Boundary-Aware Arbitrary-Shaped Scene Text Detector With Learnable Embedding Network

Mengting Xing, Hongtao Xie, Qingfeng Tan, *Member, IEEE*, Shancheng Fang, Yuxin Wang, Zhengjun Zha, *Member, IEEE*, and Yongdong Zhang, *Senior Member, IEEE*

***Abstract*—Benefiting from the popularity of deep learning theory, scene text detection algorithms have developed rapidly in recent years. Methods representing text region by text segmentation map are proved to capture arbitrary-shaped text in a more flexible and accurate way. However, such segmentation-based methods are prone to be disturbed by the text-like background patterns (like the fence, grass, etc.), which generally suffer from imprecise boundary detail problem. In this paper, LEMNet is proposed to handle the imprecise boundary problem by guiding the generation of text boundary based on a priori constraint. In the training stage, Boundary Segmentation Branch is firstly constructed to predict coarse boundary mask for each text instance. Then, through mapping pixels into an embedding space, the proposed Pixel Embedding Branch makes the embedding representation of boundary points learn to be more similar, meanwhile enlarging the characteristic distance between background points and boundary points. During inference, noise in the coarse boundary segmentation map can be effectively suppressed by a Noisy Point Suppression Algorithm among pixel embedding vectors. In this way, LEMNet can generate a more precise boundary description of text regions. To further enhance the distinguishability of boundary features, we propose a Context Enhancement Module to capture feature interactions in different representation subspaces, in which features are parallelly performed attention and concatenated to generate enhanced features. Extensive experiments are conducted over four challenging datasets, which demonstrate the effectiveness of LEMNet. Specifically, LEMNet achieves F-measure of 85.2%, 87.6% and 85.2% on CTW1500, Total-Text and MSRA-TD500 respectively, which is the latest SOTA.**

***Index Terms*—Scene text detection, boundary representation, false positive suppression.**

Manuscript received January 11, 2021; revised April 30, 2021 and June 18, 2021; accepted June 19, 2021. Date of publication June 30, 2021; date of current version June 9, 2022. This work was supported by the National Nature Science Foundation of China (62022076, U1936210, and 61972105), the Fundamental Research Funds for the Central Universities under Grant WK3480000011. The Associate Editor coordinating the review of this manuscript and approving it for publication was Dr. Lei Zhang. *(Corresponding author: Hongtao Xie and Qingfeng Tan.)*

Mengting Xing, Hongtao Xie, Shancheng Fang, Yuxin Wang, Zhengjun Zha, and Yongdong Zhang are with the School of Information Science and Technology, University of Science and Technology of China, Hefei, Anhui 230052, China (e-mail: metingx@mail.ustc.edu.cn; htxie@ustc.edu.cn; fangsc@ustc.edu.cn; wangyx58@mail.ustc.edu.cn; zhazj@ustc.edu.cn; zhyd73@ustc.edu.cn).

Qingfeng Tan is with the Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou 511442, Guangdong, China (e-mail: tqf528@gzhu.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TMM.2021.3093727.

Digital Object Identifier 10.1109/TMM.2021.3093727

I. Introduction

NOWADAYS, scene text detection and recognition [1]–[3] have a wide range of usage in smart systems, such as automatic driving, trademark recognition, blind guiding system, etc. Accurately localizing text region is the prerequisite of any text reading system. Benefiting from the rise of deep learning technology, a large number of corresponding methods for scene text detection tasks have emerged [4]–[6]. However, due to the complex backgrounds and variations of font, size, illumination condition, as well as the inconsistent expression for representing arbitrary-shaped text regions, localizing arbitrary-shaped text is still a challenging task to study nowadays.

Scene text detection methods can be briefly divided into segmentation-based methods [7]–[11] and regression-based methods [12]–[15]. Due to the strong ability for describing arbitrary-shaped texts, segmentation-based methods have attracted a lot of attention recently, which localize text regions based on pixel-level predictions. However, such methods generally suffer from the problem of imprecise boundary details because of the high complexity of natural scenario. The imprecise boundary may lead to the shape distortion of detecting result, further resulting in false positive cases (FPs).

As shown in Fig. 1(a), PSENet [7] is disturbed by the leaf texture, which makes a high response to the leaf area in the segmentation results and produces false positive detections (red rectangles in Fig. 1). To achieve more precise text boundary, Textfield [16] additionally attaches pixels with a unit vector formed by current pixel and pixel at the nearest boundary to refine text boundary in the post-processing stage. However, the complicated post-processing procedure and the inaccurate boundary vector prediction caused by definition of boundary vector in [16] limit the model performance. Thus, ContourNet [6] models texture information in two orthogonal directions and represents text with contour points by simultaneously considering the two direction results in the re-scoring algorithm. Nevertheless, the finer operation is still under a two-dimensional space, which is restricted for treating some complex background textures.

The causes of imprecise text boundary are attributed to two reasons in this paper: **1) the background texture interference.** High-diversity scenes exist a considerable amount of text-like background areas, like the fence, grass, etc. Detection methods are prone to be disturbed by those background

(a) PSENet

(b) LEMNet

Fig. 1. The segmentation result and detection visualization of PSENet [7] and LEMNet. (a) Text area segmentation is confused by background text-like noise, and leads to false positive detection. (b) LEMNet learns to represent pixels with learnable high-dimensional vectors, and accurate text boundary is achieved after the FP suppression.

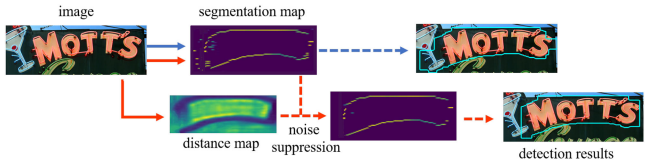


Fig. 2. Model without PE and CEM (blue flow) vs. LEMNet (red flow). Dotted arrow represents inference procedure, while solid arrow means process both in training and inference stage.

noise [17], generating imprecise text boundary representation. **2) The unlearnable noise suppression strategies.** Existing methods are to manually set the certain rule to suppress background noise in post-processing stage, like threshold filtering [18], point rescoring algorithm [6], etc. However, the unlearnable strategies are limited for treating complex background scenarios. Different from previous methods relying on unlearnable noise suppression process for boundary correction, we pay more attention to the boundary distinguishability in this paper and perform the noise suppression based on a learnable embedding space. Specifically, the text boundary is generated under a priori constraint.

To handle the imprecise boundary representation problem, we propose an arbitrary-shaped text detector in this paper, called LEMNet, which aims to constrain the boundary generating process and achieve more precise boundary representation with the help of high dimensional space (bottom branch in Fig. 2). As shown in Fig. 3, features extracted from backbone network are passed to a Region Proposal Network (RPN) [19] to generate text proposals (or RoIs). As for each RoI, we first use Boundary Segmentation (BS) Branch to predict coarse text boundary segmentation results. To better refine the boundary representation, we propose a parallel Pixel Embedding (PE) Branch to guide boundary representation learning under a priori constraint. PE branch embeds features within RoI into a high-dimensional embedding space, representing features with k-dimensional vector. Pixels within boundary area are trained to produce similar embedding representation, meanwhile the characteristic distance between background points and boundary points will be enlarged. In the test stage, the false positive boundary points can be suppressed by the proposed Noisy Point Suppression Algorithm among embedding vectors. To further enhance the discriminability of features, we introduce a Context Enhancement Module (CEM) constructed on an FPN-like backbone. CEM projects features into multiple heads and performs attention process to capture feature interactions in different representation subspaces. Then semantic powerful features can be achieved through combining multi-head enhanced features.

Compared with previous methods, LEMNet can achieve a tighter and more accurate boundary representation of text instances without complex hyperparameter settings. To the best of our knowledge, this is **the first work** to explore precise boundary details in high dimensional space for scene text detection task. In summary, the contributions of this paper are mainly summarized as three-fold:

- We explore more precise boundary segmentation in a high-dimensional embedding space. A priori constraint is introduced in Pixel Embedding Branch to guide the boundary generation in learning process. During inference, the noise in the coarse text boundary segmentation from Boundary Segmentation Branch can be effectively suppressed by the proposed Noisy Point Suppression Algorithm.
- A novel Context Enhancement Module is proposed to model the long term dependencies within text features, which helps the network extract more discriminative features and improves the detection performance.
- Extensive experiments demonstrate the effectiveness of LEMNet. Specifically, LEMNet achieves an F-measure of **85.2%**, **87.6%** and **85.2%** on CTW1500, Total-Text and MSRA-TD500 respectively, which is the latest state-of-the-art performance.

The remainder of this paper is organized as follows. Section II introduces the related work. In Section III, we describe LEMNet in detail. In Section IV, we demonstrate quantitative studies on four datasets. Finally, Section V and VI present the limitation and conclusion of our work.

II. Related Work

Benefiting from the superiority of deep learning theories, many solutions for scene text detection tasks have emerged recently and the detection performance has been greatly improved. Current scene text detectors can be briefly summarized as: segmentation-based methods [7]–[11] and regression-based methods [1], [12]–[14], [21], [22]. We will introduce these two categories of detectors respectively. Besides, we will also discuss some current instance-sensitive embedding methods.

A. Segmentation-Based Methods

Inspired by instance segmentation methods [23], [24], segmentation-based text detection methods output dense per pixel prediction to represent the text region. Pixellink [9] first links pixels in the same instance to segment different text instances, then extracts the text bounding box directly from the

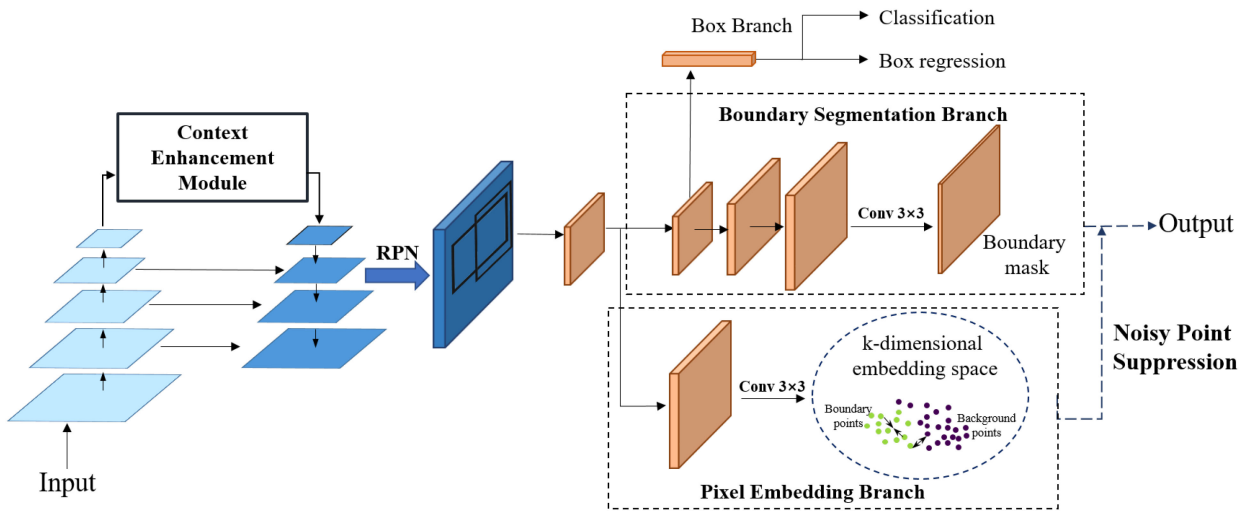


Fig. 3. The pipeline of LEMNet. Context Enhancement Module is constructed above the last stage of ResNet50 [20]. After region proposal process, the Box Branch is used for text classification and bounding box regression. Boundary Segmentation Branch outputs pixel-wise segmentation of boundary areas. Besides, the parallel Pixel Embedding Branch produces 8 channel embedding map to embed pixels into a higher dimensional space.

segmentation results. Textsnake [8] predicts text center line, text area and some geometric attributes to rebuild the text instance. A gradual scale expansion algorithm is proposed in PSENet [7] to better distinguish the closed texts. Besides, a direction field is learned in Textfield [16] to endue the 2D segmentation map with direction information, which is effective for dealing with irregular scene text detection. Tian *et al.* [10] propose a new text detection process, which uses a two step clustering strategy to split adjacent text instances from the predicted full and center segmentation maps. Pixels are mapped to an embedding space, and each text instance is considered as a cluster. Moreover, a differentiable binarization module is introduced in [11] to adaptively set the thresholds for binarization in post-processing procedure, which both improves model efficiency and performance.

Fully Convolutional Network (FCN) [25] can perform pixel-wise classification to the input image with any size, while a single network designed on FCN can hardly achieve good performance on various types of scenes. Besides, the above segmentation-based methods are sensitive to text-like background noise, which are prone to suffer from the imprecise boundary detail problem, resulting in false positive cases. Thus, LEMNet is designed in this paper to handle imprecise boundary problem by constraining the boundary generation. Particularly, the whole network is based on a two-stage framework to ensure the feature expression ability.

B. Regression-Based Methods

Regression-based methods deal with the text detection task by predicting the offset from predefined anchors or independent points. A majority of regression-based methods are inspired from common object detection means, like Faster R-CNN [19] and SSD [26]. By regressing the vertical offset and the height of predefined anchors, CTPN [12] first detects a set of partial text proposals then merges them within different instances. Inspired by SSD [26], angle offsets are additionally predicted in Seglink [13]. Besides, the within-layer and cross-layer link predictions are used to help the merge process. Thus, compared with CTPN [12], Seglink [13] is able to deal with multi-oriented texts more effectively. DDR [14] and EAST [21] perform regression operation in a different way, they directly predict offsets from points within text instances to four corner points or bounding box boundaries.

To improve the model performance for curved texts, LOMO [22] proposes an iterative refinement module to refine the directly regressed bounding boxes. Specially, a shape expression module is introduced in LOMO [22] to model three different geometric attribute of text lines for rebuilding an accurate shape representation of irregular texts. In addition, some anchor free methods also make a great contribution to curve text detection. Inspired by [27], TextRay [28] models the text geometric distribution under polar system, and the regression value is defined as the distance from the polar coordinate to the point at which the emitted N rays intersect the boundary. Under the motivation of separating adjacent texts in a better way, CRNet [29] presents a center-aware location algorithm to explicitly learn text center information by simultaneously considering the classification and location information. Although regression-based way is a relatively easy operation with many methods proposed, regression errors mentioned in [4] can still not be ignored. Besides, there is also much room for improvement in the detection performance of arbitrary-shaped texts.

C. Instance-Sensitive Embeddings

Instance-sensitive embeddings have been proved to be effective for tasks in different areas, such as human keypoints prediction [30], semantic segmentation [31], common object detection [32], [33], etc. Firstly mapping pixels in an image to a single point in high-dimensional feature space, pixel embeddings

are then assigned to different categories by various clustering methods. With the help of proposed corner pooling strategy, CornerNet [32] detects objects as paired keypoints, and embedding vectors are used to group corners which belong to the same object. For further enhancing the global information, center keypoint is introduced in CenterNet [33] to determine whether the detected boxes truly contain object. In [34], a discriminative loss is presented to pull pixels inside the same instance together and push pixels from the different instances apart. Implemented as a recurrent network, a new mean-shift clustering method is proposed in [35] for instance grouping. We draw lessons from the thought of embedding representation, and guide the model to map pixels into a learnable high-dimensional space. Different from the above mentioned methods, the embedding branch that we implemented works complementarily with the Boundary Segmentation Branch under a priori constraint. Besides, LEMNet can further refine the segmentation results by a Noisy Point Suppression Algorithm, which is also the first work to explore precise boundary prediction in high-dimensional space.

III. Our Method

The background texture in complicated scene is prone to disturb existing methods, which generally leads to imprecise boundary detection result. Besides, the unlearnable noise suppression strategies are limited for treating high-diversity scenario. Thus, LEMNet is proposed in this paper to handle the imprecise boundary representation problem based on high-dimensional embedding space. LEMNet mainly consists of four key parts: Backbone, Context Enhancement Module (CEM), Boundary Segmentation (BS) Branch and Pixel Embedding (PE) Branch. An overview of LEMNet is first briefly described in Sec.A. Then, in Section B, C, D, we will introduce the details of the proposed CEM and the implementation of BS branch, PE branch respectively. Finally, in Sec.E and F, we will describe the training targets and inference details of LEMNet.

A. Overall Pipeline

Firstly, ResNet50 [20] is used as backbone to extract shared image features. The inputs of CEM are features from the last stage of shared feature maps. CEM projects features into multiple heads to capture feature interactions in different subspaces. Taking advantage of the strong multi-scale information extraction ability of Feature Pyramid Network (FPN) [36], backbone is implemented in an FPN-like structure. Then, text proposals are generated through a Region Proposal Network (RPN) [19]. After the process of deformable RoI pooling [37], Boundary Segmentation Branch decodes boundary points by modeling local texture information. Parallelly, Pixel Embedding Branch is constructed for boundary refinement by constraining local texture representation in higher space in the training stage. During inference time, false positive boundary points can be effectively suppressed through a Noisy Point Suppression Algorithm among those high-dimensional feature vectors. In this way, text regions can be represented with a set of high-quality boundary points.

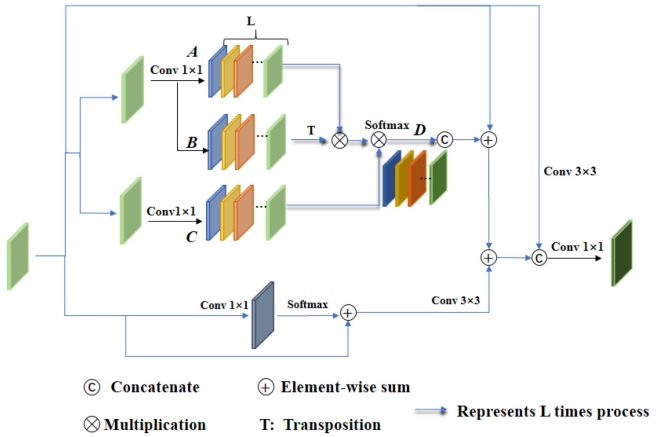


Fig. 4. The architecture of Context Enhancement Module. The top branch captures multi-head long range spatial dependencies, while the bottom branch simply maintains local discriminating information.

B. Context Enhancement Module

The attention mechanism [38], [39] has been adopted in many recent methods [40], [41] to boost the feature discriminability and further enhance the context information of achieved features. Since the boundary segmentation of each text region is predicted in a single RoI, the generation process lacks semantic context information and is prone to predict unclear boundary segmentation results. Furthermore, The imprecise boundary detail will inevitably cause boundary shift, representing as false positive detection results, which is the bottleneck for most existing methods to improve model performance. Inspired by [38], we propose a Context Enhancement Module to enhance text context information under multi-head attention strategy. Specially, by capturing pixel interactions in multiple subspaces, multi-head attention process can further improve the distinguishability of boundary semantic features.

As shown in Fig. 4, the input feature map F is parallelly processed through three convolution kernels to generate A, B and C at first. Specifically, $A_i \in \mathbf{R}^{HW\times(C/L)}$ represents different feature subspaces. To reduce the number of model parameters, A and B share the convolution kernel weight. Then, instead of performing a single attention process with C-dimensional features, we implement the attention in L heads to capture feature interactions in L different subspaces.

In each head, attention map is generated by matrix multiplication between A_i and B_i to model the long-range dependencies between any two pixels. Then, enhanced contextual feature map D_i is generated by considering each position response in attention matrix and the corresponding feature value in C_i. The process within each head i can be formulated as follows:

$$A = [A_0, A_1, \ldots, A_{L-1}], \tag{1}$$

$$D_i = Softmax(A_i \times B_i^T)C_i. \tag{2}$$

Finally, multi-head output features are concatenated and added to original feature F to generate the final contextual enhancement feature representation.

$$F_{out} = (Concat(D_0, D_1, \ldots, D_{L-1}) + F)W, \tag{3}$$

where W represents 3×3 convolution process, $Concat$ is the concatenation operation. Complementary to long-range relationship, we use a simple branch to maintain local discriminating information. First processed by a 1×1 conv, the input feature F later passes through softmax layer to generate the feature similarity map. Then enhanced local features are obtained by matrix multiplication and shortcut connection with the input features. Finally, original feature is processed by a 3×3 conv and concatenated with the summation between the multi-head enhanced features and the bottom enhanced local features. The final output of CEM is then obtained through a 1×1 conv process.

Considering the whole computation consumption and to balance the speed and accuracy of LEMNet, CEM is only performed on the last stage of shared features. The contextual information is modeled within gained semantic feature maps, which is of importance to help the network generate more discriminative boundary features.

C. Boundary Segmentation Branch

Existing boundary representation based methods [4], [5], [42] mainly regress boundary distance to localize arbitrary-shaped texts, which are limited for performing long-range prediction for long texts. In this paper, we only concern about the local texture information based on segmentation approach, which can effectively handle this problem. Specially, a Boundary Segmentation (BS) Branch is proposed to model local texture information, classifying boundary points from pixels in each RoI under the supervision of two-points wide boundary edge (B_{mask}). Based on refined boxes from Box branch, the input of BS branch is the shared features after deformable RoI pooling and bilinear interpolation. As shown in Fig. 3, we slide a convolutional kernel with size 3×3 over the feature map to model the local information and later apply a sigmoid layer to obtain the classification score map B_{seg}. Specifically, we set threshold as 0.2 to select candidate boundary points. The lower threshold setting is for achieving sufficient candidate boundary points passed to the later elimination.

D. Pixel Embedding Branch

BS branch models local texture information and represents text region with boundary segmentation map. However, the segmentation result often suffers from imprecise boundary details due to the complex background, which is sensitive for text-like background noise (e.g. the grass or fence texture). To handle the imprecise boundary problem, we design PE branch to constrain the generation of text boundary under a priori constraint. Areas other than boundary area are defined as background areas. The goal of PE branch is to create dense embeddings for input feature map, a k-dimensional vector is adopted to characterize local information.

Different from BS branch, PE branch predicts a k channel embedding map (B_{emb}). After the bilinear interpolation process, we slide a convolution kernel with size 3×3 and k output channels over the feature maps to model local information in a new approach, extending feature representation in each position to k-dimensional space. With the supervision of B_{mask}, center embeddings (c_{vector}) will be generated by calculating an average of embedding vectors of all the boundary points. Embedding vectors of boundary points will be pulled closer by constraining the distance between c_{vector} and themselves. Meanwhile, embedding vector of background points will be separated from embedding vector of boundary points by enlarging the distance between c_{vector} and embedding vectors of all the background points. During inference stage, false positive points can be eliminated effectively by performing the Noisy Point Suppression Algorithm, which will be detailed in Section F.

E. Training Objective

Implemented with a two-stage framework, LEMNet has multiple complementary tasks to learn, so the loss function takes the form of the combination of multi-tasks:

$$L = L_{RPN} + \lambda_1 L_{Box} + \lambda_2 L_{BS} + \lambda_3 L_{PE}, \tag{4}$$

where L_{RPN}, L_{Box} are the standard RPN and Box branch loss same as in [23], denoting RPN loss, box classification and regression loss. λ_1, λ_2 and λ_3 are balancing parameters that control the trade-off between these terms, and they are all set as 1 in LEMNet.

BS branch: Boundary Segmentation Branch essentially implements a binary classification between boundary area and background area. Thus we adopt a Binary Cross-Entropy Loss (BCE) for boundary segmentation learning. The loss function is formulated as:

$$L_{BS} = -y_i log\hat{y_i} - (1 - y_i) log(1 - \hat{y_i}), \tag{5}$$

where y_i and $\hat{y_i}$ denote the ground truth value and prediction respectively.

PE branch: L_{PE} is the loss function for the proposed Pixel Embedding Branch. Specifically, the main idea to constrain the generation of boundary area is that pulling the embedding representation within the boundary area together while pushing the embedding vectors between the boundary and background pixels apart. Inspired by [31], we adopt a discriminative loss in PE to consider both intra-class and inter-class distance. In order to improve the stability of learning process, the embedding vectors are not attached with certain values. Thus, we adopt two hinge as α and β to constrain the embedding learning. Loss function for PE branch is defined as follows:

$$L_{PE} = L_{pull} + L_{push}, \tag{6}$$

$$L_{pull} = \frac{1}{N} \sum_{n=1}^{N} \frac{1}{S_n} \sum_{i=1}^{S_n} [|\psi_{in} - c_n| - \alpha, 0]_{max}, \tag{7}$$

$$L_{push} = \frac{1}{N} \sum_{n=1}^{N} \frac{1}{M_n} \sum_{j=1}^{M_n} [\beta - |\omega_{jn} - c_n|, 0]_{max}. \tag{8}$$

For each RoI, c_n represents mean positive embedding vector computed within B_{mask}. ψ_{in} is the embedding features of pixel i in proposal n, which belongs to boundary points. ω_{jn} is the embedding features at pixel j in proposal n, which belongs to background points. Besides, $[\ldots]_{max}$ means taking the maximum value. S_n is the total pixel numbers of positive boundary

Algorithm 1: Noisy Point Suppression Algorithm.

Input: boundary segmentation map B_{seg}, embedding map B_{emb}.

$Output = zeros\_like(B_{seg})$
$C = SELECTPOSITIVE(B_{seg})$
$c_{vector} = Average(C)$
for (i, j) in B_{seg} **do**
  if $B_{seg}(i, j) > \theta_{mask}$ **then**
    if $|B_{emb}[i, j] - c_{vector}| <= \theta_d$ **then**
      $Output[i, j] = B_{seg}[i, j]$
    end if
  end if
end for
return $Output$

pixels in proposal n, M_n is the number of background pixels in proposal n, and N is the number of text instance proposals. L_{push} for training PE branch takes the average of loss calculated among all text instances. As for text instance n, the corresponding boundary mask segmentation results B_{mask} and embedding map B_{emb} can be gained first. Then, mean positive embedding vector c_n takes the mean value of all the embedding vectors belonging to the positive points in B_{mask}. Finally, for each positive pixel i in B_{mask}, the L1 distance will be calculated between the embedding features ψ_{in} and c_n. As illustrated in Eq. (7) and (8), the push loss takes the similar calculation process as above.

Moreover, we have pretested L1 and L2 norm to calculate embedding distances and finally choose L1 norm, because both can achieve the same performance and L1-based embedding is less susceptible to gradient explosions thus easier to train.

F. Inference Procedure

To distinguish different text instances. [10] maps pixels onto an embedding space, which uses DBSCAN [43] for clustering full segmentation map and center map, and further distributes pixels within full map while outside center map to the nearest clusters. However, the twice clustering operations are pretty time-consuming. In this paper, BS branch and PE branch are jointly working during inference time. As shown in Algorithm 1, without complicated clustering method [43], we simply use a Noisy Point Suppression Algorithm to eliminate background noise. We will pick one initial cluster center vector (c_{vector}) for boundary points and cluster the candidate boundary points to two categories. This operation is only performed on positive boundary points filtering from segmentation result, which is proved to be powerful in our experiments while keeping the efficiency of LEMNet.

We set two thresholds to filter the boundary segmentation result. If pixel in position (x_i, y_i) has a greater mask value than the mask threshold θ_{mask}, the pixel will be classified as candidate boundary points and used for point suppression. Besides, if the mask value is greater than the boundary threshold, the pixel will be averaged and used to generate c_{vector}. The algorithm is to calculate an L1 distance map between the boundary center vector and the embedding vector of candidate boundary points. Pixels that have a distance lower than distance threshold θ_d are kept and formed $Output$ map. Finally, boundary points representation will be obtained by performing Non-Maximum Suppression (NMS) on $Output$ map. The visualization of embedding distance map is shown in Fig. 5.

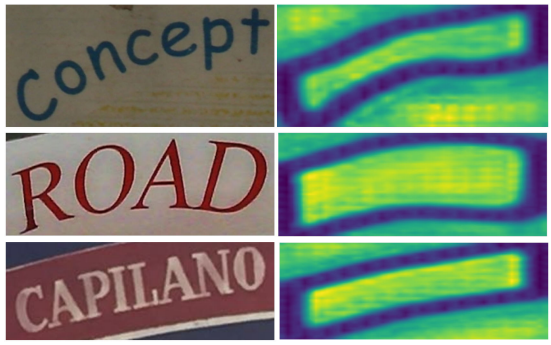


Fig. 5. The L_1 distance map used for Noisy Point Suppression during testing stages. The darker color means the smaller distance.

IV. Experiments

In this section, quantitative analysis will be conducted around the model performance on four popular text detection benchmarks: ICDAR2015 [44], CTW1500 [45], Total-Text [46] and MSRA-TD500 [47]. We will verify the effectiveness of the proposed module and discuss the details of the setting parameters. In addition, specific comparison will be made between LEMNet and some existing detection methods.

A. Benchmarks

Datasets used for evaluating LEMNet are briefly introduced as following:

1) ICDAR2015 [44]: ICDAR2015 [44] is a dataset for incidental scene text detection proposed in the Challenge 4 of ICDAR 2015 Robust Reading Competition. Taking by google glass, ICDAR2015 [44] totally includes 1500 images, 1000 for training and 500 for testing. Besides, the text instances are labeled on word level.

2) CTW1500 [45]: CTW1500 [45] is a challenging dataset for long curved text detection. It consists of 1000 training images and 500 testing images, including multi-oriented text and curved text. Texts in this dataset are largely in English and Chinese, and the text regions are all annotated in text-line manner with 14-vertex polygons.

3) Total-Text [46]: Total-Text [46] consists of 1255 training images and 300 testing images. It contains a large number of multi-oriented curved text instances, and the text regions are all annotated in word level with polygon.

4) MSRA-TD500 [47]: MSRA-TD500 [47] contains totally 500 images with multi-oriented text instances, 300 training images and 200 images for testing. Texts in this dataset are in various languages, including Chinese, English or the mixture of both. Besides, the annotation labels are all in line-level in MSRA-TD500 [47]. Because of the small number of images and

TABLE I
PERFORMANCE GAIN OF PIXEL EMBEDDING BRANCH AND CONTEXT ENHANCEMENT MODULE ON CURVE TEXTS

Dataset	Method	Recall	Precision	F-measure
CTW1500	Baseline	82.8	84.6	83.7
	Baseline + PE	83.3	85.7	84.5
	Baseline + PE + CEM	**83.8**	**86.6**	**85.2**
Total-Text	Baseline	**87.4**	85.3	86.3
	Baseline + PE	85.4	88.4	86.9
	Baseline + PE + CEM	85.4	**89.9**	**87.6**

the large variable text length, MSRA-TD500 [47] is an effective dataset to evaluate the performance of long text detection.

B. Implementation Details

ResNet50 [20] pretrained on ImageNet [48] is adopted as backbone. Implemented in pytorch, LEMNet uses Adam [49] as optimizer with batch size 1, 0.9 momentum and 0.0001 weight decay. The initial learning rate is 5×10^{-3} for CTW1500 [45] and Total-Text [46] dataset, 2.5×10^{-3} for ICDAR2015 [44] and MSRA-TD500 [47]. As for data augmentation, we perform random rotation, random horizontal flip and random crop for input images. Besides, in the training stage, the short side of the input images is randomly resized to (400, 600, 720, 1000, 1200), while the long side is resized to 2000. The aspect ratios of anchors are set to (0.25, 0.5, 1.0, 2.0, 4.0) and the strides of anchors are correspondingly set to (4, 8, 16, 32, 64) in all experiments.

In this paper, baseline model is LEMNet without CEM and PE branch. Besides, we use $cv2.drawContours$ function in $opencv$ module to obtain a two-points wide edge. Points within the generated edge are defined as ground-truth boundary points and are used to supervise the training process. During inference, θ_{mask} for boundary segmentation filtering is set to 0.2. The boundary threshold for filtering boundary points that corresponding to generate center embedding vectors is set to 0.9. Besides, the distance threshold θ_d is set to 1.0. α, β used for constraining embedding distances take the value of 0.5 and 1.5 respectively.

C. Ablation Studies

1) Discussions About PE Branch and CEM: We evaluate the benefits of Pixel Embedding Branch on CTW1500 [45] and Total-Text [46], which mainly contain curved texts. The baseline model only uses Boundary Segmentation Branch to perform pixel-wise classification, which is prone to detect the text-like background area as text regions. As can be seen in Table I, compared with baseline model, model with PE branch can achieve gains of 0.8% for CTW1500 [45] in F-measure. Specially, PE branch increases the precision value of model performance by 1.1% and 3.1% on CTW1500 [45] and Total-text [46] respectively. Besides, combined with CEM, LEMNet finally achieves 85.2% F-measure for CTW1500 [45] and 87.6% F-measure for Total-text [46].

Although with 2% recall rate drops on Total-Text, it is worth noting that Baseline + PE surpasses Baseline by 0.6% in F-measure and gains 3.1% improvement on Precision. There exists a fundamental trade-off between recall and precision evaluation indicators, where F-measure balances the two and gives a convincing measurement. To more intuitively present the effectiveness of PE branch, we further show the precision vs. recall curve sampled by different θ_{mask} for Baseline and Baseline + PE. As shown in Fig. 6, the PR curve can be divided into two sets. Curves in legend of θ_{mask} represent the performance of Baseline under different mask threshold (0.1, 0.2, 0.3), while curves in legend of θ^*_{mask} are achieved by Baseline + PE. It is obvious to see that curves of Baseline + PE totally encase that of Baseline, which demonstrates that Baseline + PE surpasses Baseline under different mask threshold settings all the time. Moreover, $\theta^*_{mask} = 0.2$ gains the best performance, while model with a higher or lower θ_{mask} performs minor difference. Thus, PE branch is effective for predicting more precise results.

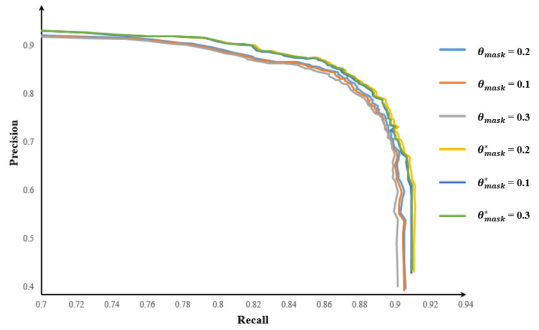


Fig. 6. Precision-Recall Curve. θ_{mask} represents for mask threshold setting for Baseline. θ^*_{mask} is for Baseline + PE.

TABLE II
ABLATION STUDY OF THE DIMENSION OF EMBEDDING VECTORS, α AND β, TESTING ON CTW1500 [45]

k-dimension	α/β	Recall	Precision	F-measure
4	0.5/1.5	85.1	84.4	84.7
	1.5/0.5	84.9	83.9	84.4
	1.5/1.5	84.3	84.0	84.1
8	0.5/1.5	**83.8**	**86.6**	**85.2**
	1.5/0.5	82.5	86.2	84.3
	1.5/1.5	82.3	86.2	84.2
16	0.5/1.5	83.7	86.4	85.0
	1.5/0.5	82.2	85.9	84.0
	1.5/1.5	83.8	84.9	84.3

2) Discussion of Settings in PE Branch: Moreover, we investigate the effect of embedding space dimension on the performance of LEMNet in CTW1500 dataset [45]. We set different dimension of embedding vectors while keeping the other settings unchanged. The results are shown in Table II, from which we can see that when the embedding dimension is set to 8, the model can achieve the best performance with 83.8%, 86.6% and 85.2 % for recall, precision and F-measure respectively. The performance will get worse when the dimension becomes smaller or larger than this setting. PE branch with large predictive output dimension will enlarge the network parameter scale, which

TABLE III
THE RELATIONSHIP BETWEEN MODEL PERFORMANCE AND THE NUMBER OF ATTENTION HEAD IN CONTEXT ENHANCEMENT MODULE, TESTING ON CTW1500 [45]

L-head	Recall	Precision	F-measure
1	83.8	84.6	84.2
4	**84.6**	85.0	84.8
8	83.8	**86.6**	**85.2**
16	82.0	86.1	84.0

is difficult to converge with no extra dataset used in our training stage. Meanwhile, if the output dimension is too small, the learned representation ability of high-dimensional embedding space can not perform effective filtering operation to those false positive boundary points, which leads to 0.5% falls in F-measure. So we set the embedding dimension to 8 in the remaining experiments.

Specifically, α and β are parameters for specifying the near and far distance threshold with which the calculated embedding distance compared, which means the embedding vectors are not enforced to converge to a single point but can exist on a local scope in the feature space. We further attach k together with different α and β to verify the relationship. As shown in Table II, in all dimension settings (k = 4, 8, 16), LEMNet performs best when α equals to 0.5 and β equals to 1.5, which is not relevant to dimension space size. Thus we choose this setting as default.

3) Influences of the Attention Head Number: The relationship between the network performance and the number of attention head in Context Enhancement Module is explored on CTW1500 [45] dataset then. Table III shows the experimental results, from which we can find that CEM with 8 heads achieves higher performance than the other three settings (85.2% vs. 84.8%, 84.0%). The head number L indicates the number of feature subspaces in CEM that we project to. As shown in Table III, CEM with 4 heads shows poorer feature representation ability than 8-head CEM, resulting in 0.4% falls in F-measure. However, when L increases to 16, the network capability to express each subspaces cannot learn well due to the limited dataset scale (84.0% vs. 85.2%). Moreover, we further reduce the head setting L to 1. Compared to 8-head CEM, LEMNet with L equals to 1 results in 1.0% falls in F-measure, which further verifies the multi-head feature projection is effective.

4) Discussions About the Deformable RoI Pooling: After the proposal generating stage, the important part before passing features to each branch is the sampling process. To further enhance the network learning ability, we introduce deformable RoI pooling [37] to perform sampling operation, which determines the position of the sample point by learning the additional offsets. Specifically, we compare the model performance with deformable RoI pooling and the model without this strategy on Total-Text [46]. As shown in Table IV, using deformable RoI pooling strategy can improve LEMNet by 0.5% in F-measure, which verifies the effectiveness of this strategy in LEMNet. Besides, Baseline + PE with deformable RoI pooling can also gain 0.5% improvement.

TABLE IV
ABLATION STUDY EXPERIMENTS OF DEFORMABLE ROI POOLING STRATEGY, W/O MEANS MODEL WITHOUT THE DEFORMABLE ROI POOLING PROCESS, TESTING ON TOTAL-TEXT [46]

Method	Recall	Precision	F-measure
Baseline + PE(w/o)	85.2	87.6	86.4
Baseline + PE	**85.4**	**88.4**	**86.9**
LEMNet (w/o)	85.2	89.0	87.1
LEMNet	**85.4**	**89.9**	**87.6**

TABLE V
THE DISCUSSION OF WEIGHTED BINARY CROSS ENTROPY (WBCE) LOSS IN BS BRANCH ON CTW1500

Method	Loss	R	P	F
Baseline	BCE	**82.8**	84.6	83.7
	WBCE	81.7	**86.0**	**83.8**
Baseline + PE	BCE	**83.3**	**85.7**	**84.5**
	WBCE	82.6	85.6	84.1
LEMNet	BCE	**83.8**	86.6	**85.2**
	WBCE	82.5	**87.1**	84.7

5) Discussions About Boundary Weight in Boundary Segmentation Branch: We further adopt the Weighted Binary Cross Entropy Loss in BS branch to compare its performance with BCE loss on CTW1500 dataset. Specifically, the weight to balance boundary and background points is calculated under the proportion of boundary points in total points. As shown in Table V, for Baseline, the addition of balance weight makes 0.1% increase in F-measure, which is a better performance compared to our implementation. However, for Baseline + PE and LEMNet, the model performance decreases 0.4% and 0.5% respectively on F-measure. We attribute this phenomenon to the network architecture and the model design. Under a two-stage framework, LEMNet assembles multi-task loss to achieve the training objective, which is able to improve the generalization of each task. Moreover, we design a Pixel Embedding branch to map features into a higher dimensional space, which aims at constrain the boundary learning process and suppress the FP points existing in two-dimensional boundary generation, further attaching boundary segmentation with a stronger constraint. Thus, BCE loss is used in our method.

6) Effectiveness of Noisy Point Suppression Algorithm: Benefiting from the Noisy Point Suppression Algorithm, LEMNet can effectively gain accurate boundary details. Although higher threshold can filter more FP boundary points, the true positive points may be over suppressed simultaneously. Besides, the lack of support of texture information in PE branch will also lead to the serious decline of precision rate. As shown in Table VI, compared with results after Noisy Point Suppression Algorithm, directly filtering boundary segmentation from BS branch with 0.2 mask threshold results in 0.3% falls in F-measure for LEMNet. Notably, we find $\theta_{mask} = 0.2$ can help our method obtain the best results. As for Baseline, improving the filter threshold to 0.3 results in 0.1% falls in F-measure. Thus, the PE branch and Noisy Point Suppression Algorithm are essential for achieving better detection results.

TABLE VI
COMPARISONS OF DIFFERENT MASK THRESHOLD IN TOTAL-TEXT. W/O REPRESENTS WITHOUT NOISY POINT SUPPRESSION ALGORITHM DURING INFERENCE

Method	θ_{mask}	Recall	Precision	F-measure
Baseline	0.2	**87.4**	**85.3**	**86.3**
	0.3	87.3	85.1	86.2
LEMNet	0.2	**85.4**	**89.9**	**87.6**
	0.2 (w/o)	85.2	89.6	87.3
	0.3 (w/o)	85.1	89.5	87.2

TABLE VII
COMPARISON WITH STATE-OF-THE-ART METHODS ON CTW1500 [45]. * INDICATES THE RESULTS FROM [8]. MULTI-SCALE TESTING AND ENSEMBLE ARE NOT INCLUDED

Method	Ext	Recall	Precision	F-measure
CTPN [12]*	-	53.8	60.4	56.9
EAST [21]*	-	49.1	78.7	60.4
CTD+TLOC [45]	-	69.8	77.4	73.4
Textsnake [8]	✓	**85.3**	67.9	75.6
LOMO [22]	✓	76.5	85.7	80.8
Tian *et al.* [10]	✓	77.8	82.7	80.1
Wang *et al.* [5]	-	80.2	80.1	80.1
PAN [50]	-	77.7	84.6	81.0
Textfield [16]	✓	79.8	83.0	81.4
MSR [4]	✓	78.3	85.0	81.5
PSENet-1s [7]	✓	79.7	84.8	82.2
DBNet [11]	✓	80.2	**86.9**	83.4
ContourNet [6]	-	84.1	83.7	83.9
LEMNet	-	83.8	86.6	**85.2**

D. Evaluation on Long Curved Text Benchmark

We evaluate LEMNet on CTW1500 [45] to test its ability for detecting long curved texts. Comparison results are listed in Table VII.

As shown in Table VII, LEMNet performs better than all other existing methods, achieving a result of 83.8%, 86.6% and 85.2% in recall, precision and F-measure respectively. Specifically designed for treating curved texts, CTD+TLOC [45], Textsnake [8] and Textfield [16] achieve 73.4%, 75.6% and 81.4% in F-measure respectively. LEMNet outperforms those by at least 3.8% in F-measure, proved to be a more effective method to treat curved texts. Compared with the boundary based representation methods [4]–[6], [22], LEMNet shows superior performance in recall, precision and F-measure. Though the proposal-based text shape expression module in LOMO [22] aims to achieve tighter text representation, LEMNet obtains more higher performance (85.2% vs 80.8%). We attribute the high model performance to the Pixel Embedding Branch and semantic powerful features gained in Context Enhancement Module. The former makes boundary pixels to be more aggregated in high-dimensional embedding space and performs proposal-based embedding vectors distance constraining, achieving false positive suppression in an effective way. The latter helps LEMNet to extract more distinguishing boundary features to accurately classify proposals.

It is worth mentioning that LEMNet outperforms Tian *et al.* [10] by a large margin (85.2% vs. 80.1% in F-measure). Tian *et al.* [10] introduce embedding vectors to distinguish different text instances, while LEMNet aims to deal with imprecise boundary detail problem which widely exists in normal scene text detection methods. Different from [10], PE branch in LEMNet learns feature expression with the help of the calculated center embedding vector (c_{vector}), and adopts Noisy Point Suppression Algorithm to distinguish the boundary points. Besides, [10] uses external Synthtext [51] dataset to pretrain the whole network, which is a pretty large synthetic dataset containing more than 800 thousand synthetic images with nearly 8 million text instances. Without any extra dataset to help the learning process, LEMNet can still gain significant improvement on recall, precision and F-measure. The visualization results on CTW1500 [45] are shown in Fig. 7.

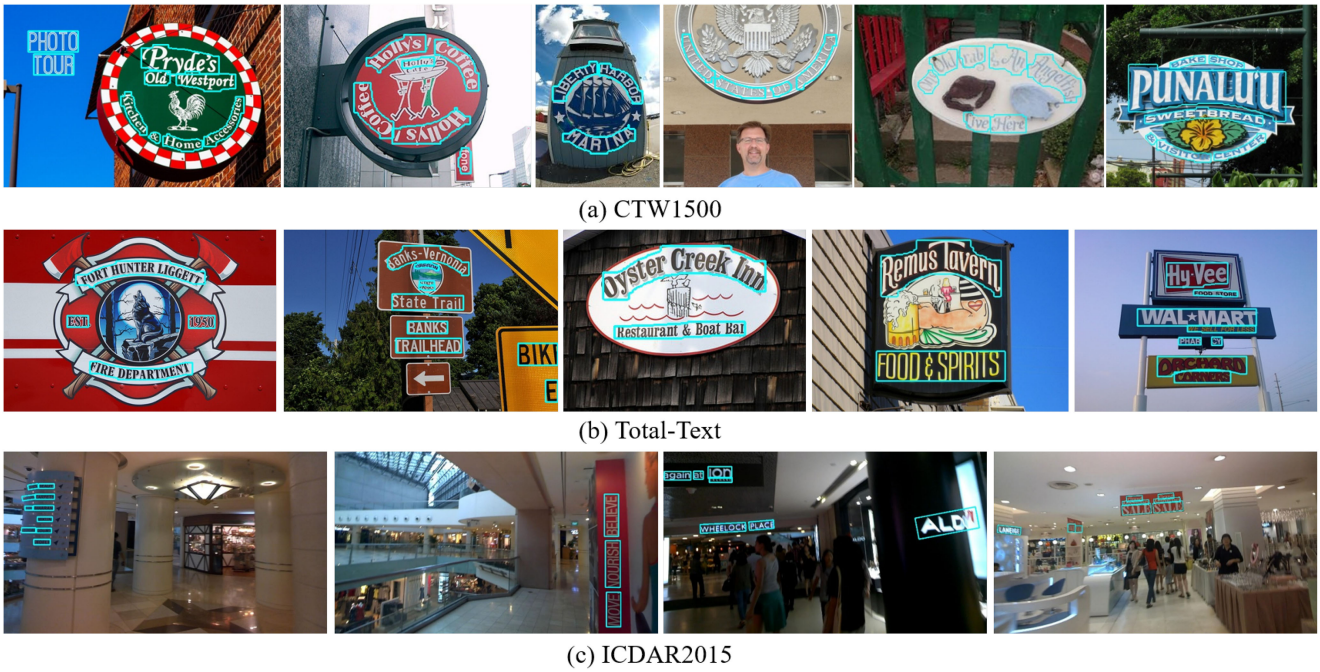
TABLE VIII
THE SINGLE SCALE RESULT ON TOTAL-TEXT [46]. * INDICATES THE RESULTS FROM [8]

Method	Ext	Recall	Precision	F-measure
SegLink* [13]	-	23.8	30.3	26.7
EAST* [21]	-	36.2	50.0	42.0
Textsnake [8]	✓	74.5	67.9	75.6
Wang *et al.* [5]	-	76.2	80.9	78.5
Lyu *et al.* [52]	✓	75.4	81.8	78.5
MSR [4]	✓	74.8	83.8	79.0
TextDragon [53]	✓	74.2	84.5	79.0
Textfield [16]	✓	79.9	81.2	80.6
PSENet [7]	✓	78.0	84.0	80.9
LOMO [22]	✓	75.7	88.6	81.6
SPCNET [17]	✓	82.8	83.0	82.9
PAN [50]	-	79.4	88.0	83.5
CRAFT [54]	✓	79.9	87.6	83.6
DBNet [11]	✓	82.5	87.1	84.7
ContourNet [6]	-	83.9	86.9	85.4
Wang *et al.* [42]	✓	85.0	88.9	87.0
LEMNet	-	**85.4**	**89.9**	**87.6**

E. Evaluation on Curved Text Benchmark

To show the model performance for detecting curved texts, we evaluate LEMNet on Total-Text [46] and compare its performance with state-of-the-arts. The compared results are listed in Table VIII. Specifically, model performance is evaluated using the protocols provided in [46]. Considering the difficulty to make a fair comparison of multi-scale testing strategy due to the inconsistent standard and too much speed sacrifice, multi-scale testing operation and other strategies are not included in our testing stage.

As shown in Table VIII, LEMNet in single testing scale outperforms all existing methods and obtains a new state-of-the-art performance (87.6% in F-measure). LEMNet significantly surpasses existing regression methods [5], [13], [21], [22] by at least 6.0% in F-measure due to the strong feature representation ability of LEMNet. Text boundary representation based method [5] adopts RNN as refinement network to treat arbitrary-shaped texts. Although squeeze-and-excitation block is implemented in their backbone, LEMNet shows superiority with 9.1% gains in F-measure benefiting from Noisy Point Suppression Algorithm. For fair comparison, [42] is shown here without recognition branch. Although [42] is pretrained in Synthtext [51], LEMNet gets 0.6% improvement in F-measure with only official dataset. CRAFT [54] additionally uses character-level annotations to supervise the learning procedure, which is difficult for practical application. Compared with [54], LEMNet only trained with



(a) CTW1500

(b) Total-Text

(c) ICDAR2015

Fig. 7. Visualization results of LEMNet on three universal scene text datasets. From top to bottom: CTW1500, Total-Text and ICDAR2015.

word-level supervision shows advantage in F-measure (87.6% vs. 83.6%).

Specifically, PSENet [7] gradually expands the predicted various scale text kernels to split close text instances, which is prone to suffer from false positive detections (see in Fig. 1). However, LEMNet can achieve better results (87.6% vs. 80.9%) benefiting from the Noisy Point Suppression Algorithm based on embedding space. Compared with SPCNET [17] eliminating FP cases through rescoring process and contextual information enhancement, LEMNet achieves much better performance (87.6% vs. 82.9% in F-measure) by modeling pixel relationships in high dimension space. Besides, ContourNet [6] models texture information from two orthogonal directions and refines the text boundary representation by considering the response of both direction, which achieves 85.4% in F-measure. However, the boundary finer procedure is unlearnable, which is limited for dealing with some complex background textures. Compared with ContourNet [6], LEMNet explores distinguishability boundary texture in high-dimensional space with the help of the boundary discriminative loss function, achieving 2.2% enhancement in F-measure.

As shown in the bottom of Fig. 8, the imprecise boundary can be rectified and a more accurate boundary description can be achieved by LEMNet, which intuitively proves that the proposed PE branch is helpful for suppressing the false positive samples.

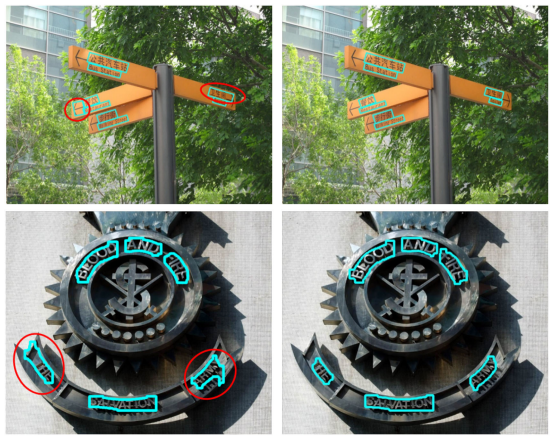


Fig. 8. The visualization of false positive suppression samples. The left column shows the results of baseline model. The right column shows the results of LEMNet. Samples circled in red circle exist imprecise boundary detail problem, even representing as false positive detection results.

F. Evaluation on Oriented Text Benchmark

We evaluate LEMNet on ICDAR2015 [44] to test its ability for multi-oriented text detection. The initial learning rate for ICDAR2015 [44] is set to 2.5×10^{-3}, and is divided by 10 at 120 k and 160 K iterations. Evaluation results are shown in Table IX, from which we can see that LEMNet achieves

TABLE IX
THE SINGLE SCALE RESULTS ON ICDAR2015. DETEVAL IS USED TO EVALUATE THE RESULTS

Method	Ext	Recall	Precision	F-measure
RRPN [1]	✓	73.0	82.0	77.0
Lyu *et al.* [52]	✓	81.2	85.8	83.4
Wang *et al.* [42]	✓	82.2	88.1	85.0
Liu *et al.* [18]	✓	83.8	89.4	86.5
Tian *et al.* [10]	✓	85.0	88.3	86.6
Wang *et al.* [5]	-	83.3	90.4	86.8
ContourNet [6]	-	**86.1**	87.6	86.9
SPCNET [17]	✓	85.8	88.7	87.2
DBNet [11]	✓	83.2	**91.8**	**87.3**
LEMNet	-	85.9	88.3	87.1

TABLE X
THE SINGLE SCALE RESULTS ON MSRA-TD500. † DENOTES RESULTS BASED ON MULTI-SCALE TESTING

Method	Ext	Recall	Precision	F-measure
RRPN [1]	✓	68.0	82.0	74.0
Pixellink [9]	-	73.2	83.0	77.8
Textsnake [8]	✓	73.9	83.2	78.3
DBNet [11]	✓	79.2	**91.5**	84.9
Liu *et al.* [18]	✓	80.5	89.6	84.8
CRNet [29]	✓	82.0	86.0	84.0
CRNet† [29]	✓	83.6	86.5	85.1
Baseline	-	80.2	77.5	78.8
LEMNet	-	**84.8**	85.6	**85.2**

85.9%, 88.3% and 87.1% for recall, precision and F-measure respectively. As shown in Table IX, LEMNet surpasses most existing proposal-based methods [1], [18] by at least 0.6% in F-measure. [18] treats the FP problem by obtaining more compact bounding boxes through adjusting the rescoring process. However, LEMNet can generate a more compact and accurate boundary representation for text regions by eliminating those false positive boundary points with the help of learnable embedding space. Benefiting from the high-level semantic information, LEMNet surpasses ContourNet [6] by 0.2% in F-measure. Though SPCNET [17] and DBNet [11] pretrain their model on SynthText [51], LEMNet only trained with official training images can achieve a comparable performance (87.1% vs. 87.2%, 87.3% in F-measure). For fair comparison, we compare the performance of "det only" without recognition branch, and LEMNet can also significantly outperform Lyu *et al.* [52] by 3.7% in F-measure.

For fair comparison with embedding-based method [10], we further explore the efficiency of LEMNet, testing LEMNet on 720P images from ICDAR2015 [44] on a single NVIDIA TITAN X Pascal GPU. Tian *et al.* [10] perform clustering twice and achieve 3FPS speed, while LEMNet can achieve faster speed (4FPS vs. 3FPS). Besides, LEMNet explores the distinguishability of text boundary information and deals with the common imprecise text boundary problem, which has been ignored by existing method. Some visualization results of LEMNet are shown in Fig. 7. Although some scene suffers from complex background, LEMNet can accurately capture the location of texts as well.

G. Evaluation on Long Oriented Text Benchmark

We evaluate LEMNet on MSRA-TD500 [47] to test its ability for multi-oriented text detection. Without any other extended data, we only use official training images in MSRA-TD500 [47] to train LEMNet. Besides, we additionally add hard examples in the training process to deal with the small dataset scale.

Testing results are shown in Table X. With the help of proposed PE branch and CEM, LEMNet achieves state-of-the-art results in recall, precision and F-measure respectively, surpassing most existing methods (e.g. RRPN [1], Textsnake [8], Pixellink [9], Liu *et al.* [18]). CRNet† [29] adopts multi-scale testing in testing process and achieves 1.1% improvement than CRNet [29]. However, the settings of multi-scale testing in public methods have no uniform standard, which is hard for fair comparison. LEMNet with single testing scale can achieve comparable results with CRNet† [29] and further surpasses CRNet [29] with single testing scale by 1.2% in F-measure. Some visualization results on MSRA-TD500 [47] are shown in Fig. 9.

Fig. 9. Text detection results on MSRA-TD500.

Moreover, as shown in the last two lines in Table X, LEMNet surpasses baseline model by a significant margin (85.2% vs. 78.8%), which shows the effectiveness of PE branch and CEM in a direct way. To further intuitively indicate the false positive suppression effect, we visualize some results on MSRA-TD500 [47] based on baseline and full model respectively. As shown in the top row in Fig. 8, the baseline model mistakes some text-like background area (circled by red ellipse) as texts, which will be further suppressed in LEMNet. Besides, the shape deformation of detection caused by neighbor text-like noise will also be refined by Noisy Point Suppression Algorithm. We attribute the high model performance to the help of high-dimensional embedding space in Pixel Embedding Branch and the semantic powerful features obtained through Context Enhancement Module, which is important to obtain accurate arbitrary-shaped text detection result.

H. Performance Discussion

1) Effectiveness of LEMNet: We think provide the end-to-end evaluation can further support our method. Thus, we additionally conduct experiments in two aspects: 1) evaluate LEMNet based on TIoU metric [55] to prove its effectiveness in providing tight detection results. 2) Evaluate the recognition performance by sending the detection results to recent popular text recognition method.

As shown in Table XI, although the performance indexes decrease under TIoU metric [55], LEMNet also obtains best performance. Evident TIoU results from [55] are also listed in Table XI for comparison. Compared with MASK R-CNN++ [23] which is trained by using additional data selected from RCTW-17 [56], LEMNet gains 1.8% and 0.9% improvement in F-measure under IoU and TIoU respectively, which verifies the compactness of our detection regions.

TABLE XI
TESTING RESULTS UNDER TIOU METIRC ON MSRA-TD500. TR, TP, TF MEANS TIOU RECALL, TIOU PRECISION AND TIOU F-MEASURE RESPECTIVELY

Method	R	P	F	TR	TP	TF
EAST [21]	61.5	49	54.6	41.1	36.9	38.9
MASK R-CNN++ [55]	83.2	83.7	83.4	**63.8**	67.9	65.8
LEMNet	**84.8**	**85.6**	**85.2**	63.2	**70.6**	**66.7**

TABLE XII
WORD RECOGNITION EVALUATION ON ICDAR2015. CRW IS SHORT FOR CORRECTLY RECOGNISED WORDS, TED IS SHORT FOR TOTAL EDIT DISTANCE. CA REPRESENTS FOR CHARACTER-LEVEL ACCURACY

Method	CRW	TED	CA
Baseline	74.1	187	90.1
LEMNet	75.2	174	90.8

TABLE XIII
COMPARISON OF MODEL PERFORMANCE ON CTW1500 AT DIFFERENT TEXT SIZE. SMALL MEANS TEXTS WITH AREA LESS THAN 1500. LARGE MEANS TEXTS WITH AREA LARGER THAN 4500. MIDDLE MEANS THE REMAINING TEXTS. F-MEASURE IS LISTED HERE

Method	Small	Middle	Large
Baseline	75.4	86.4	90.1
LEMNet	77.8	88.4	90.3
Gain	**+2.4**	**+2.0**	**+0.2**

Moreover, detection results from LEMNet are sent to ASTER [57] for evaluating recognition performance on ICDAR2015. We firstly screen out the detection results of Baseline and LEMNet for the same text words, and the corresponding text annotations form the final text words test set for evaluating recognition results. As shown in Table XII, the correctly recognized words from LEMNet account for 75.2% in total words, surpassing Baseline 1.1%, which effectively demonstrates the accuracy of detection results from LEMNet. Besides, character-level accuracy [58] of the test results from LEMNet reaches 90.8%, improving Baseline by 0.7%. As illustrated above, LEMNet can achieve superior performance compared with Baseline, and further obtains a tighter detection results.

2) Comparisons of Results in Different Sizes: We further conduct additional experiment to test model performance in varying text sizes. For better comparison, we divide the text in CTW1500 into three parts, representing as Small (area $<$ 1500), Middle (1500 $<$ area $<$ 3500) and Large (area $>$ 3500). As shown in Table XIII, taking advantage of the high dimensional space for achieving more accurate segmentation boundary details, LEMNet gains 2.4%, 2.0% and 0.2% F-measure in these three parts respectively.

We further visualize some small text instance detection results of LEMNet vs. Baseline on CTW1500. As shown in Fig. 10, LEMNet can accurately localize the missed small text instances in Baseline detection results, which intuitively confirms the 2.4% performance gains on small text detection shown in Table XIII. Besides, compared with Baseline, LEMNet enhances small text detection benefitting mainly from following two aspects: 1) the improved capability of modeling context information. Textures

Fig. 10. The visualization of small text detection samples on CTW1500. The left column shows the results from Baseline. The right column shows the results from LEMNet. Samples circled in red circle are the missed or false detected small texts.

of small texts are prone to be confuse with background texture, thus not easy to be distinguished. However, CEM models long dependencies within text features and further enhances the feature discriminability, which is proved to be effective for boosting the whole model performance. Thus, the strong context information can help distinguish small texts from background and improve model performance for small text detection. 2) The boundary rectification through Noisy Point Suppression Algorithm, which is detailed in Sec. F. Besides the missed small detected texts, small texts which are prone to be confuse with background texture will also be apt to cause imprecise boundary details, further resulting in false positive detections. Taking advantage of the learned embedding map, Noisy Point Suppression Algorithm can suppress the false detected points, thus achieves more accurate small text detection results. Benefitting from the above two aspects, LEMNet can more effectively detect small texts and achieve 2.4% performance gains on small text detection testing on CTW1500.

3) Compared With ContourNet and Region-Based Segmentation Methods: Different from ContourNet [6], LEMNet benefits mainly in following two aspects: 1) we model text boundary features in a higher dimension space, which is more robust to handle complex background interference. 2) A new Noisy Point Suppression Algorithm is proposed in LEMNet. Compared to ContourNet [6] which suppresses false positive samples by only considering the response in orthogonal direction, our noisy point suppression strategy is performed upon a learnable embedding space, which can achieve more accurate detection results. As shown in Table VII and Table VIII, LEMNet surpasses ContourNet [6] 1.3% and 2.2% in F-measure respectively. Thus, benefiting from the above two aspects, LEMNet can more effectively suppress false positive samples and achieve more accurate detection results.

Moreover, compared with region-based segmentation methods like SPCNet [17], LEMNet can achieve an impressive

TABLE XIV
INFERENCE SPEED AND MODEL FLOPS ON CTW1500

Method	Recall	Precision	F-measure	FPS	GFLOPs
Baseline	82.8	84.6	83.7	**3.79**	89.78
Baseline + PE	83.3	85.7	84.5	3.47	94.81
LEMNet	**83.8**	**86.6**	**85.2**	3.14	112.35

performance especially for detecting curved texts (87.6% vs. 82.9% in F-measure on Total-text). We attribute the good model performance to the following two aspects: 1) accurate boundary representation. Region-based segmentation methods often suffer from coarse boundary detail problem, which is prone to cause FP detection results. LEMNet handles imprecise boundary problem by constraining the boundary feature representation in high-dimensional space, which is proved to be effective for suppressing background noise. 2) Noisy Point Suppression Algorithm. Almost region-segmentation based methods use threshold filtering in post-processing stage to correct segmentation results in two-dimensional space, which is limited for treating complex background scenarios. Different from these methods, our Noisy Point Suppression Algorithm is performed upon a learnable embedding space, effectively dealing with background interference. 3) Focuses on boundary areas. Different from region-based methods which pay attention to the entire text areas, we only focus on the boundary area of the texture transition, thereby reducing the difficulty of network learning and achieving better detection results.

4) Inference Speed: As LEMNet is implemented under a two-stage framework, the highlight of this paper is not at the speed performance but providing more accurate boundary representation. We have tested the inference speed of LEMNet on CTW1500 dataset. Notably, LEMNet is tested on 720P images from CTW1500 [44] on a single NVIDIA TITAN X Pascal GPU. As shown in Table XIV, our baseline model can reach 3.79 FPS with 83.7% F-measure, and the addition of PE branch improves F-measure 0.8% with only 0.32 FPS decrease. Furthermore, LEMNet can finally reach 85.2% F-measure and 3.14 FPS. Our model achieves impressive improvement vs. baseline (1.5% in F-measure) with only a slight sacrifice in speed, which demonstrates the effectiveness of LEMNet.

5) Model Complexity: As shown in Table XIV, PE branch can enhance the model performance by 0.8% on F-measure with only 5.03 GFLOPs addition. Moreover, with the introduction of multi-head dependency modeling in CEM, LEMNet finally achieves 85.2% F-measure with 22.57 GFLOPs more than Baseline, which can achieve a better balance between F-measure and model complexity.

V. LIMITATION

LEMNet has reached superior performance on both multi-oriented and curved text detection tasks. However, there are still some limitations. As shown in Fig. 11, when two text centers almost overlap, LEMNet will fail to separately localize the two because of the mix of texture information, which also exists in many segmentation-based methods [11]. Moreover, LEMNet struggles with over exposure and badly illuminated text regions, owing to the unclear text boundaries. More corresponding researches are needed for addressing these challenging problems.

Fig. 11. Failure cases. Cyan polygons are detection results from LEMNet, while red polygons are ground truth.

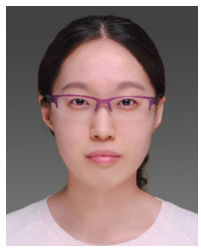
VI. CONCLUSION

In this paper, we propose an arbitrary-shaped scene text detector, called LEMNet, to handle the imprecise boundary problem in text segmentation results. LEMNet takes advantage of the strong feature expression ability of high dimensional space to constrain the boundary learning process in Pixel Embedding (PE) Branch. To further enhance the boundary feature distinguishability in complex background, a novel Context Enhancement Module (CEM) is introduced to enhance feature representation by capturing feature interactions in different representation subspaces. The comprehensive ablation studies demonstrate the effectiveness of PE and CEM. Furthermore, experimental results confirm that LEMNet can effectively detect arbitrary-shaped scene texts and achieve state-of-the-art performance. In the future, we will try and extend LEMNet to an end-to-end scene text reading system.

REFERENCES

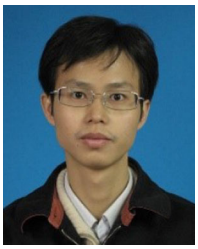
[1] J. Ma *et al.*, "Arbitrary-oriented scene text detection via rotation proposals," *IEEE Trans. Multimedia*, vol. 20, no. 11, pp. 3111–3122, Nov. 2018.
[2] X. Ren, Y. Zhou, J. He, K. Chen, X. Yang, and J. Sun, "A convolutional neural network-based chinese text detection algorithm via text structure modeling," *IEEE Trans. Multimedia*, vol. 19, no. 3, pp. 506–518, Mar. 2017.
[3] H. Xie, S. Fang, Z.-J. Zha, Y. Yang, Y. Li, and Y. Zhang, "Convolutional attention networks for scene text recognition," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 15, no. 1s, pp. 1–17, 2019.
[4] C. Xue, S. Lu, and W. Zhang, "MSR: Multi-scale shape regression for scene text detection," in *Proc. 28th Int. Joint Conf. Artif. Intell., 2019, S. Kraus*, Ed., ijcai.org, 2019, pp. 989–995.
[5] X. Wang, Y. Jiang, Z. Luo, C. Liu, H. Choi, and S. Kim, "Arbitrary shape scene text detection with adaptive text region representation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 6449–6458.
[6] Y. Wang, H. Xie, Z.-J. Zha, M. Xing, Z. Fu, and Y. Zhang, "ContourNet: Taking a further step toward accurate arbitrary-shaped scene text detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11 753–11 762.
[7] W. Wang *et al.*, "Shape robust text detection with progressive scale expansion network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 9336–9345.
[8] S. Long, J. Ruan, W. Zhang, X. He, W. Wu, and C. Yao, "TextSnake: A flexible representation for detecting text of arbitrary shapes," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 20–36.
[9] D. Deng, H. Liu, X. Li, and D. Cai, "PixelLink: Detecting scene text via instance segmentation," in *Proc. AAAI Conf. Artif. Intell.*, vol. 32, no. 1, 2018.

[10] Z. Tian *et al.*, "Learning shape-aware embedding for scene text detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4234–4243.
[11] M. Liao, Z. Wan, C. Yao, K. Chen, and X. Bai, "Real-time scene text detection with differentiable binarization," in *Proc. Assoc. Adv. Artif. Intell.*, 2020, pp. 11 474–11 481.
[12] Z. Tian, W. Huang, T. He, P. He, and Y. Qiao, "Detecting text in natural image with connectionist text proposal network," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 56–72.
[13] B. Shi, X. Bai, and S. Belongie, "Detecting oriented text in natural images by linking segments," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2550–2558.
[14] W. He, X.-Y. Zhang, F. Yin, and C.-L. Liu, "Deep direct regression for multi-oriented scene text detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 745–753.
[15] Y. Wang, H. Xie, Z.-J. Zha, Y. Tian, Z. Fu, and Y. Zhang, "R-Net: A relationship network for efficient and accurate scene text detection," *IEEE Trans. Multimedia*, vol. 23, pp. 1316–1329, May 19, 2020.
[16] 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 Process.*, vol. 28, no. 11, pp. 5566–5579, Nov. 2019.
[17] E. Xie, Y. Zang, S. Shao, G. Yu, C. Yao, and G. Li, "Scene text detection with supervised pyramid context network," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 9038–9045.
[18] Y. Liu, S. Zhang, L. Jin, L. Xie, Y. Wu, and Z. Wang, "Omnidirectional scene text detection with sequential-free box discretization," *Proc. 28th Int. Joint Conf. Artifi. Intell., IJCAI 2019*, 2019, pp. 3052–3058.
[19] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
[20] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[21] X. Zhou *et al.*, "EAST: An efficient and accurate scene text detector," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5551–5560.
[22] C. Zhang *et al.*, "Look more than once: An accurate detector for text of arbitrary shapes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 10 552–10 561.
[23] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2961–2969.
[24] H. Chen, K. Sun, Z. Tian, C. Shen, Y. Huang, and Y. Yan, "Blendmask: Top-down meets bottom-up for instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 8573–8581.
[25] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3431–3440.
[26] W. Liu *et al.*, "Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 21–37.
[27] W. Xu, H. Wang, F. Qi, and C. Lu, "Explicit shape encoding for real-time instance segmentation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 5168–5177.
[28] F. Wang, Y. Chen, F. Wu, and X. Li, "Textray: Contour-based geometric modeling for arbitrary-shaped scene text detection," in *Proc. 28th ACM Int. Conf. Multimedia*, 2020, pp. 111–119.
[29] Y. Zhou, H. Xie, S. Fang, Y. Li, and Y. Zhang, "CRNET: A center-aware representation for detecting text of arbitrary shapes," in *Proc. 28th ACM Int. Conf. Multimedia*, 2020, pp. 2571–2580.
[30] A. Newell, Z. Huang, and J. Deng, "Associative embedding: End-to-end learning for joint detection and grouping," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 2277–2287.
[31] A. W. Harley, K. G. Derpanis, and I. Kokkinos, "Learning dense convolutional embeddings for semantic segmentation," in *Proc. Int. Conf. Learn. Represent. (Workshop)*, 2015.
[32] H. Law and J. Deng, "Cornernet: Detecting objects as paired keypoints," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 734–750.
[33] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, "CenterNet: Keypoint triplets for object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 6569–6578.
[34] B. De Brabandere, D. Neven, and L. Van Gool, "Semantic instance segmentation with a discriminative loss function," 2017, *arXiv:1708.02551*.
[35] S. Kong and C. C. Fowlkes, "Recurrent pixel embedding for instance grouping," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 9018–9028.
[36] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2117–2125.
[37] J. Dai *et al.*, "Deformable convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 764–773.
[38] A. Vaswani *et al.*, "Attention is all you need," *Adv. Neural Inf. Process. Syst.*, vol. 30, pp. 5998–6008, 2017.
[39] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7794–7803.
[40] J. Fu *et al.*, "Dual attention network for scene segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3146–3154.
[41] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, and W. Liu, "CCNet: Criss-cross attention for semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 603–612.
[42] H. Wang *et al.*, "All you need is boundary: Toward arbitrary-shaped text spotting," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 12 160–12 167.
[43] M. Ester *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise." *Kdd*, vol. 96, no. 34, pp. 226–231, 1996.
[44] D. Karatzas *et al.*, "ICDAR 2015 competition on robust reading," in *Proc. 13th Int. Conf. Document Anal. Recognit.*, 2015, pp. 1156–1160.
[45] L. Yuliang, J. Lianwen, Z. Shuaitao, and Z. Sheng, "Detecting curve text in the wild: New dataset and new solution," 2017, *arXiv:1712.02170*.
[46] C. K. Ch'ng and C. S. Chan, "Total-text: A comprehensive dataset for scene text detection and recognition," in *Proc. 14th IAPR Int. Conf. Document Anal. Recognit.*, vol. 1, 2017, pp. 935–942.
[47] C. Yao, X. Bai, and W. Liu, "A unified framework for multioriented text detection and recognition," *IEEE Trans. Image Process.*, vol. 23, no. 11, pp. 4737–4749, Nov. 2014.
[48] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.
[49] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, 2015.
[50] W. Wang *et al.*, "Efficient and accurate arbitrary-shaped text detection with pixel aggregation network," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 8440–8449.
[51] A. Gupta, A. Vedaldi, and A. Zisserman, "Synthetic data for text localisation in natural images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2315–2324.
[52] 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.*, vol. 43, no. 2, pp. 532–548, Feb. 2021.
[53] W. Feng, W. He, F. Yin, X. Zhang, and C. Liu, "Textdragon: An end-to-end framework for arbitrary shaped text spotting," in *Proc. IEEE/CVF Int. Conf. Comput. Vision,*, 2019, pp. 9075–9084.
[54] Y. Baek, B. Lee, D. Han, S. Yun, and H. Lee, "Character region awareness for text detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 9365–9374.
[55] Y. Liu, L. Jin, Z. Xie, C. Luo, S. Zhang, and L. Xie, "Tightness-aware evaluation protocol for scene text detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 9612–9620.
[56] B. Shi *et al.*, "ICDAR 2017 competition on reading chinese text in the wild (RCTW-17)," in *Proc. 14th IAPR Int. Conf. Document Anal. Recognit.*, vol. 1, 2017, pp. 1429–1434.
[57] B. Shi, M. Yang, X. Wang, P. Lyu, C. Yao, and X. Bai, "Aster: An attentional scene text recognizer with flexible rectification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 9, pp. 2035–2048, Sep. 2019.
[58] D. Yu *et al.*, "Towards accurate scene text recognition with semantic reasoning networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 12 113–12 122.



Mengting Xing received the B.S. degree from XiDian University, Xi'an, China, in 2019. She is currently working toward the M.S. degree with the University of Science and Technology of China, Hefei, China. Her research interests include computer vision and signal processing.

Hongtao Xie received the Ph.D. degree in computer application technology from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2012. He is currently a Professor with the School of Information Science and Technology, University of Science and Technology of China, Hefei, China. His research interests include multimedia content analysis and retrieval, deep learning, and computer vision.



Qingfeng Tan (Member, IEEE) received the Ph.D. degree in information security from the University of Chinese Academy of Sciences, Beijing, China, in 2017. He is currently an Associate Professor with the Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou, China. His current research interests include anonymous communication and privacy protection.

Shancheng Fang received the Ph.D. degree in computer software and theory from the Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China, in 2020. He is currently a Postdoctoral Fellow with the School of Information Science and Technology, University of Science and Technology of China, Hefei, China. His research interests include multimedia analysis and computer vision.

Yuxin Wang received the B.S. degree from XiDian University, Xi'an, China, in 2018. He is currently working toward the Ph.D. degree with the School of Information Science and Technology, University of Science and Technology of China, Hefei, China. His research interests include computer vision and signal processing.

Zhengjun Zha received the B.E. and Ph.D. degrees from the University of Science and Technology of China, Hefei, China, in 2004 and 2009, respectively. He is currently a Full Professor with the School of Information Science and Technology, University of Science and Technology of China, the Vice Director of National Engineering Laboratory for Brain-Inspired Intelligence Technology and Application. He was a Researcher with the Hefei Institute of Physical Science, Chinese Academy of Sciences, from 2013 to 2015, a Senior Research Fellow with the School of Computing, National University of Singapore (NUS), Singapore, from 2011 to 2013, and a Research Fellow with the National University of Singapore from 2009 to 2010. He has authored or coauthored more than 100 papers, with a series of publications on top journals and conferences, in his research interests, which include multimedia analysis, retrieval and applications, and computer vision. Prof. Zha was the recipient of multiple paper awards from prestigious multimedia conferences, including the Best Paper Award and Best Student Paper Award in ACM Multimedia.

Yongdong Zhang (Senior Member, IEEE) received the Ph.D. degree in electronic engineering from Tianjin University, Tianjin, China, in 2002. He is currently a Professor with the School of Information Science and Technology, University of Science and Technology of China, Hefei, China. He has authored more than 100 refereed journal and conference papers. His research interests include multimedia content analysis and understanding, multimedia content security, video encoding, and streaming media technology. Prof. Zhang was the recipient of the Best Paper Awards in PCM 2013, ICIMCS 2013, and ICME 2010, and the Best Paper Candidate in ICME 2011. He is an Editorial Board Member of the *Multimedia Systems* and the IEEE TRANSACTIONS ON MULTIMEDIA.