape text instances based on the instance segmentation similar to Mask R-CNN. Due to the high post-processing time cost of transforming the segmentation into a binary mask, Liao *et al.* [18] proposes Differentiable Binarization (DB) and directly inserts DB into the network for jointly optimizing the segmentation.

Nevertheless, detecting the blurry texts remains a challenging task. In this paper, we argue that it is beneficial to consider the blurry texts since the blurry texts are still important for robot agents to explore the world. For example, given detected blurry texts, the agents can move forward to make the texts clear or use super-resolution techniques [22] for finding the visual auxiliaries. To detect both clear and blurry scene texts, one naïve method is to directly annotate more blurry texts for training the models. However, without a careful design, the detection models are inclined to detect backgrounds or textures since these areas may be similar to the blurry texts. Another basic approach is to use multi-level feature extraction [29], [33]. Nevertheless, it requires a complicated network for extracting the multi-level features.

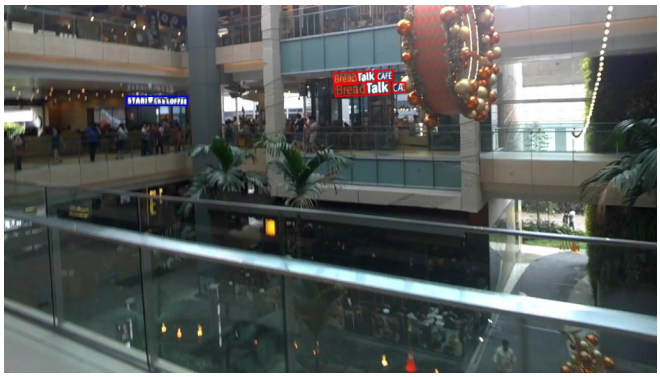
To effectively detect the scene texts, we propose a re-attention mechanism that mimics human learners' reactions. That is, when people find the objects are too small/blurry to be detected, we would look closer. Therefore, the re-attention mechanism first detects possible regions for text instance as the candidate region and leverages the same network to detect the candidate region again. In other words, by enlarging and cropping the candidate region, the part of the background is removed and the scene texts become clearer, which can facilitate the detection even with the same network and thus reduce the required memory for storing the detection models.

Fig. 1 illustrates an example of the proposed re-attention mechanism, where the red and blue boxes represent the clear and blurry texts, respectively. By re-attending the candidate regions, the proposed method effectively detects the blurry texts. It is worth noting that previous work of recurrent attention [4], [6], [21] or object detection based on recurrent neural networks [23], [35], [39], [40], [41] can be used to learn where to refocus. However, it requires additional learnable parameters and is inclined to be unstable during training according to previous work [4]. After applying the re-attention, we further propose a fusion module with a new loss to better integrate the first prediction and the re-attended prediction for avoiding the error propagation between two predictions. The contributions are summarized as follows.

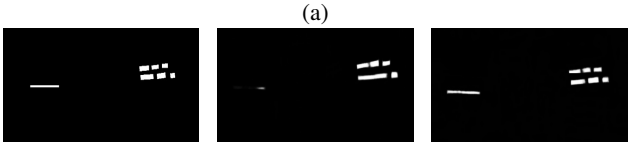
- To the best of our knowledge, this is the first work

[1]Hsiang-Chun Chang, Hung-Jen Chen, and Hong-Han Shuai are with the Department of Electrical and Computer Engineering, National Yang Ming Chiao Tung University, 30010 Taiwan.

[2]Wen-Huang Cheng is with the Institute of Electronics, National Yang Ming Chiao Tung Univresity, Hsinchu, 30010 Taiwan.



(a)



(b) (c) (d)

Fig. 1: The visualization of blurry ground truth and predictions. (a) is the input image. (b) GT (c) DB [18]. (d) Ours.

considering the blurry and short texts, which are ignored before. The detection can not only facilitate many applications but also help the learning model understand different levels of clarity.

- We propose a novel re-attention approach that does not require additional parameters and a fusion module that only slightly increases the number of parameters to avoid error propagation. Moreover, we propose a new loss that consider the different text and non-text ratios in the backbone and the re-attention modules.
- Experimental results show that the proposed approach outperforms 3 state-of-the-art methods on four public datasets. The source codes and pretrained models are released as a public download for the future research at http://basiclab.nctu.edu.tw/IROS21_STD.zip.

II. Related Work

To correctly detect scene texts, traditional methods, such as sliding-window-based method [11], [42], and connect-components based method [2], [10], [25], [31], [28] have been proposed to obtain the region of interest. With the advance of deep learning, a recent line of studies proposes different learning models to detect scene texts, which can be categorized into regression-based methods and segmentation-based methods. Regression-based methods predict the bounding boxes of text instances [17], [27], [36], while segmentation-based methods [1], [15], [16], [18], [26] describe different shapes of texts according to the pixel-wise prediction. Generally, regression-based methods usually enjoy simple post-processing algorithms (such as non-maximum suppression), but most of them cannot detect irregular shapes (such as curved shapes) well due to the limitation of bounding boxes. On the other hand, the segmentation-based methods are capable of detecting arbitrary text instances in scene images. For instance, inspired by Mask R-CNN, Mask TextSpotter [20] detects scene texts in a segmentation manner of arbitrary-shape text instances. Moreover, LOMO [39] localizes the text progressively for multiple times to reconstruct the irregular text by considering the geometry properties of text instances. Baek *et al.* propose CRAFT [1], which is a weakly-supervised framework that trains the character-level detector by using an iterative refinement module. However, none of these previous work leverages the labels for blurry/short texts. In contrast, our proposed approach leverages the same network twice to detect both clear and blurry texts, which improves the performance without increasing the required memory. Richardson *et al.* [26] propose to create a compact image containing only the initially-detected text regions and resize the compact image to a canonical scale and detect again. Nevertheless, the new results cannot be adaptively integrated with the initial results while forming a compact image requires more time.

III. Methodology

A. Overview

Fig. 2 illustrates the proposed architecture, which consists of three modules: 1) backbone network, which extracts features from images and detects the regions of texts, 2) re-attention network, which shares the same parameters with the backbone network but is fed with the re-attended images extracted by the re-attention algorithm, and 3) fusion module, which integrates the results of the first prediction and the re-attended prediction. In the backbone network, ResNet50 [9] is leveraged to extract multi-scale feature maps and Bi-directional Feature Pyramid Network (BiFPN) module is then used to fuse feature maps of different scales. Afterward, to predict the scene text region, we adopt the Differentiable Binarization (DB) module [18] as an important block for the following re-attention since the threshold of the segmentation should be optimized with the model. As such, the backbone network can be reused and generate different thresholds in a data-driven manner. The re-attention algorithm crops the image based on the first predition and scales the images into the original size. Finally, we concatenate the predictions from the standard segmentation and re-attention modules to generate the final prediction. After the backbone network and the re-attention network both generate a probability map and a binary map, the fusion module takes these four feature maps as the input and learn to obtain a better result.

B. Backbone Network

The goal of the backbone network is to i) extract features from images and detect the regions of texts and ii) make the prediction of the re-attention network different even with the same parameters. To achieve the first goal, we first leverage ResNet50 [9] to extract features. Afterward, inspired by [30], we enhance the extracted features by bidirectional cross-scale connections and weighted feature fusion modules. Specifically, ResNet50 generates four different resolution feature maps $P_i^{in}, i \in 2, \ldots, 5$ in each residual block, and outputs four feature maps $P_i^{out}, i \in 2, \ldots, 5$ with the

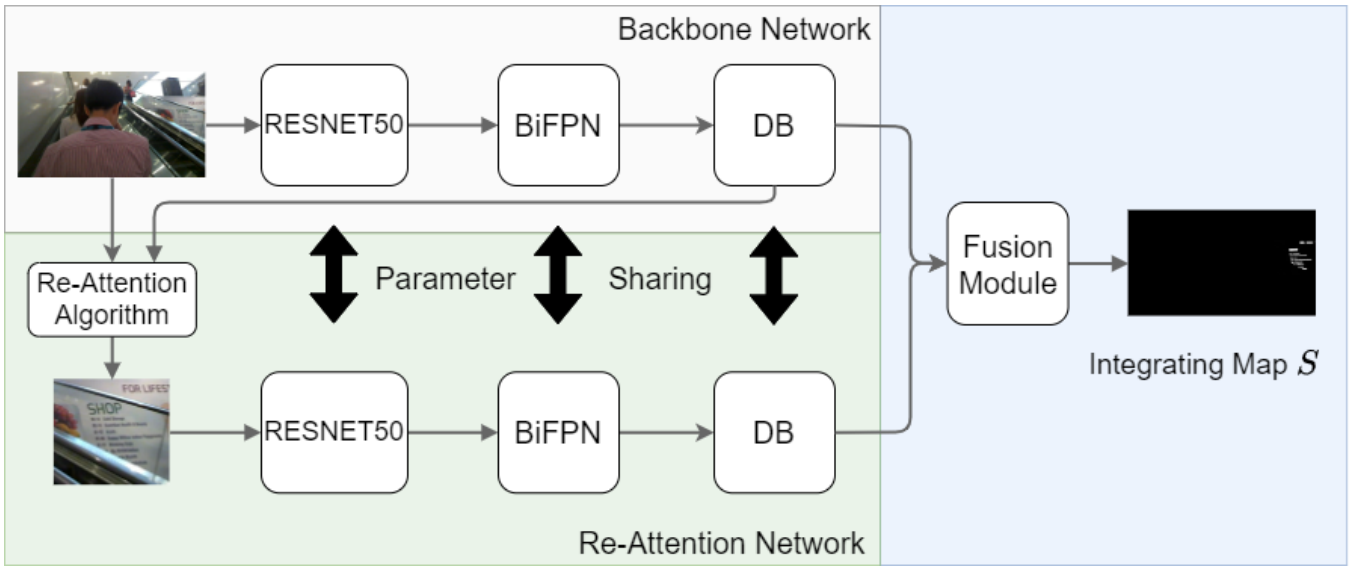


Fig. 2: The pipeline of our proposed model. The gray area is the standard segmentation (backbone network), the green area is the re-attention module, and the blue are is the fusion module. The weights ResNet50, BiFPN, and DB modules are shared between the standard segmentation and re-attention module.

same input size.[1] We use P^{in} to generate the intermediate features P^{td}, which are fused from top-down, so that lower-level features contain more semantic information. Take the intermediate features of the level 4 in BiFPN as an example:

$$P_4^{td} = Conv\left(\frac{w_{4_1}\cdot P_4^{in} + w_{4_2}\cdot Resize(P_5^{in})}{w_{4_1}+w_{4_2}+\epsilon}\right)$$

where $Resize$ is an up-sampling operation for resolution matching, $Conv$ is a convolutional operation for feature processing, and w_{4_1}, w_{4_2} are the weights for different features. Unlike the original work, we use the standard convolution instead of the depth-wise convolution, and only use one layer of BiFPN. Moreover, we fuse features from bottom-up and add skip connections to keep the geometry information for higher-level features, which can make multi-scale feature maps more powerful and have more geometric information. Again, take the output features of the level 4 in BiFPN as an example:

$$P_4^{out} = Conv\left(\frac{w'_{4_1}\cdot P_4^{in} + w'_{4_2}\cdot P_4^{td} + w'_{4_3}\cdot Resize(P_3^{out})}{w'_{4_1}+w'_{4_2}+w'_{4_3}+\epsilon}\right)$$

where $Resize$ is a down-sampling operation for resolution matching, $Conv$ is standard convolutional operation, $w'_{4_1}, w'_{4_2}, w'_{4_3}$ are the weights for different features. The output of the BiFPN module fuses the multi-scale features, so the text instances of different scales can be better detected.

To achieve the second goal, we adopt the Differentiable Binarization (DB) module [18] since the threshold of the segmentation should be optimized with the model. Fig. 3 illustrates the process of DB. This detector take $P2$ to $P5$ as the input and uses a 3×3 convolutional operator and two deconvolutional operators to transform the features for

[1]Different from traditional methods, the weights of the features are learnable parameters instead of addition or concatenation. Moreover, $P1$ is not used here since the features are too low-level.

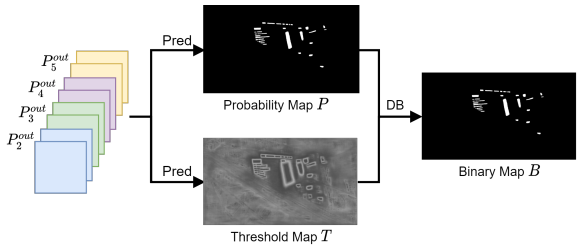


Fig. 3: The pipeline of the differential binarization module, where "Pred" consist of a 3x3 convolutional operator and two deconvolutional operators.

predicting the probability map (P) and the adaptive threshold map (T). The probability map represents the probability of each pixel belonging to the text instances. The adaptive threshold map is used to convert the probability map into the binary map. The pixel-wise threshold is a customized threshold for each pixel. The pixels close to the boundary of the bounding box have a higher threshold than pixels far away from the boundary. The probability map and threshold map are used together to approximate the binary map B, and make the operation differentiable so the threshold can be determined by an end-to-end training. Let $B_{i,j}$ denote the value locating at (i, j) of the binary map. We have

$$B_{i,j} = \frac{1}{1+e^{-k(P_{i,j}-T_{i,j})}}, \tag{1}$$

where k is the amplifying factor, which is set to 50 empirically. In other words, by pixelwisely calculating whether the probability ($P_{i,j}$) is greater than the threshold ($T_{i,j}$), we obtain the binary segmentation as the first prediction of scene texts.

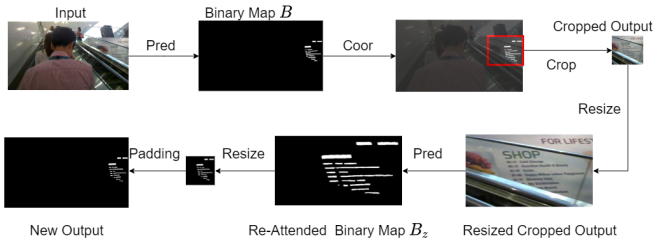


Fig. 4: The re-attention module can be divided into two parts. The first part is to generate a cropped image, and the second part is to pad the output of the re-attention module to the input image size. "Pred" is to get predictions from the model. "Coor" represents the operation of obtaining the coordinate from the area of interest.

C. Re-Attention Module

The scale of the text instances varies in the text detection task, especially when the text instances are located at different distances. To detect the blurry/small text instance, one basic approach is to use a supervised learning with annotations, *e.g.*, the labels of blurry or small in many scene text datasets. However, without a careful design, the detection models are inclined to detect backgrounds or textures since these areas may be similar to the blurry texts. Another approach is to use multi-level feature extraction [33], [29] or recurrent attention [6], [21], [4]. However, it requires additional parameters or complicated model architectures and may suffer from unstable training.

In this paper, we mimic the learning process of how human detect texts, *i.e.*, when we find the object is too small/blurry to detect, we would look closer. By focusing on the areas with texts and removing other backgrounds, the blurry/small text instances will be further detected, even with the same network. Fig. 4 shows that the binary map from a cropped image makes the text boundary clearer. Therefore, based on this observation, we design a re-attention module to deal with small/blurry text instances and use the re-attended image for further predictions.

Fig. 4 shows the pipeline to get a cropped image and the output prediction of the re-attention module. Specifically, we use the binary map (B) generated from the standard segmentation to locate the candidate areas, and crop the input image according to these coordinates.[2] The cropped image is then resized to the size of the input image as shown in Fig. 4. The resized image is fed into the same model mentioned above to obtain the results, including the re-attention probability map (P_z), the re-attention threshold map (T_z), and the re-attention binary map (B_z). The results of the re-attention module will be resized to the cropped size and padded to the input size as shown in Fig. 4. In order to prevent the cropped coordinates from being too close to the text boundary, we slightly enlarge the boundary regions. The model weights in the standard segmentation and the re-attention module branch are shared, which reduces the required memory during inference time.

[2]When there are many small text instances scattering over the image, one simple extension is to cluster the texts according to the coordinates, e.g., using DBSCAN [7], and send different clustered regions into the re-attention network.

D. Fusion Module

In order to prevent the situation that the predictions of the re-attention module are worse than the predictions of standard segmentation, we propose the fusion module to integrate the four predictions, *i.e.*, P, T, P_z and T_z. One simple fusion method is to learn a weighting vector for linearly combining four predictions. However, the boundary may be blurred since the combined values may not be close to 0 or 1, which deteriorates the performance. In order to overcome these problems, we use several 1×1 convolution kernels to fuse only the corresponding pixels for prediction. Compared with the pixel-wise integration with the image-wise integration, pixel-wise integration use different weighting for different regions, which yields a clearer boundary and removes the unacceptable predictions from re-attention module.

For the loss of the backbone network, since the number of text and non-text pixels is extremely imbalance, we adopt hard example mining to select negative pixels and apply a weighted binary cross-entropy loss to supervise pixel classification, i.e.,

$$L_{back} = \sum_{i \in S_{back}} y_i \log x_i + (1 - y_i) \log(1 - x_i), \quad (2)$$

where S_{back} is the sample set with the negative-positive ratio as 3. The loss function of the re-attention module is similar to L_{back}, but the sample set S_{att} is with the negative-positive ratio as 1 since the text/non-text regions are more balanced in the re-attention module, *i.e.*,

$$L_{att} = \sum_{i \in S_{att}} y_i \log x_i + (1 - y_i) \log(1 - x_i). \quad (3)$$

Finally, the prediction of the fusion module, denoted by L_{fuse}, is similar to L_{back} but with a consistency loss that keeps the prediction of non-attended regions consistent with the results from the backbone network, *i.e.*,

$$L_{fuse} = \sum_{i \in S_{fuse}} w_i (y_i \log x_i + (1 - y_i) \log(1 - x_i)), \quad (4)$$

where w_i represents the distance between the backbone network and the fusion module. The total loss function, denoted by L_{total}, can be regarded as the weighted sum of the losses for the backbone network (L_{back}), the re-attention network (L_{att}), and the fusion module (L_{fuse}), i.e.,

$$L_{total} = L_{back} + \lambda_1 L_{att} + \lambda_2 L_{fuse}, \quad (5)$$

where λ_1 and λ_2 are the hyperparameters controlling the importance of the three parts.

IV. Experiments

In this section, we pre-train all the models on *SynthText dataset* [8], and evaluate our method on 4 public datasets, *ICDAR 2015 dataset (IC15)* [12], *MSRA-TD500 dataset (TD500)* [38], *CTW1500 dataset (CT15)* [19], and *Total-Text dataset* [5]. Moreover, we compare the proposed approach with several state-of-art methods with open source codes, DB [18], PSENet [15] and ContourNet [34].

A. Datasets

Training text detection models usually pre-train the models by synthetic datasets. Following the setting, we introduce a synthetic dataset that is commonly-used in recent text detection papers for pre-training all the models, and four benchmarks for scene text detection consisting of regular text instances, irregular text instances, and multi-language text instances to demonstrate our model can handle curved, distorted, and oriented text. *It is worth noting that the blurry and small texts are annotated in all the datasets but ignored in previous work (named "DO NOT CARE"). This is because these datasets are designed for computer vision and do not consider the scenario that the mobile robots can move forward if they detect blurry or small texts.* The following descriptions of these datasets are used in our experiments:

- **SynthText** [8] contains 800,000 images. It has been widely used in scene text detection tasks because of the high cost of generating ground truth from real-world datasets. These images are synthesized from 8k background images. This dataset is only used to pre-train our model.
- **ICDAR 2015 dataset (IC15)** [12] is collected through a pair of Google Glasses. IC15 was introduced in the ICDAR 2015 Robust Reading Competition for incidental scene text detection. It consists of 1000 training images and 500 testing images with a resolution of 720 x 1280. The annotations are at the word level using quadrilateral boxes.
- **MSRA-TD500 dataset (TD500)** [38] is a multilingual dataset that contains 300 training images and 200 testing images. The images contain English and Chinese. Text instances are labeled in rectangles with rotation angle.
- **CTW1500 dataset (CT15)** [19] is a dataset for curved text detection. It contains 1,000 training images and 500 testing images. The text instances are annotated by a polygon of 14 vertices.
- **Total-Text dataset** [5] is a text benchmark containing the texts with various shapes, including horizontal, multi-oriented, and curved. There are 1255 training images and 300 testing images in total, while the text instances are labeled at the word-level.

B. Implementation details

We train the model in two stages: pre-training on SynthText and fine-tuning on the real-world datasets. Specifically, in the pre-training stage, we use the ResNet50 pre-trained on ImageNet [13] as the initial weights and use SynthText for pre-training the scene text detection. The batch size is set to 16. All the input images are resized to 640×640. The data augmentation of training data includes random rotation, random flipping, and random cropping. We use the same label generation method from [18] which generates the segmentation maps and the threshold maps. Then, we pre-train them for 5 epochs with the SGD optimizer with the initial learning rate as 0.007. Moreover, a customized decay learning rate is adopted by multiplying $(1 - \frac{iter}{max\_iter})^{power}$, where $power$ is 0.9 and max_iter means the maximum iterations, depending on the maximum epochs. Gradient clipping is used with the magnitude of 1. In the fine-tuning stage, we set the batch size to 8. The ground truth image size and the label generation are the same as in the pre-training stage. Finally, we fine-tune the model for 1200 epochs on the corresponding real-world datasets. The settings are the same as the pre-training stage.

In the testing phase, we keep the aspect ratio of the test images and resize the input images by setting a suitable height for each dataset. We use the single-scale image as input because the test images of different scales will have a great impact on performance. We find the hyperparameters for each dataset via a grid search with 0.05 step on a hold-out validation set. The whole model is implemented by PyTorch on GeForce RTX 2080 Ti.

C. Comparisons with State-of-the-Art Methods

We evaluate the proposed method on different datasets with different state-of-the-art methods. We choose DB [18], PSENet [15], and ContourNet [34] as the baseline to evaluate our model since they provide models, codes, and configuration files for reproducing their results. To fairly evaluate the performance, all the models are trained with annotations of blurry or small texts. It is worth noting that there are many papers provided their results on these four public datasets. However, without the codes, it is difficult to know the results when considering the blurry texts. In the following tables, "P", "R", and "F" respectively indicate the precision, recall, and f-measure.

Table I shows the performance of different approaches on 4 datasets. On average, the proposed approach outperforms DB, PSENet, and ContourNet by 4.46%, 13.23%, and 11.86%, respectively. Compared with PSENet and ContourNet, DB performs the best since the differential binarization is important to find a good threshold for both clear and blurry texts. Moreover, the frames per second (FPS) shows that the inference speed of the proposed approach is nearly real-time but greater than DB due to the re-attention mechanism. However, FPS of the proposed approach is not double the FPS of DB since the model weights can be reused instead of reloading them twice. We further analyze the performance with different characteristics of datasets.

Multi-oriented text detection. IC15 dataset is used to evaluate the performance on multi-oriented text detection with small and low-resolution text instances. Table I shows that the F-measure of DB is only 76.9%, while the proposed approach reaches 80.9%, showing that the proposed re-attention module and fusion module are effective. The

TABLE I: The evaluation on the real world dataset with blurry and small texts.

Model	IC15			TD500			CTW1500			Total-Text			Average			
	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	FPS
DB	79.3	74.6	76.9	89.0	81.4	84.5	81.0	69.3	74.7	**86.1**	77.4	82.1	83.8	75.4	79.5	**38.4**
PSENet	72.3	81.3	76.9	61.8	74.1	67.4	**83.6**	68.2	75.1	81.0	67.4	73.6	74.7	72.8	73.3	4.56
ContourNet	75.4	**81.8**	78.5	60.7	62.0	61.3	76.3	**77.7**	77.2	81.9	77.8	79.8	73.8	74.8	74.2	4.83
ours	**82.7**	79.2	**80.9**	**92.8**	**84.2**	**88.3**	83.0	76.0	**79.3**	**86.1**	**80.7**	**83.3**	**86.1**	**80.0**	**83.0**	29.3

(a) GT (b) DB (c) Ours

Fig. 5: Comparative results with DB on ICDAR2015 datasets.

TABLE II: The design of the fusion module.

Method	channels	bias	P	R	F
1x1	4-1	-	-	-	-
1x1	4-1	√	**94.3**	71.5	81.3
1x1	4-32-1	-	92.8	**84.2**	**88.3**

TABLE III: The input of the fusion module.

P	B	P_{att}	B_{att}	P	R	F
√	-	√	-	89.3	**85.6**	87.4
-	√	-	√	88.1	78.7	83.1
√	√	√	√	**92.8**	84.2	**88.3**

TABLE IV: The performance of five predictions.

method	P	R	F
Probability Map P	91.3	83.2	87.1
Binary Map B	76.7	**86.0**	81.0
Re-attention Probability Map P_{att}	89.4	78.0	83.3
Re-attention Binary Map B_{att}	67.1	82.0	73.8
Fusion Map	**92.8**	84.2	**88.3**

qualitative results comparing to DB are shown in Fig. 5, where the text instances are correctly detected by our model.

Curved text detection. CTW1500 and Total-text datasets are used to evaluate the performance of curved texts. ContourNet is the state-of-the-art methods on these two datasets. With the suppression of the false positives by only outputting predictions with a high response value in both orthogonal directions, ContourNet better describes the text regions. However, when considering the blurry/small texts, the proposed approach still outperforms ContourNet. Moreover, Fig. 6 shows that the curve text instances with artistic fonts can be detected in our method but are ignored by DB.

Multilingual text detection. TD500 dataset is a multilingual text detection. As shown in Tab. I, the proposed model also achieves state-of-the-art performance, which shows the generalizability of detecting scene texts. The performance considering the blurry text is 88.3%, which is better than other baselines. We observe that the text instances are shown together instead of being scattered on other datasets. The grouped text instances allow the proposed re-attention module to crop the area of interest more efficiently, which ignores the distracting background. As shown in Fig. 7, the multilingual text can be detected correctly by our method. In contrast, without using the re-attention module to eliminate the distracting background, DB detects the background of the signboard as texts since the texture behind is relatively similar to texts as compared with the ceiling.

D. Ablation study

We conduct ablation studies on TD500 dataset to show the performance improvement of different modules. All the models are trained and tested using the blurry text instances.

Fusion Module. There are several alternatives of the fusion module in Table II, where "Method" means different kinds of convolutional kernels. "(numbers)-(numbers)" indicates the input and output channels. For example, "4-32-1" indicates two convolutions with input channels 4, and output channels 32 and 1. "Bias" indicates whether to add learnable bias to the output. The first row in the table does not produce results since the model cannot effectively express the mixed results without bias. The best fusion module is designed as two 1×1 convolution and the corresponding output channels are 32 and 1. There are four predictions from the standard segmentation (backbone network) and the re-attention module. Table III shows the choice of the input features, where "√" marks the usage of different combinations of the probability map P, binary map B, re-attention probability map P_{att}, and re-attention binary map B_{att}. We choose 1) both probability maps, 2) both binary maps, and 3) four maps, as the input of the fusion module. The experiment shows that the fusion module with all of the predictions can obtain the best result in 88.3% on TD500 dataset.

Different Prediction Results. When the proposed model converges, Probability Map, Binary Map B, Re-Attention Probability Map P_{att}, Re-Attention Binary Map B_{att} and Fusion Map can all be regarded as the final result. As

(a) GT (b) DB (c) Ours

Fig. 6: Comparative results with DB on Total-text datasets.

(a) GT (b) DB (c) Ours

Fig. 7: Comparative results with DB on TD500 datasets.

TABLE V: The performance of different hyperparameters.

λ_1	λ_2	P	R	F
1	1	**92.8**	**84.2**	**88.3**
1	1.5	89.8	83	86.2
1.5	1	92.4	81.6	86.7
0.67	0.67	91.1	84	87.4
1	0.67	90.1	82.8	86.3
0.67	1	90.5	83.2	86.7
1.5	1.5	91.1	83	86.9

shown in Tab. IV, we can see that fusion module improves the performance to 88.3%, while the predictions from re-attention module are worse than standard segmentation, which indicates that the fusion module is necessary for avoiding the error propagation and predicting a better results. The fusion map shows the best result, and thus we use fusion map as the output of the proposed model.

Hyperparameters. In the proposed model, the total loss L_{total} is the weighted sum of the losses of the backbone network (L_{back}), re-attention network (L_{att}), and fusion module (L_{fuse}). Table V shows the performance of different weightings in terms of F1-score. The results indicate that the three branches are of the same importance since $\lambda_1 = 1$ and $\lambda_2 = 1$ yield the best performance. Moreover, when the importance of L_{back} and L_{att} are set to be the same, λ_2 cannot be too large (1.5) or too small (0.67), which yields the worst performance.

V. Conclusion

In this paper, a novel scene text detection framework is proposed to improve the performance by looking closer and ignoring background interference. Our method has two new design modules, re-attention and fusion modules, which effectively deal with both clear and blurry texts instance. Experiments show that the proposed approach outperforms state-of-the-art methods. In the future, as the scene text detection and text recognition are close tasks, we plan to formulate a multi-task learning problem to tackle these two problems together. Moreover, we plan to explore the possibility of extending the idea of "re-attention" to other related tasks such as 3D object detection.

Acknowledgment

This work is supported in part by the Ministry of Science and Technology (MOST) of Taiwan under the grants MOST-109-2218-E-009-025, MOST-109-2221-E-009-114-MY3, MOST-110-2218-E-A49-018, MOST-109-2221-E-009-097, MOST-110-2634-F-009-021, MOST-109-2223-E-009-002-MY3, MOST-110-2634-F-007-015, MOST-109-2218-E-002-015, and MOST-109-2327-B-010-005. This work was also supported by the Higher Education Sprout Project of the National Yang Ming Chiao Tung University and Ministry of Education (MOE), Taiwan. We are grateful to the National Center for High-performance Computing for computer time and facilities.

References

[1] Youngmin Baek, Bado Lee, Dongyoon Han, Sangdoo Yun, and Hwalsuk Lee. Character region awareness for text detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9365–9374, 2019.

[2] Michal Busta, Lukas Neumann, and Jiri Matas. Fastext: Efficient unconstrained scene text detector. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1206–1214, 2015.

[3] C. Case, B. Suresh, A. Coates, and A. Y. Ng. Autonomous sign reading for semantic mapping. In *2011 IEEE International Conference on Robotics and Automation*, pages 3297–3303, 2011.

[4] S. Chen, L. Zheng, Y. Zhang, Z. Sun, and K. Xu. Veram: View-enhanced recurrent attention model for 3d shape classification. *IEEE Transactions on Visualization and Computer Graphics*, 2019.

[5] Chee Kheng Ch'ng and Chee Seng Chan. Total-text: A comprehensive dataset for scene text detection and recognition. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 1, pages 935–942. IEEE, 2017.

[6] Xin Dai, Xiangnan Kong, Tian Guo, John Boaz Lee, Xinyue Liu, and Constance Moore. Recurrent networks for guided multi-attention classification. In *Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, 2020.

[7] Martin Ester, Hans-Peter Kriegel, Jorg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, 1996.

[8] Ankush Gupta, Andrea Vedaldi, and Andrew Zisserman. Synthetic data for text localisation in natural images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2315–2324, 2016.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[10] Han Hu, Chengquan Zhang, Yuxuan Luo, Yuzhuo Wang, Junyu Han, and Errui Ding. Wordsup: Exploiting word annotations for character based text detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4940–4949, 2017.

[11] Max Jaderberg, Andrea Vedaldi, and Andrew Zisserman. Deep features for text spotting. In *European conference on computer vision*, pages 512–528. Springer, 2014.

[12] Dimosthenis Karatzas, Lluis Gomez-Bigorda, Anguelos Nicolaou, Suman Ghosh, Andrew Bagdanov, Masakazu Iwamura, Jiri Matas, Lukas Neumann, Vijay Ramaseshan Chandrasekhar, Shijian Lu, et al. Icdar 2015 competition on robust reading. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 1156–1160. IEEE, 2015.

[13] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[14] B. Li, D. Zou, D. Sartori, L. Pei, and W. Yu. Textslam: Visual slam with planar text features. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2102–2108, 2020.

[15] Xiang Li, Wenhai Wang, Wenbo Hou, Ruo-Ze Liu, Tong Lu, and Jian Yang. Shape robust text detection with progressive scale expansion network. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[16] Minghui Liao, Pengyuan Lyu, Minghang He, Cong Yao, Wenhao Wu, and Xiang Bai. Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(2):532–548, 2021.

[17] Minghui Liao, Baoguang Shi, and Xiang Bai. Textboxes++: A single-shot oriented scene text detector. *IEEE transactions on image processing*, 27(8):3676–3690, 2018.

[18] Minghui Liao, Zhaoyi Wan, Cong Yao, Kai Chen, and Xiang Bai. Real-time scene text detection with differentiable binarization. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 11474–11481, 2020.

[19] Yuliang Liu, Lianwen Jin, Shuaitao Zhang, Canjie Luo, and Sheng Zhang. Curved scene text detection via transverse and longitudinal sequence connection. *Pattern Recognition*, 90:337–345, 2019.

[20] Pengyuan Lyu, Minghui Liao, Cong Yao, Wenhao Wu, and Xiang Bai. Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[21] Volodymyr Mnih, Nicolas Heess, Alex Graves, and Koray Kavukcuoglu. Recurrent models of visual attention. In *Proceedings of the International Conference on Neural Information Processing Systems*, 2014.

[22] Yongqiang Mou, Lei Tan, Hui Yang, Jingying Chen, Leyuan Liu, Rui Yan, and Yaohong Huang. Plugnet: Degradation aware scene text recognition supervised by a pluggable super-resolution unit. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[23] Y. Piao, W. Ji, J. Li, M. Zhang, and H. Lu. Depth-induced multi-scale recurrent attention network for saliency detection. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[24] I. Posner, P. Corke, and P. Newman. Using text-spotting to query the world. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3181–3186, 2010.

[25] Siyang Qin and Roberto Manduchi. A fast and robust text spotter. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–8. IEEE, 2016.

[26] Elad Richardson, Yaniv Azar, Or Avioz, Niv Geron, Tomer Ronen, Zach Avraham, and Stav Shapiro. It's all about the scale - efficient text detection using adaptive scaling. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1844–1853, 2020.

[27] B. Shi, X. Bai, and S. Belongie. Detecting oriented text in natural images by linking segments. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[28] Baoguang Shi, Xiang Bai, and Serge Belongie. Detecting oriented text in natural images by linking segments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2550–2558, 2017.

[29] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern recognition (CVPR)*, 2019.

[30] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[31] Zhi Tian, Weilin Huang, Tong He, Pan He, and Yu Qiao. Detecting text in natural image with connectionist text proposal network. In *European conference on computer vision*, pages 56–72. Springer, 2016.

[32] H. Wang, C. Finn, L. Paull, M. Kaess, R. Rosenholtz, S. Teller, and J. Leonard. Bridging text spotting and slam with junction features. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3701–3708, 2015.

[33] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, Wenyu Liu, and Bin Xiao. Deep high-resolution representation learning for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2020.

[34] Yuxin Wang, Hongtao Xie, Zheng-Jun Zha, Mengting Xing, Zilong Fu, and Yongdong Zhang. Contournet: Taking a further step toward accurate arbitrary-shaped scene text detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11753–11762, 2020.

[35] Zhe Wu, Li Su, and Qingming Huang. Stacked cross refinement network for edge-aware salient object detection. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[36] Lele Xie, Yuliang Liu, Lianwen Jin, and Zecheng Xie. Derpn: Taking a further step toward more general object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 33, pages 9046–9053, 2019.

[37] Linjie Xing, Zhi Tian, Weilin Huang, and Matthew R. Scott. Convolutional character networks. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[38] Cong Yao, Xiang Bai, Wenyu Liu, Yi Ma, and Zhuowen Tu. Detecting texts of arbitrary orientations in natural images. In *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1083–1090. IEEE, 2012.

[39] Chengquan Zhang, Borong Liang, Zuming Huang, Mengyi En, Junyu Han, Errui Ding, and Xinghao Ding. Look more than once: An accurate detector for text of arbitrary shapes. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[40] L. Zhang, J. Wu, T. Wang, A. Borji, G. Wei, and H. Lu. A multistage refinement network for salient object detection. *IEEE Transactions on Image Processing*, 29:3534–3545, 2020.

[41] Xiaoning Zhang, Tiantian Wang, Jinqing Qi, Huchuan Lu, and Gang Wang. Progressive attention guided recurrent network for salient object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[42] Siyu Zhu and Richard Zanibbi. A text detection system for natural scenes with convolutional feature learning and cascaded classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 625–632, 2016.