Real-Time Scene Text Detection with Differentiable Binarization and Adaptive Scale Fusion

Minghui Liao, Zhisheng Zou, Zhaoyi Wan, Cong Yao, Xiang Bai

Abstract—Recently, segmentation-based scene text detection methods have drawn extensive attention in the scene text detection field, because of their superiority in detecting the text instances of arbitrary shapes and extreme aspect ratios, profiting from the pixel-level descriptions. However, the vast majority of the existing segmentation-based approaches are limited to their complex post-processing algorithms and the scale robustness of their segmentation models, where the post-processing algorithms are not only isolated to the model optimization but also time-consuming and the scale robustness is usually strengthened by fusing multi-scale feature maps directly. In this paper, we propose a Differentiable Binarization (DB) module that integrates the binarization process, one of the most important steps in the post-processing procedure, into a segmentation network. Optimized along with the proposed DB module, the segmentation network can produce more accurate results, which enhances the accuracy of text detection with a simple pipeline. Furthermore, an efficient Adaptive Scale Fusion (ASF) module is proposed to improve the scale robustness by fusing features of different scales adaptively. By incorporating the proposed DB and ASF with the segmentation network, our proposed scene text detector consistently achieves state-of-the-art results, in terms of both detection accuracy and speed, on five standard benchmarks.

Index Terms—Scene Text Detection, Arbitrary Shapes, Real-Time

1 Introduction

Reading text in scene images [36] is of great importance in both academia and industry due to its abundant real-world applications, including office automation, visual search, geo-location, and blind auxiliary. Scene text detection, which aims to localize the text instances in the images, is an essential component in scene text reading. Although huge progress has been achieved in recent years, scene text detection is still challenging due to the diverse scales, irregular shapes, and extreme aspect ratios of the text instances.

As a mainstream of scene text detection, segmentation-based scene text detectors usually have advantages in handling text instances of irregular shapes and extreme aspect ratios due to their pixel-level representation and local prediction. However, most of them rely on complex post-processing algorithms to group the pixels into text regions, resulting in a considerable time cost in the inference period. For instance, PSENet [49] applied a progressive scale expansion algorithm to integrate multi-scale results and Tian *et al.* [46] adopted pixel embedding to group the pixels by calculating the feature distances among pixels. Besides, they mostly boosted the scale robustness of the segmentation network by applying a feature-pyramid [29] or U-Net [42] structure to fuse the feature maps of different scales, which did not explicitly fuse the multi-scale features adaptively for the text instances of different scales.

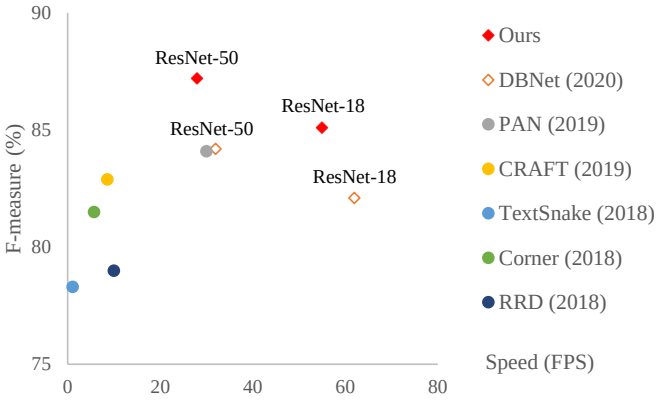


Fig. 1: The comparisons of several recent scene text detection methods on the MSRA-TD500 dataset [59], in terms of both accuracy (F-measure) and speed. Our method achieves the ideal tradeoff between effectiveness and efficiency.

A basic post-processing pipeline is described in Fig. 2 (following the blue arrows): First, the probability map produced from the segmentation network is converted to a binary image by applying a step function with a constant threshold; Then, some heuristic techniques like pixel clustering are used to group pixels into text regions. The above-mentioned two processes are standalone, without participating in the training process of the segmentation network, which may cause low detection accuracy. For more effective post-processing procedures while keeping the efficiency, we propose to insert the binarization operation into the segmentation network for joint optimization (following the red arrows in Fig. 2). First, a threshold map is predicted

- *M. Liao and Z. Zou are with the School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan, 430074, China.*
- *X. Bai is with the School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan, 430074, China.*
- *Z. Wan is with University of Rochester, Rochester, 14627, US.*
- *C. Yao is with Alibaba DAMO Academy, Beijing, 100102, China.*

Corresponding author: Xiang Bai.

adaptively, where the thresholds can be diverse in different regions. This design is inspired by the observation that the boundary regions of the text instances may be with lower confidence scores than the central regions in the segmentation results. Then, we introduce an approximate function for binarization called Differentiable Binarization (DB), which binarizes the segmentation map using the threshold map. In this manner, the segmentation network is jointly optimized with the binarization process, leading to better detection results.

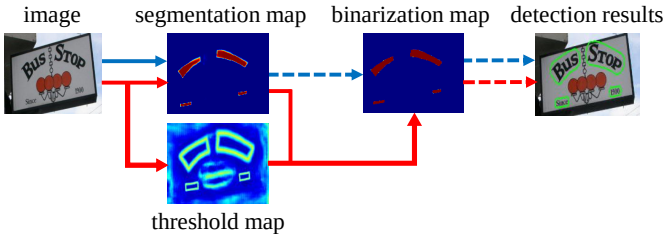


Fig. 2: A traditional pipeline (blue flow) and our pipeline (red flow). Dashed arrows are the inference only operators; solid arrows indicate differentiable operators in both training and inference.

Different from the previous methods that directly fused the multi-scale feature maps to improve the scale robustness of the segmentation network, we propose an Adaptive Scale Fusion (ASF) module to adaptively fuse the multi-scale feature maps. ASF integrates a spatial attention module into a stage-wise attention module. The stage-wise attention module learns the weights of the feature maps of different scales and the spatial attention module learns the attention across the spatial dimensions, leading to scale-robust feature fusion.

This paper is an extension of its conference version DBNet [26]. It extends the conference version from two aspects. First, it proposes an ASF module to further enhance the scale robustness of the segmentation model, without obvious loss of efficiency. The improvements brought by the proposed ASF are positively related to the scale distribution of the scene text benchmarks. Second, we give a more comprehensive theoretical analysis for the proposed DB module.

An accurate, robust, and efficient scene text detector, named DBNet++, is created by integrating the proposed DB module and ASF module into a segmentation network. Profiting from the joint optimization with the segmentation network, DB not only improves the quality of the segmentation results but also contributes to a simpler post-processing algorithm. By applying ASF to the segmentation network, its ability to detect the text instances of diverse scales is distinctly strengthened. The prominent advantages of DBNet++ over the previous state-of-the-art methods are concluded as follows:

1) Jointly optimized with the proposed DB module, our segmentation network can produce highly robust segmentation results, significantly improving the text detection results.
2) As DB can be removed in the inference period without sacrificing the accuracy, there is no extra memory/time cost for inference.
3) The scale robustness of the segmentation model can be efficiently improved by the proposed ASF module.
4) DBNet++ achieves consistently state-of-the-art accuracy on five scene text detection benchmarks, including horizontal, multi-oriented, and curved text.

The rest paper is organized as follows: Sec. 2 reviews the relevant scene text detection methods. We describe DBNet++ in Sec. 3. The experiments are discussed and analyzed in Sec. 4. The conclusions are summarized in Sec. 5.

2 Related Work

2.1 Text Detection

The early scene text detection methods usually detected the individual characters or components and then grouped them into words. Neumann and Matas [41] proposed to locate characters by classifying Extremal Regions (ERs). They posed the character detection problem as an efficient sequential selection from the set of Extremal Regions. Then, the detected ERs were grouped into words. Jaderberg *et al.* [21] first generated word candidates with a proposal generator. Then, the word candidates were filtered by a random forest classifier. Finally, the remaining word candidates were refined with a regression network.

Recently, deep learning has dominated the scene text detection area. The deep-learning-based scene text detection methods can be roughly classified into three categories according to the granularity of the predicted target: regression-based methods, part-based methods, and segmentation-based methods.

Regression-based methods are a series of models which directly regress the bounding boxes of the text instances. TextBoxes [25] modified the anchors and the scale of the convolutional kernels based on SSD [30] for text detection. TextBoxes++ [24] and DMPNet [31] applied quadrilaterals regression to detect multi-oriented text. SSTD [15] proposed an attention mechanism to roughly identifies text regions. RRD [27] decoupled the classification and regression by using rotation-invariant features for classification and rotation-sensitive features for regression, for better effect on multi-oriented and long text instances. EAST [65] and DeepReg [17] are anchor-free methods, which applied pixel-level regression for multi-oriented text instances. DeRPN [54] proposed a dimension-decomposition region proposal network to handle the scale problem in scene text detection. Regression-based methods usually enjoy simple post-processing algorithms (e.g. non-maximum suppression). However, most of them are limited to represent accurate bounding boxes for irregular shapes, such as curved shapes.

Part-based methods firstly detect the small parts of the text instances and then link/combine them into word/text-line bounding boxes. SegLink [43] regressed the bounding boxes of the text segment and predicted their links, to deal with long text instances. SegLink++ [44] further proposed an instance-aware component grouping algorithm to separate the close text instances more effectively and improved the linking algorithm to fit the arbitrary-shape text instances. These methods usually are skilled at detecting long text lines. However, the linking algorithms are quite complex

with hand-crafted super-parameters, which makes them hard to tune.

Segmentation-based methods usually combine pixel-level prediction and post-processing algorithms to get the bounding boxes. Zhang *et al.* [63] detected multi-oriented text by semantic segmentation and MSER-based algorithms. Text border is used in Xue *et al.* [56] to split the text instances. Mask TextSpotter [23], [38] detected arbitrary-shape text instances in an instance segmentation manner based on Mask R-CNN [13]. PSENet [49] proposed progressive scale expansion by segmenting the text instances with different scale kernels. Pixel embedding is proposed in Tian *et al.* [46] to cluster the pixels from the segmentation results. PSENet [49] and SAE [46] proposed new post-processing algorithms for the segmentation results, resulting in lower inference speed. Instead, our method focuses on improving the segmentation results by including the binarization process into the training period, without the loss of the inference speed.

Fast scene text detection methods focus on both the accuracy and the inference speed. TextBoxes [25], TextBoxes++ [24], SegLink [43], and RRD [27] achieved fast text detection by following the detection architecture of SSD [30]. EAST [65] proposed to use an anchor-free design to achieve a good tradeoff between accuracy and speed. Most of them can not deal with text instances of irregular shapes, such as curved shapes. Compared to the previous fast scene text detectors, our method not only runs faster but also can detect text instances of arbitrary shapes. Recently, PAN [50] proposed to adopt a low computational-cost segmentation head and learnable post-processing for scene text detection, yielding an efficient and accurate arbitrary-shaped text detector. Our proposed DBNet++ performs more accurately and more efficiently owing to the simple and efficient differentiable binarization algorithm.

2.2 Attention Mechanisms for Image Classification

Some previous image classification methods use channel attention and spatial attention to enhance the accuracy of image classification. Wang *et al.* [48] proposed residual attention network for image classification. It adopted a mixture of channel attention and spatial attention to produce a soft mask for the features. Hu *et al.* [19] proposed a "Squeeze-and-Excitation" block that recalibrates channel-wise feature responses by explicitly modeling interdependencies between channels. Woo *et al.* [52] proposed to adopt a channel attention module and a spatial attention module to refine the features. These methods mainly refine the independent features by various types of attention. Our proposed adaptive scale fusion focuses on fusing the features of different scales.

2.3 Multi-Scale Feature Fusion and Context Enhancement for Semantic Segmentation

Context is critical for semantic segmentation methods. Context and scale are two highly related concepts, where the context can help perceive large-scale objects/scenes while the multi-scale fusion strategies can usually provide more context. Thus, multi-scale feature fusion is commonly applied in semantic segmentation methods.

Multi-Scale Feature Fusion FCN [35] firstly proposed the fully convolutional network to fuse multi-scale features with upsampling layers. U-net [42] applied the skip connection which directly connected the low-level features and high-level features while inherited the structure from the FCN. PSPNet [64] and Deeplabv3 [4] proposed a pyramid pooling module (PPM) and an Atrous Spatial Pyramid Pooling (ASPP) for multi-scale feature fusion respectively. RefineNet [28], Deeplabv3+ [5], DFN [60], and SPGNet [6] adopted encoder-decoder structure which fuse high-level and low-level features to get more discriminating feature. Compared to these multi-scale feature fusion methods, our proposed ASF learns the weights of multi-scale features along with the attention across both the scale and the spatial dimensions.

Context Enhancement with Attention Mechanisms Attention mechanisms are popular in the semantic segmentation methods for enhancing the context. ANN [67], CCNet [20], GCNet [3], DANet [11] and ACFNet [62] all used self-attention to fuse different features for contextual information. Choi *et al.* [7] used height-driven attention to get height-dimensional information from multi-scale feature. Compared to these methods that applied attention mechanisms to the spatial dimensions to enhance the context, our proposed ASF focuses on the attentional fusion of multi-scale features.

3 Methodology

The architecture of our proposed method is shown in Fig. 3. Firstly, the input image is fed into a feature-pyramid backbone. Secondly, the pyramid features are up-sampled to the same scale and passed to the Adaptive Scale Fusion (ASF) module to produce contextual feature F. Then, feature F is used to predict both the probability map (P) and the threshold map (T). After that, the approximate binary map ($\hat{B}$) is calculated by P and F. In the training period, the supervision is applied on the probability map, the threshold map, and the approximate binary map, where the probability map and the approximate binary map share the same supervision. In the inference period, the bounding boxes can be obtained easily from the approximate binary map or the probability map by a box formation process.

3.1 Adaptive Scale Fusion

The features of different scales are with different perceptions and receptive fields, thus they focus on describing the text instances of different scales. For example, the shallow or large-size features can perceive details of the small text instances but can not capture a global view of the large text instances while the deep or small-size features are opposite. To take full advantage of the features of different scales, feature-pyramid [29] or U-Net [42] structures are commonly adopted in semantic segmentation methods. Different from most of the semantic segmentation methods that fuse the features of different scales by simply cascading or summing up, our proposed Adaptive Scale Fusion is designed to dynamically fuse the features of different scales.

As shown in Fig. 4, the features of different scales are scaled into the same resolution before being fed into the

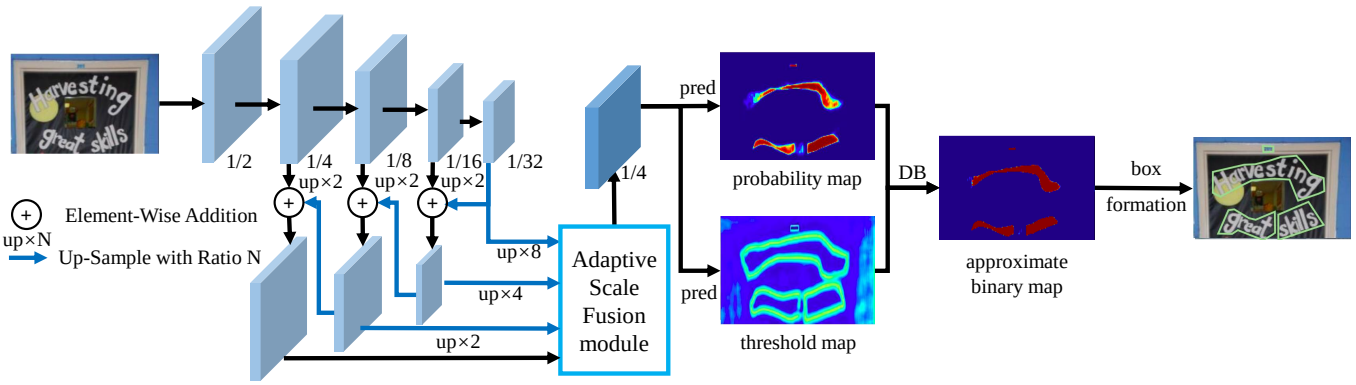


Fig. 3: Architecture of our proposed DBNet++, where the Adaptive Scale Fusion module is shown in Fig. 4.

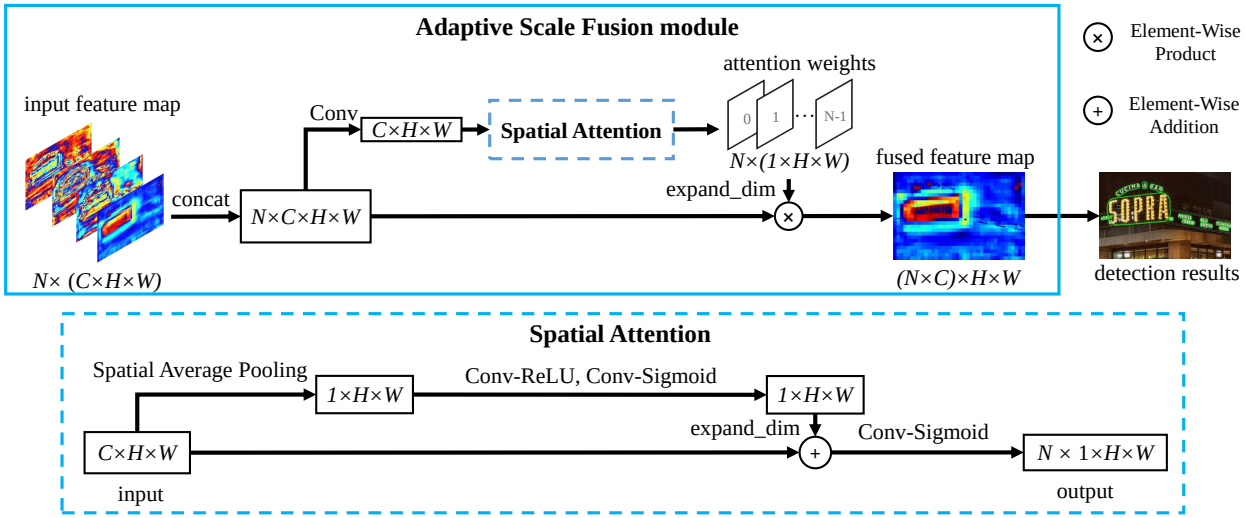


Fig. 4: Illustration of the Adaptive Scale Fusion module.

ASF module. Assuming that the input feature maps consist of N feature maps $X \in \mathcal{R}^{N\times C\times H\times W} = \{X_i\}_{i=0}^{N-1}$, where N is set to 4. Firstly, we concatenate the scaled input features X and then a 3×3 convolutional layer is followed to obtain an intermediate feature $S \in \mathcal{R}^{C\times H\times W}$. Secondly, the attention weights $A \in \mathcal{R}^{N\times H\times W}$ can be calculated by applying a spatial attention module to the feature S. Thirdly, the attention weights A can be split into N parts along the channel dimension and weighted multiply with corresponding scaled feature to get the fused feature $F \in \mathcal{R}^{N\times C\times H\times W}$. In this way, the scale attention is defined as:

$$\begin{aligned} S &= Conv(concat([X_0, X_1, ..., X_{N-1}])) \\ A &= Spatial\_Attention(S) \\ F &= concat([E_0X_0, E_1X_1, ..., E_{N-1}X_{N-1}]) \end{aligned} \tag{1}$$

where $concat$ indicates the concatenation operator; $Conv$ represents the 3×3 convolutional operator; $Spatial\_Attention$ indicates a spatial attention module, which is illustrated in Fig. 4. The spatial attention mechanism in the ASF makes the attention weights more flexible across the spatial dimension.

3.2 Binarization

Standard Binarization Given a probability map $P \in R^{H\times W}$ produced by a segmentation network, where H and W indicate the height and width of the map, it is essential to convert it to a binary map $B \in R^{H\times W}$, where pixels with value 1 are considered as valid text areas. Usually, this binarization process can be described as follows:

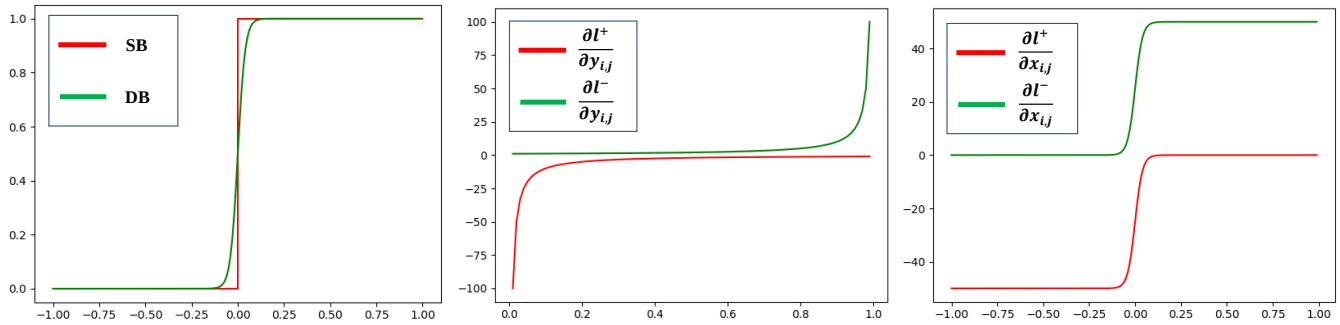
$$B_{i,j} = \begin{cases} 1 & \text{if } P_{i,j} \geq t, \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

where t is the predefined threshold and (i, j) indicates the coordinate point in the map.

Differentiable Binarization The standard binarization described in Eq. 2 is not differentiable. Thus, it can not be optimized along with the segmentation network in the training period. To solve this problem, we propose to perform binarization with an approximate step function:

$$\hat{B}_{i,j} = \frac{1}{1 + e^{-k(P_{i,j} - T_{i,j})}} \tag{3}$$

where $\hat{B}$ is the approximate binary map; T is the adaptive threshold map learned from the network; k indicates the amplifying factor. k is set to 50 empirically. This approximate binarization function behaves similar to the standard



(a) SB and DB (b) The derivative of losses in Eq. 6 (c) The derivative of losses in Eq. 8

Fig. 5: Numerical comparisons of different functions and derivatives.

binarization function (see Fig 5a) but is differentiable thus can be optimized along with the segmentation network in the training period. The differentiable binarization with adaptive thresholds can not only help differentiate text regions from the background, but also separate text instances which are closely jointed. Some examples are illustrated in Fig.9.

3.3 Analysis of Differentiable Binarization

The reasons that DB improves the performance can be explained by the gradients in the backpropagation. Let's take the binary cross-entropy loss as an example. The binary cross-entropy loss can be expressed as:

$$L_{bce} = -\frac{1}{N}\sum_{i=1}^{H}\sum_{j=1}^{W} y_{\hat{i},j} log(y_{i,j}) + (1 - y_{\hat{i},j}) log(1 - y_{i,j}) \quad (4)$$

where $y_{i,j} \in [0, 1]$ and $y_{\hat{i},j} \in \{0, 1\}$ indicate the output value with logits and the target value. Thus, in the segmentation task, the loss l^+ for positive labels and l^- for negative labels are:

$$\begin{aligned} l^+ &= -\log(y_{i,j}) \\ l^- &= -\log(1 - y_{i,j}) \end{aligned} \quad (5)$$

Without Considering Activation Function The differential of the segmentation loss can be calculated with the chain rule:

$$\begin{aligned} \frac{\partial l^+}{\partial y_{i,j}} &= \frac{-1}{y_{i,j}} \\ \frac{\partial l^-}{\partial y_{i,j}} &= \frac{1}{1 - y_{i,j}} \end{aligned} \quad (6)$$

Let $x_{i,j} = P_{i,j} - T_{i,j}$. The DB function can be expressed as $f(x) = \frac{1}{1+e^{-kx_{i,j}}}$. Similarly, the loss l_b^+ for positive labels and l_b^- for negative labels are:

$$\begin{aligned} l_b^+ &= -\log \frac{1}{1 + e^{-kx_{i,j}}} \\ l_b^- &= -\log(1 - \frac{1}{1 + e^{-kx_{i,j}}}) \end{aligned} \quad (7)$$

The differential of the losses with the DB function are as follows:

$$\begin{aligned} \frac{\partial l_b^+}{\partial x_{i,j}} &= \frac{-ke^{-kx_{i,j}}}{1 + e^{-kx_{i,j}}} \\ \frac{\partial l_b^-}{\partial x_{i,j}} &= \frac{k}{1 + e^{-kx_{i,j}}} \end{aligned} \quad (8)$$

The numerical comparison of the derivatives of the losses in Eq. 6 and Eq. 8 are also shown in Fig. 5b and Fig. 5c respectively, from which we can perceive:

(1) The magnitude of the differential around the boundary value. For the standard binary cross-entropy loss with logits (top), the magnitudes of $\frac{\partial l^+}{\partial y_{i,j}}$ and $\frac{\partial l^-}{\partial y_{i,j}}$ are quite small around the boundary value (0.5) between positive value (> 0.5) and negative value (< 0.5). As a result, the backpropagation or the feedback may not be significant when a predicted value is ambiguous, such as 0.4 or 0.6; For the binary cross-entropy with differentiable binarization (bottom), the magnitudes of $\frac{\partial l_b^+}{\partial x_{i,j}}$ and $\frac{\partial l_b^-}{\partial x_{i,j}}$ are large around the boundary value (0) between positive value (> 0) and negative value (< 0), where is augmented by the amplifying factor k. Thus, the proposed differentiable binarization helps to produce more distinctive predictions around the boundary value.

(2) The least upper bound and the greatest lower bound. For the standard binary cross-entropy loss with logits (top), there is no greatest lower bound for $\frac{\partial l^+}{\partial y_{i,j}}$ and no least upper bound for $\frac{\partial l^-}{\partial y_{i,j}}$; For the binary cross-entropy with differentiable binarization (bottom), the greatest lower bound of $\frac{\partial l_b^+}{\partial x_{i,j}}$ and the least upper bound of $\frac{\partial l_b^-}{\partial x_{i,j}}$ is determined by the amplifying factor k. Thus, the proposed differentiable binarization tends to work more stable for some extremely small or large values.

Considering Activation Function In practice, the activation function (Sigmoid) is applied in both formulations, which could bound the derivatives and alleviate the problem of "The least upper bound and the greatest lower bound". Considering the Sigmoid function as follows:

$$\begin{aligned} y_{i,j} &= \frac{1}{1 + e^{-v_{i,j}}} \\ x_{i,j} &= \frac{1}{1 + e^{-v_{i,j}^{db}}} - T_{i,j}, \end{aligned} \quad (9)$$

where $-v_{i,j}$ and $v_{i,j}^{db}$ indicate the output values before the Sigmoid activations; $T_{i,j}$ indicates the adaptive threshold. In this way, the differential of the losses can be updated by

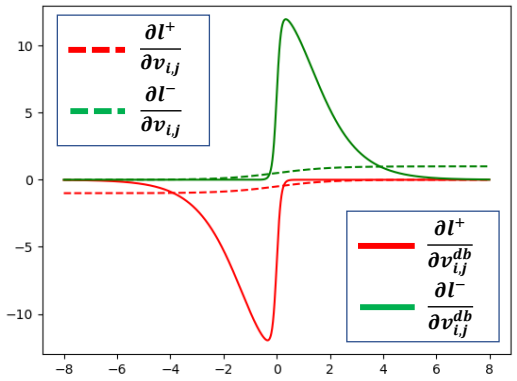


Fig. 6: The derivative of losses in Eq. 10 and Eq. 11.

chain rule as follows:

$$\begin{aligned}\frac{\partial l^+}{\partial v_{i,j}} &= \frac{-e^{-v_{i,j}}}{1+e^{-v_{i,j}}} \\ \frac{\partial l^-}{\partial v_{i,j}} &= \frac{1}{1+e^{-v_{i,j}}}\end{aligned} \tag{10}$$

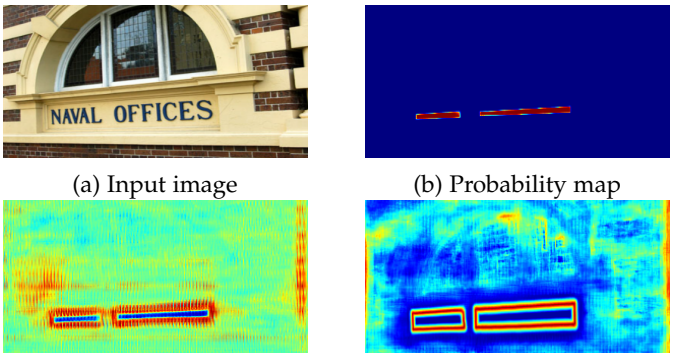
$$\begin{aligned}\frac{\partial l_b^+}{\partial v_{i,j}^{db}} &= \frac{-ke^{-k(\frac{1}{1+e^{-v_{i,j}^{db}}}-T_{i,j})-v_{i,j}^{db}}}{(1+e^{-kv_{i,j}^{db}})^2(1+e^{-k(\frac{1}{1+e^{-v_{i,j}^{db}}}-T_{i,j})})} \\ \frac{\partial l_b^-}{\partial v_{i,j}^{db}} &= \frac{k}{1+e^{-kv_{i,j}^{db}}}\end{aligned} \tag{11}$$

Without loss of generality, Eq. 10 and Eq. 11 can be visualized as Fig. 6, by setting $k = 50$ and $T_{i,j} = 0.5$.

We can perceive from Fig. 6 that the differentiable binarization enlarges the feedback of backpropagation when the wrongly predicted values are near the boundary value. Thus, the proposed differentiable binarization makes the model focus on optimizing the prediction of ambiguous regions. Besides, the Sigmoid function alleviates the problem of "The least upper bound and the greatest lower bound" and DB further decreases the penalty for extremely small/large values.

3.4 Adaptive Threshold

The threshold map in Fig. 1 is similar to the text border map in [56] from appearance. However, the motivation and usage of the threshold map are different from the text border map. The threshold map with/without supervision is visualized in Fig. 7. The threshold map would highlight the text border region even without supervision for the threshold map. This indicates that the border-like threshold map is beneficial to the final results. Thus, we apply border-like supervision on the threshold map for better guidance. An ablation study about the supervision of the adaptive threshold is discussed in the Experiments section. For the usage, the text border map in [56] is used to split the text instances while our threshold map is served as thresholds for the binarization.



(a) Input image (b) Probability map

(c) Threshold map without supervision (d) Threshold map with supervision

Fig. 7: The threshold map with/without supervision.

3.5 Deformable Convolution

Deformable convolution [9], [66] can provide a flexible receptive field for the model, which is especially beneficial to the text instances of extreme aspect ratios. Following [66], modulated deformable convolutions are applied in all the 3×3 convolutional layers in stages conv3, conv4, and conv5 in the ResNet-18 or ResNet-50 backbone [14].

3.6 Label Generation

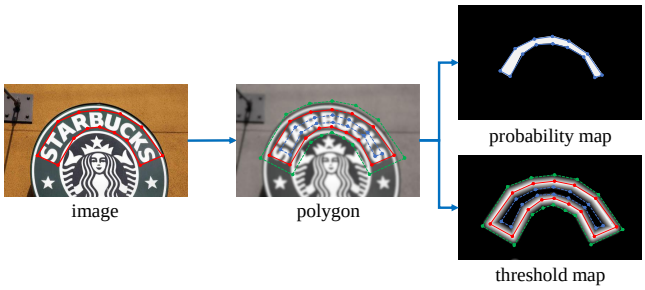


Fig. 8: Label generation. The annotation of text polygon is visualized in red lines. The shrunk and dilated polygon are displayed in blue and green lines, respectively.

The label generation for the probability map is inspired by PSENet [49]. Given a text image, each polygon of its text regions is described by a set of segments:

$$G = \{S_k\}_{k=1}^{n} \tag{12}$$

n is the number of vertexes, which may be different in different datasets, e.g, 4 for the ICDAR 2015 dataset [22] and 16 for the CTW1500 dataset [32]. Then the positive area is generated by shrinking the polygon G to G_s using the Vatti clipping algorithm [47]. The offset D of shrinking is computed from the perimeter L and area A of the original polygon:

$$D = \frac{A(1-r^2)}{L} \tag{13}$$

where r is the shrink ratio, set to 0.4 empirically.

With a similar procedure, we can generate labels for the threshold map. Firstly the text polygon G is dilated with the same offset D to G_d. We consider the gap between G_s and G_d as the border of the text regions, where the label of the

threshold map can be generated by computing the distance to the closest segment in G.

3.7 Optimization

The loss function L can be expressed as a weighted sum of the loss for the probability map L_s, the loss for the binary map L_b, and the loss for the threshold map L_t:

$$L = L_s + \alpha \times L_b + \beta \times L_t \tag{14}$$

According to the numeric values of the losses, α and β are set to 1.0 and 10 respectively.

We apply a binary cross-entropy (BCE) loss for both L_s and L_b. To overcome the unbalance of the number of positives and negatives, hard negative mining is used in the BCE loss by sampling the hard negatives.

$$L_s = L_b = \sum_{i \in S_l} y_i \log x_i + (1 - y_i) \log (1 - x_i) \tag{15}$$

S_l is the sampled set where the ratio of positives and negatives is $1 : 3$. It consists of all the positives and the top-k negatives (sorted by the values of the predicting probability), where k is 3 times the number of positives.

L_t is computed as the sum of $L1$ distances between the prediction and label inside the dilated text polygon G_d:

$$L_t = \sum_{i \in R_d} |y_i^* - x_i^*| \tag{16}$$

where R_d is a set of indexes of the pixels inside the dilated polygon G_d; y^* is the label for the threshold map.

In the inference period, we can either use the probability map or the approximate binary map to generate text bounding boxes, which produces almost the same results. For better efficiency, we use the probability map so that the threshold branch can be removed. The box formation process consists of three steps: (1) the probability map/the approximate binary map is firstly binarized with a constant threshold (0.2), to get the binary map; (2)the connected regions (shrunk text regions) are obtained from the binary map; (3) the shrunk regions are dilated with an offset D' the Vatti clipping algorithm [47]. D' is calculated as

$$D' = \frac{A' \times r'}{L'} \tag{17}$$

where A' is the area of the shrunk polygon; L' is the perimeter of the shrunk polygon; r' is set to 1.5 empirically.

4 Experiments

4.1 Datasets

The scene text datasets used in the experiments are described as follows.

SynthText [12] is a synthetic dataset which consists of $800k$ images. These images are synthesized from $8k$ background images. This dataset is only used to pre-train our model.

MLT-2017 dataset [1] is a multi-language dataset. It includes 9 languages representing 6 different scripts. There are $7,200$ training images, $1,800$ validation images, and $9,000$ testing images in this dataset. We use both the training set and the validation set in the finetune period.

MLT-2019 dataset [2] is a multi-language dataset. It is an extension of MLT-2017. It includes 10 languages representing 7 different scripts. The languages include Chinese, Japanese, Korean, English, French, Arabic, Italian, German, Bangla, and Hindi (Devanagari). There are $10,000$ training images, $2,000$ validation images, and $10,000$ testing images in this dataset. We use the training set in the finetune period.

ICDAR 2015 dataset [22] consists of $1,000$ training images and 500 testing images, which are captured by Google glasses with a resolution of 720×1280. The text instances are labeled at the word level.

MSRA-TD500 dataset [59] is a multi-language dataset that includes English and Chinese. There are 300 training images and 200 testing images. The text instances are labeled at the text-line level. Following the previous methods [37], [39], [65], we include extra 400 training images from HUST-TR400 [58].

CTW1500 dataset [32] mainly focuses on curved text. It consists of $1,000$ training images and 500 testing images. The text instances are annotated at the text-line level.

Total-Text dataset [8] includes the text of various shapes, including horizontal, multi-oriented, and curved. They are $1,255$ training images and 300 testing images. The text instances are labeled at the word level.

4.2 Implementation Details

For all the models, we first pre-train them with the SynthText dataset for $100k$ iterations. Then, we finetune the models on the corresponding real-world datasets for 1200 epochs. The training batch size is set to 16. We follow a "poly" learning rate policy where the learning rate at the current iteration equals the initial learning rate multiplying $(1 - \frac{iter}{max\_iter})^{power}$, where the initial learning rate is set to 0.007 and $power$ is 0.9. We use a weight decay of 0.0001 and a momentum of 0.9. The max_iter means the maximum number of iterations, which depends on the maximum epochs.

The data augmentation for the training data includes: (1) Random rotation with an angle range of $(-10°, 10°)$; (2) Random cropping; (3) Random Flipping. All the processed images are re-sized to 640×640 for better training efficiency.

In the inference period, we keep the aspect ratio of the test images and re-size the input images by setting a suitable height for each dataset. The inference speed is tested with a batch size of 1, with a single GTX 1080Ti GPU. The inference time cost consists of the model forward time cost and the post-processing time cost. The post-processing time cost is about 30% of the inference time.

4.3 Ablation Study

We conduct an ablation study on the MSRA-TD500 dataset and the CTW1500 dataset to show the effectiveness of the modules including differentiable binarization, deformable convolution, and adaptive scale fusion. The detailed experimental results are shown in Tab. 1, Tab. 2, and Tab. 3.

1. https://rrc.cvc.uab.es/?ch=8

2. https://rrc.cvc.uab.es/?ch=15

Fig. 9: Some visualization results on text instances of various shapes, including curved text, multi-oriented text, vertical text, and long text lines. For each unit, the top right is the threshold map; the bottom right is the probability map.

TABLE 1: Detection results with different settings of deformable convolution, differentiable binarization and adaptive scale fusion module. "DConv" indicates deformable convolution. "ASF" indicates adaptive scale fusion module. "P", "R", and "F" indicate precision, recall, and f-measure respectively.

Backbone	DConv	DB	ASF	MSRA-TD500				CTW1500			
				P	R	F	FPS	P	R	F	FPS
ResNet-18	×	×	×	85.5	70.8	77.4	**66**	76.3	72.8	74.5	**59**
ResNet-18	✓	×	×	86.8	72.3	78.9	62	80.9	75.4	78.1	55
ResNet-18	×	✓	×	87.3	75.8	81.1	66	82.4	76.6	79.4	59
ResNet-18	×	×	✓	84.9	78.5	81.6	53	83.5	75.9	79.5	45
ResNet-18	✓	✓	×	**90.4**	76.3	82.8	62	84.8	77.5	81.0	55
ResNet-18	✓	×	✓	87.1	79.9	83.3	55	86.4	80.8	83.5	40
ResNet-18	✓	✓	✓	87.9	**82.5**	**85.1**	55	**86.7**	**81.3**	**83.9**	40
ResNet-50	×	×	×	84.6	73.5	78.7	**40**	81.6	72.9	77.0	**27**
ResNet-50	✓	×	×	90.5	77.9	83.7	32	86.2	78.0	81.9	22
ResNet-50	×	✓	×	86.6	77.7	81.9	40	84.3	79.1	81.6	27
ResNet-50	×	×	✓	84.5	83.2	83.8	32	83.3	79.1	81.2	24
ResNet-50	✓	✓	×	91.5	79.2	84.9	32	86.9	80.2	83.4	22
ResNet-50	✓	×	✓	90.7	**83.5**	86.9	29	89.2	81.4	85.1	21
ResNet-50	✓	✓	✓	**91.5**	83.3	**87.2**	29	**87.9**	**82.8**	**85.3**	21

Differentiable Binarization In Tab. 1, we can see that our proposed DB improves the performance significantly for both ResNet-18 and ResNet-50 on the two datasets. For the ResNet-18 backbone, DB achieves 3.7% and 4.9% performance gain in terms of F-measure on the MSRA-TD500 dataset and the CTW1500 dataset. For the ResNet-50 backbone, DB brings 3.2% (on the MSRA-TD500 dataset) and 4.6% (on the CTW1500 dataset) improvements. Moreover, since DB can be removed in the inference period, the speed is the same as the one without DB.

Deformable Convolution As shown in Tab. 1, the deformable convolution can also brings $1.5\% - 5.0\%$ performance gain since it provides a flexible receptive field for the backbone, with small extra time costs. For the MSRA-TD500 dataset, the deformable convolution increase the F-measure by 1.5% (with ResNet-18) and 5.0% (with ResNet-50). For the CTW1500 dataset, 3.6% (with ResNet-18) and 4.9% (with ResNet-50) improvements are achieved by the deformable convolution.

Backbone The proposed detector with the ResNet-50 backbone achieves better performance than the ResNet-18 but runs slower. Specifically, The best ResNet-50 model outperforms the best ResNet-18 model by 2.1% (on the MSRA-TD500 dataset) and 2.4% (on the CTW1500 dataset), with approximate double time cost.

Supervision of Threshold Map Although the threshold maps with/without supervision are similar in appearance, the supervision can bring performance gain. As shown in Tab. 3, the supervision improves 0.7% (ResNet-18) and 2.6% (ResNet-50) on the MLT-2017 dataset.

Adaptive Scale Fusion As shown in Tab. 1, the adaptive scale fusion module improves the F-measure by 2.3% and 1.9% on the MSRA-TD500 dataset and the CTW1500 dataset, respectively. The inference speed decreases slightly. As shown in Tab. 2, the spatial attention in the ASF module brings 0.5% and 1.0% on the MSRA-TD500 dataset and the

TABLE 2: Detection results with different settings of ASF. "Spatial" means spatial attention in the adaptive scale fusion module; "Scale" means adaptive scale fusion.

Base Method	Scale	Spatial	MSRA-TD500				CTW1500			
			P	R	F	FPS	P	R	F	FPS
DBNet (ResNet-50)	×	×	91.5	79.2	84.9	32	86.9	80.2	83.4	22
DBNet (ResNet-50)	✓	×	92.2	81.8	86.7	30	85.4	83.2	84.3	21
DBNet (ResNet-50)	✓	✓	91.5	83.3	87.2	29	87.9	82.8	85.3	21

Fig. 10: Some visualization results of DBNet and DBNet++ on text instances of various shapes, including curved text, vertical text and multi-oriented text. For each unit, the top is the result of DBNet; the bottom is the result of DBNet++. More results are shown in the supplementary.

TABLE 3: Effect of supervising the threshold map on the MLT-2017 dataset. "Thr-Sup" denotes applying supervision on the threshold map.

Backbone	Thr-Sup	P	R	F	FPS
ResNet-18	×	81.3	63.1	71.0	41
ResNet-18	✓	**81.9**	**63.8**	**71.7**	41
ResNet-50	×	81.5	64.6	72.1	19
ResNet-50	✓	**83.1**	**67.9**	**74.7**	19

CTW1500 dataset by providing more flexible and adaptive attention weights across the spatial dimension.

Comparisons with PPM and CCA We compare our proposed ASF with a multi-scale feature fusion module (Pyramid Pooling Module, PPM [64]) and a context enhancement module (Criss-Cross Attention, CCA [20]). We integrate them into the encoder of the DBNet for a fair comparison. As shown in Tab. 4, the proposed ASF outperforms the PPM and CCA in terms of both the detection accuracy and the inference speed. The experimental results demonstrate that the proposed ASF is more effective than PPM and CCA.

Our proposed ASF performs better than PPM and CCA on the text detection task because that ASF is not designed to simply enlarge the receptive field or introduce more context for the segmentation model. It fuses the multi-scale feature maps with stage-wise and spatial-wise attention weights, which can be guided by the scales of the corresponding text regions.

4.4 Comparisons with Previous Methods

We compare our proposed method with previous methods on five standard benchmarks, including two benchmarks for curved text, one benchmark for multi-oriented text, and two multi-language benchmarks for long text lines. Some qualitative results are visualized in Fig. 9.

Curved Text Detection We prove the shape robustness of our method on two curved text benchmarks (Total-Text and CTW1500). As shown in Tab. 5 and Tab. 6, our method achieves state-of-the-art performance both on accuracy and speed. Specifically, "DBNet++ (ResNet-50)" outperforms the previous state-of-the-art method by 1.0% and 1.6% on the Total-Text and the CTW1500 dataset. "DBNet (ResNet-50)" runs faster than all previous methods and the speed can be further improved by using a ResNet-18 backbone, with a small performance drop. Compared to the recent fast text detector [50], DBNet++ achieves better accuracy with a comparable inference speed.

Multi-Oriented Text Detection The ICDAR 2015 dataset is a multi-oriented text dataset that contains lots of small

TABLE 4: Comparisons with multi-scale feature fusion and context enhancement modules in semantic segmentation methods. "PPM": Pyramid Pooling Module; "CCA": Criss-Cross Attention.

Base Method	Module	MSRA-TD500				CTW1500			
		P	R	F	FPS	P	R	F	FPS
DBNet (ResNet-50)	PPM [64]	91.2	79.7	85.1	23	87.0	79.9	83.3	16
DBNet (ResNet-50)	CCA [20]	92.9	80.9	86.5	22	88.4	81.5	84.8	15
DBNet (ResNet-50)	ASF(ours)	91.5	83.3	**87.2**	**29**	87.9	82.8	**85.3**	**21**

TABLE 5: Detection results on the Total-Text dataset. The values in the bracket mean the height of the input images. "*" indicates testing with multiple scales.

Method	P	R	F	FPS
TextSnake [37]	82.7	74.5	78.4	-
ATRR [51]	80.9	76.2	78.5	-
Mask TextSpotter [38]	82.5	75.6	78.6	-
TextField [55]	81.2	79.9	80.6	-
LOMO [61]*	87.6	79.3	83.3	-
CRAFT [1]	87.6	79.9	83.6	-
CSE [34]	81.4	79.1	80.2	-
PSENet-1s [49]	84.0	78.0	80.9	3.9
PAN [50]	**89.3**	81.0	85.0	39.6
DBNet (ResNet-18) (800) [26]	88.3	77.9	82.8	**50**
DBNet (ResNet-50) (800) [26]	87.1	82.5	84.7	32
DBNet++ (ResNet-18) (800)	87.4	79.6	83.3	48
DBNet++ (ResNet-50) (800)	88.9	**83.2**	**86.0**	28

TABLE 6: Detection results on the CTW1500 dataset. The methods with "*" are collected from [32]. The values in the bracket mean the height of the input images.

Method	P	R	F	FPS
CTPN*	60.4	53.8	56.9	7.14
EAST*	78.7	49.1	60.4	21.2
SegLink*	42.3	40.0	40.8	10.7
TextSnake [37]	67.9	**85.3**	75.6	1.1
TLOC [32]	77.4	69.8	73.4	13.3
PSENet-1s [49]	84.8	79.7	82.2	3.9
SAE [46]	82.7	77.8	80.1	3
PAN [50]	86.4	81.2	83.7	39.8
DBNet (ResNet-18) (1024) [26]	84.8	77.5	81.0	**55**
DBNet (ResNet-50) (1024) [26]	86.9	80.2	83.4	22
DBNet++ (ResNet-18) (1024)	86.7	81.3	83.9	40
DBNet++ (ResNet-18) (800)	84.3	81.0	82.6	49
DBNet++ (ResNet-50) (1024)	**88.5**	82.0	85.1	21
DBNet++ (ResNet-50) (800)	87.9	82.8	**85.3**	26

and low-resolution text instances. In Tab. 7, we can see that "DBNet++ (ResNet-50) (1152)" and "DBNet (ResNet-50) (1152)" achieve the state-of-the-art performance on accuracy. Compared to EAST [65], "DBNet++ (ResNet-50) (736)" outperforms it by 7.2% on accuracy and runs twice faster. Compared to PAN [50], "DBNet++ (ResNet-18) (736)" performs better in terms of accuracy and inference speed.

Multi-Language Text Detection Our method is robust on multi-language text detection. As shown in Tab. 8 and Tab. 9, "DBNet++ (ResNet-50)" is superior to previous methods on accuracy and speed. For the accuracy, "DBNet++ (ResNet-50)" surpasses the previous state-of-the-art method by 3.1% and 3.3% on the MSRA-TD500 dataset and the MLT-2019 dataset respectively. For the speed, "DBNet++ (ResNet-18) (736)" is faster than the previous fastest method [50] while achieving better accuracy on the MSRA-TD500 dataset. The speed can be further accelerated to 80 FPS ("DBNet++ (ResNet-18) (512)") by decreasing the input size.

TABLE 7: Detection results on the ICDAR 2015 dataset. The values in the bracket mean the height of the input images.

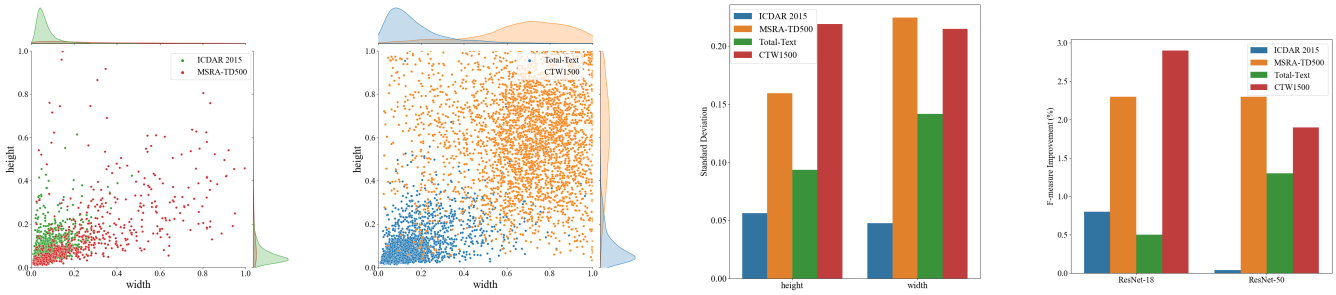
Method	P	R	F	FPS
CTPN [45]	74.2	51.6	60.9	7.1
EAST [65]	83.6	73.5	78.2	13.2
SSTD [15]	80.2	73.9	76.9	7.7
WordSup [18]	79.3	77	78.2	-
Lyu *et al.* [39]	**94.1**	70.7	80.7	3.6
TextBoxes++ [24]	87.2	76.7	81.7	11.6
RRD [27]	85.6	79	82.2	6.5
MCN [33]	72	80	76	-
TextSnake [37]	84.9	80.4	82.6	1.1
PSENet-1s [49]	86.9	84.5	85.7	1.6
SPCNet [53]	88.7	**85.8**	87.2	-
LOMO [61]	91.3	83.5	87.2	-
CDAFT [1]	89.8	84.3	86.9	-
SAE(720) [46]	85.1	84.5	84.8	3
SAE(990) [46]	88.3	85.0	86.6	-
PAN [50]	84.0	81.9	82.9	26.1
DBNet (ResNet-18) (736) [26]	86.8	78.4	82.3	**48**
DBNet (ResNet-50) (1152) [26]	91.8	83.2	**87.3**	12
DBNet++ (ResNet-18) (736)	90.1	77.2	83.1	44
DBNet++ (ResNet-50) (1152)	90.9	83.9	**87.3**	10

TABLE 8: Detection results on the MSRA-TD500 dataset. The values in the bracket mean the height of the input images.

Method	P	R	F	FPS
He *et al.* [16]	71	61	69	-
DeepReg [17]	77	70	74	1.1
RRPN [40]	82	68	74	-
RRD [27]	87	73	79	10
MCN [33]	88	79	83	-
PixelLink [10]	83	73.2	77.8	3
Lyu *et al.* [39]	87.6	76.2	81.5	5.7
TextSnake [37]	83.2	73.9	78.3	1.1
Xue *et al.* [56]	83.0	77.4	80.1	-
MSR [57]	87.4	76.7	81.7	-
CRAFT [1]	88.2	78.2	82.9	8.6
SAE [46]	84.2	81.7	82.9	-
PAN [50]	84.4	**83.8**	84.1	30.2
DBNet (ResNet-18) (512) [26]	85.7	73.2	79.0	**82**
DBNet (ResNet-18) (736) [26]	90.4	76.3	82.8	62
DBNet (ResNet-50) (736) [26]	**91.5**	79.2	84.9	32
DBNet++ (ResNet-18) (512)	89.7	76.5	82.6	80
DBNet++ (ResNet-18) (736)	87.9	82.5	85.1	55
DBNet++ (ResNet-50) (736)	**91.5**	83.3	**87.2**	29

4.5 Comparisons with the Conference Version

The major extension of this paper over the conference version is the proposed ASF module. Some qualitative results



(a) ICDAR 2015 dataset and MSRA-TD500 dataset

(b) Total-Text dataset and CTW1500 dataset

(c) The standard deviation of height and width

(d) The improvements obtained by ASF

Fig. 11: The scale distributions of the test sets of different datasets. The points in (a) and (b) represent the text instances of various scales. The scales of the bounding boxes are measured by the width and height of their minimum bounding rectangles.

TABLE 9: Detection results on the MLT-2019 dataset. *CRAFTS used character-level annotations and integrated a recognition model.

Method	P	R	F	FPS
PSENet [49]	73.5	59.6	65.8	-
CRAFTS* [2]	79.5	59.6	68.1	-
DBNet (ResNet-18) [26]	75.3	60.2	66.9	**19**
DBNet (ResNet-50) [26]	78.3	64.0	70.4	10
DBNet++ (ResNet-18)	77.5	61.0	68.2	18
DBNet++ (ResNet-50)	**78.6**	**65.4**	**71.4**	10

are shown in Fig. 10 and more results are shown in the supplementary. As shown, DBNet++ performs better in detecting the text instances of diverse scales, especially for the large-scale text instances. In contrast, DBNet may generate inaccurate bounding boxes or discrete bounding boxes for large-scale text instances. This indicates that the proposed ASF module strengthens the scale robustness of the text detection model.

The quantitative results in the standard scene text benchmarks show that the proposed DBNet++ outperforms the conference version in terms of accuracy with little speed drop. Specifically, The accuracy increases 0.5% (1.3%), 2.9% (1.9%), 0.8% (0.0%), 3.6% (2.3%), and 1.3% (1.0%) in terms of F-measure on the Total-Text dataset, the CTW1500 dataset, the ICDAR 2015 dataset, the MSRA-TD500 dataset, and the MLT-2019 dataset, respectively, with the backbone of ResNet-18 (ResNet-50).

The performance improvements on the CTW1500 dataset and the MSRA-TD500 dataset are more significant than those on the Total-Text dataset and the ICDAR 2015 dataset. Thus, we visualize the scale distributions of these datasets in Fig. 11 for further analysis. As shown in Fig. 11, the scale distributions of the ICDAR 2015 dataset and the Total-Text dataset are less diverse than those of the MSRA-TD500 dataset and the CTW1500 dataset. As shown in Fig. 11 (c) and Fig. 11 (d), the performance improvements approximately have a positive correlation with the diversity of the scales, which quantitatively reflects that DBNet++ is superior to DBNet on scale robustness.

4.6 Limitation

One limitation of our method is that it is difficult to deal with cases "text inside text", which means that a text instance is inside another text instance. Although the shrunk text regions are helpful to the cases that the text instance is not in the center region of another text instance, it fails when the text instance is exactly located in the center region of another text instance. This is a common limitation for segmentation-based scene text detectors.

5 Conclusion

In this paper, we have presented a novel framework for detecting arbitrary-shape scene text, which improves the segmentation-based scene text detection methods from two aspects: (1) A differentiable binarization module is proposed to integrate the binarization process into the training period; (2) The proposed ASF module efficiently enhances the scale robustness of the segmentation network. Both two modules significantly improve the text detection accuracy. The experiments have verified that our method (ResNet-50 backbone) consistently outperforms the state-the-the-art methods on five standard scene text benchmarks, in terms of speed and accuracy. In particular, even with a lightweight backbone (ResNet-18), our method can achieve competitive performance on all the testing datasets with real-time inference speed.

Acknowledgments

This work was supported by National Key R&D Program of China No.2018YFB1004600 and NSFC No.61733007.

References

[1] Y. Baek, B. Lee, D. Han, S. Yun, and H. Lee. Character region awareness for text detection. In *Proc. Conf. Comput. Vision Pattern Recognition*, pages 9365–9374, 2019.

[2] Y. Baek, S. Shin, J. Baek, S. Park, J. Lee, D. Nam, and H. Lee. Character region attention for text spotting. In *European Conf. Comput. Vision*, pages 504–521, 2020.

[3] Y. Cao, J. Xu, S. Lin, F. Wei, and H. Hu. Gcnet: Non-local networks meet squeeze-excitation networks and beyond. In *Proc. Int. Conf. Comput. Vision Workshops*, pages 0–0, 2019.

[4] L. Chen, G. Papandreou, F. Schroff, and H. Adam. Rethinking atrous convolution for semantic image segmentation. *CoRR*, abs/1706.05587, 2017.

[5] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *European Conf. Comput. Vision*, pages 801–818, 2018.

[6] B. Cheng, L.-C. Chen, Y. Wei, Y. Zhu, Z. Huang, J. Xiong, T. S. Huang, W.-M. Hwu, and H. Shi. Spgnet: Semantic prediction guidance for scene parsing. In *Proc. Int. Conf. Comput. Vision*, pages 5218–5228, 2019.

[7] S. Choi, J. T. Kim, and J. Choo. Cars can't fly up in the sky: Improving urban-scene segmentation via height-driven attention networks. In *Proc. Conf. Comput. Vision Pattern Recognition*, pages 9373–9383, 2020.

[8] C. K. Ch'ng and C. S. Chan. Total-text: A comprehensive dataset for scene text detection and recognition. In *Proc. Int. Conf. on Document Analysis and Recognition*, pages 935–942, 2017.

[9] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei. Deformable convolutional networks. In *Proc. Int. Conf. Comput. Vision*, pages 764–773, 2017.

[10] D. Deng, H. Liu, X. Li, and D. Cai. Pixellink: Detecting scene text via instance segmentation. In *AAAI Conf. on Artificial Intelligence*, 2018.

[11] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu. Dual attention network for scene segmentation. In *Proc. Conf. Comput. Vision Pattern Recognition*, pages 3146–3154, 2019.

[12] A. Gupta, A. Vedaldi, and A. Zisserman. Synthetic data for text localisation in natural images. In *Proc. Conf. Comput. Vision Pattern Recognition*, 2016.

[13] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. In *Proc. Int. Conf. Comput. Vision*, pages 2961–2969, 2017.

[14] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. Conf. Comput. Vision Pattern Recognition*, pages 770–778, 2016.

[15] P. He, W. Huang, T. He, Q. Zhu, Y. Qiao, and X. Li. Single shot text detector with regional attention. In *Proc. Int. Conf. Comput. Vision*, pages 3047–3055, 2017.

[16] T. He, W. Huang, Y. Qiao, and J. Yao. Text-attentional convolutional neural network for scene text detection. *IEEE Trans. Image Processing*, 25(6):2529–2541, 2016.

[17] W. He, X. Zhang, F. Yin, and C. Liu. Deep direct regression for multi-oriented scene text detection. In *Proc. Int. Conf. Comput. Vision*, 2017.

[18] H. Hu, C. Zhang, Y. Luo, Y. Wang, J. Han, and E. Ding. Wordsup: Exploiting word annotations for character based text detection. In *Proc. Int. Conf. Comput. Vision*, pages 4940–4949, 2017.

[19] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. In *Proc. Conf. Comput. Vision Pattern Recognition*, pages 7132–7141, 2018.

[20] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, and W. Liu. Ccnet: Criss-cross attention for semantic segmentation. In *Proc. Int. Conf. Comput. Vision*, 2019.

[21] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman. Reading text in the wild with convolutional neural networks. *Int. J. Comput. Vision*, 116(1):1–20, 2016.

[22] D. Karatzas, L. Gomez-Bigorda, A. Nicolaou, S. K. Ghosh, A. D. Bagdanov, M. Iwamura, J. Matas, L. Neumann, V. R. Chandrasekhar, S. Lu, F. Shafait, S. Uchida, and E. Valveny. ICDAR 2015 competition on robust reading. In *Proc. Int. Conf. on Document Analysis and Recognition*, 2015.

[23] M. Liao, P. Lyu, M. He, C. Yao, W. Wu, and X. Bai. Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2019.

[24] M. Liao, B. Shi, and X. Bai. Textboxes++: A single-shot oriented scene text detector. *IEEE Trans. Image Processing*, 27(8):3676–3690, 2018.

[25] M. Liao, B. Shi, X. Bai, X. Wang, and W. Liu. Textboxes: A fast text detector with a single deep neural network. In *AAAI Conf. on Artificial Intelligence*, 2017.

[26] 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*, pages 11474–11481, 2020.

[27] M. Liao, Z. Zhu, B. Shi, G. Xia, and X. Bai. Rotation-sensitive regression for oriented scene text detection. In *Proc. Conf. Comput. Vision Pattern Recognition*, pages 5909–5918, 2018.

[28] G. Lin, A. Milan, C. Shen, and I. Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *Proc. Conf. Comput. Vision Pattern Recognition*, pages 1925–1934, 2017.

[29] T. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie. Feature pyramid networks for object detection. In *Proc. Conf. Comput. Vision Pattern Recognition*, 2017.

[30] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, and S. E. Reed. SSD: single shot multibox detector. In *European Conf. Comput. Vision*, 2016.

[31] Y. Liu and L. Jin. Deep matching prior network: Toward tighter multi-oriented text detection. In *Proc. Conf. Comput. Vision Pattern Recognition*, 2017.

[32] Y. Liu, L. Jin, S. Zhang, C. Luo, and S. Zhang. Curved scene text detection via transverse and longitudinal sequence connection. *Pattern Recognition*, 90:337–345, 2019.

[33] Z. Liu, G. Lin, S. Yang, J. Feng, W. Lin, and W. L. Goh. Learning markov clustering networks for scene text detection. In *Proc. Conf. Comput. Vision Pattern Recognition*, pages 6936–6944, 2018.

[34] Z. Liu, G. Lin, S. Yang, F. Liu, W. Lin, and W. L. Goh. Towards robust curve text detection with conditional spatial expansion. In *Proc. Conf. Comput. Vision Pattern Recognition*, pages 7269–7278, 2019.

[35] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proc. Conf. Comput. Vision Pattern Recognition*, pages 3431–3440, 2015.

[36] S. Long, X. He, and C. Yao. Scene text detection and recognition: The deep learning era. *Int. J. Comput. Vision*, pages 1–24, 2020.

[37] S. Long, J. Ruan, W. Zhang, X. He, W. Wu, and C. Yao. Textsnake: A flexible representation for detecting text of arbitrary shapes. In *European Conf. Comput. Vision*, pages 20–36, 2018.

[38] 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. In *European Conf. Comput. Vision*, pages 67–83, 2018.

[39] P. Lyu, C. Yao, W. Wu, S. Yan, and X. Bai. Multi-oriented scene text detection via corner localization and region segmentation. In *Proc. Conf. Comput. Vision Pattern Recognition*, pages 7553–7563, 2018.

[40] J. Ma, W. Shao, H. Ye, L. Wang, H. Wang, Y. Zheng, and X. Xue. Arbitrary-oriented scene text detection via rotation proposals. *IEEE Trans. on Multimedia*, 20(11):3111–3122, 2018.

[41] L. Neumann and J. Matas. Real-time scene text localization and recognition. In *Proc. Conf. Comput. Vision Pattern Recognition*, pages 3538–3545, 2012.

[42] 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*, pages 234–241, 2015.

[43] B. Shi, X. Bai, and S. J. Belongie. Detecting oriented text in natural images by linking segments. In *Proc. Conf. Comput. Vision Pattern Recognition*, 2017.

[44] J. Tang, Z. Yang, Y. Wang, Q. Zheng, Y. Xu, and X. Bai. Seglink++: Detecting dense and arbitrary-shaped scene text by instance-aware component grouping. *Pattern recognition*, 96:106954, 2019.

[45] Z. Tian, W. Huang, T. He, P. He, and Y. Qiao. Detecting text in natural image with connectionist text proposal network. In *European Conf. Comput. Vision*, 2016.

[46] Z. Tian, M. Shu, P. Lyu, R. Li, C. Zhou, X. Shen, and J. Jia. Learning shape-aware embedding for scene text detection. In *Proc. Conf. Comput. Vision Pattern Recognition*, pages 4234–4243, 2019.

[47] B. R. Vati. A generic solution to polygon clipping. *Communications of the ACM*, 35(7):56–64, 1992.

[48] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang. Residual attention network for image classification. In *Proc. Conf. Comput. Vision Pattern Recognition*, pages 3156–3164, 2017.

[49] W. Wang, E. Xie, X. Li, W. Hou, T. Lu, G. Yu, and S. Shao. Shape robust text detection with progressive scale expansion network. In *Proc. Conf. Comput. Vision Pattern Recognition*, pages 9336–9345, 2019.

[50] W. Wang, E. Xie, X. Song, Y. Zang, W. Wang, T. Lu, G. Yu, and C. Shen. Efficient and accurate arbitrary-shaped text detection with pixel aggregation network. In *Proc. Int. Conf. Comput. Vision*, pages 8440–8449, 2019.

[51] X. Wang, Y. Jiang, Z. Luo, C.-L. Liu, H. Choi, and S. Kim. Arbitrary shape scene text detection with adaptive text region representation. In *Proc. Conf. Comput. Vision Pattern Recognition*, pages 6449–6458, 2019.

[52] S. Woo, J. Park, J. Lee, and I. S. Kweon. Cbam: Convolutional block attention module. In *European Conf. Comput. Vision*, volume 11211, pages 3–19, 2018.

[53] E. Xie, Y. Zang, S. Shao, G. Yu, C. Yao, and G. Li. Scene text detection with supervised pyramid context network. In *AAAI Conf. on Artificial Intelligence*, volume 33, pages 9038–9045, 2019.

[54] L. Xie, Y. Liu, L. Jin, and Z. Xie. Derpn: Taking a further step toward more general object detection. In *AAAI Conf. on Artificial*

Intelligence, volume 33, pages 9046–9053, 2019.
[55] Y. Xu, Y. Wang, W. Zhou, Y. Wang, Z. Yang, and X. Bai. Textfield: Learning a deep direction field for irregular scene text detection. *IEEE Trans. Image Processing*, 28(11):5566–5579, 2019.
[56] C. Xue, S. Lu, and F. Zhan. Accurate scene text detection through border semantics awareness and bootstrapping. In *European Conf. Comput. Vision*, pages 355–372, 2018.
[57] C. Xue, S. Lu, and W. Zhang. MSR: multi-scale shape regression for scene text detection. In *Int. Joint Conf. on Artificial Intelligence*, pages 989–995, 2019.
[58] C. Yao, X. Bai, and W. Liu. A unified framework for multioriented text detection and recognition. *IEEE Trans. Image Processing*, 23(11):4737–4749, 2014.
[59] C. Yao, X. Bai, W. Liu, Y. Ma, and Z. Tu. Detecting texts of arbitrary orientations in natural images. In *Proc. Conf. Comput. Vision Pattern Recognition*, 2012.
[60] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang. Learning a discriminative feature network for semantic segmentation. In *Proc. Conf. Comput. Vision Pattern Recognition*, pages 1857–1866, 2018.
[61] C. Zhang, B. Liang, Z. Huang, M. En, J. Han, E. Ding, and X. Ding. Look more than once: An accurate detector for text of arbitrary shapes. In *Proc. Conf. Comput. Vision Pattern Recognition*, 2019.
[62] F. Zhang, Y. Chen, Z. Li, Z. Hong, J. Liu, F. Ma, J. Han, and E. Ding. Acfnet: Attentional class feature network for semantic segmentation. In *Proc. Int. Conf. Comput. Vision*, pages 6798–6807, 2019.
[63] Z. Zhang, C. Zhang, W. Shen, C. Yao, W. Liu, and X. Bai. Multi-oriented text detection with fully convolutional networks. In *Proc. Conf. Comput. Vision Pattern Recognition*, 2016.
[64] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *Proc. Conf. Comput. Vision Pattern Recognition*, pages 2881–2890, 2017.
[65] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, and J. Liang. EAST: an efficient and accurate scene text detector. In *Proc. Conf. Comput. Vision Pattern Recognition*, 2017.
[66] X. Zhu, H. Hu, S. Lin, and J. Dai. Deformable convnets v2: More deformable, better results. In *Proc. Conf. Comput. Vision Pattern Recognition*, pages 9308–9316, 2019.
[67] Z. Zhu, M. Xu, S. Bai, T. Huang, and X. Bai. Asymmetric non-local neural networks for semantic segmentation. In *Proc. Int. Conf. Comput. Vision*, pages 593–602, 2019.

Minghui Liao received his B.S. degree from the School of Electronic Information and Communications, Huazhong University of Science and Technology (HUST), China, in 2016. He is currently a Ph.D. student with the School of Electronic Information and Communications, HUST. His main research interests include scene text detection and recognition.

Zhisheng Zou received his B.S. degree from the school of Electronic Information and Communications, Huazhong University of Science and Technology (HUST), China, in 2018. He is currently a Master student with the School of Electronic Information and Communications, HUST. His main research is focus on scene text detection.

Zhaoyi Wan is a Ph.D. student in computer science at the University of Rochester. He received his B.S. degree in software engineering from Beihang University in 2016. Previously, he worked as an algorithm researcher at Megvii and a quantization trading researcher at JQ Investments, respectively. His research focus is on computer vision and artificial intelligence.

Cong Yao is currently with Alibaba DAMO Academy, Beijing, China. He received the B.S. and Ph.D. degrees in electronics and information engineering from the Huazhong University of Science and Technology (HUST), Wuhan, China, in 2008 and 2014, respectively. He was a research intern at Microsoft Research Asia (MSRA), Beijing, China, from 2011 to 2012. He was a Visiting Research Scholar with Temple University, Philadelphia, PA, USA, in 2013. His research has focused on computer vision and machine learning, in particular, the area of text detection and recognition in natural images.

Xiang Bai received his B.S., M.S., and Ph.D. degrees from the Huazhong University of Science and Technology (HUST), Wuhan, China, in 2003, 2005, and 2009, respectively, all in electronics and information engineering. He is currently a Professor with the School of Electronic Information and Communications, HUST. His research interests include object recognition, shape analysis, and OCR. He received IAPR/ICDAR Young Investigator Award in 2019. He is an associate editor for Pattern Recognition, Pattern Recognition Letters, and Frontiers of Computer Science.