SLOAN: Scale-Adaptive Orientation Attention Network for Scene Text Recognition

Pengwen Dai, Hua Zhang, and Xiaochun Cao, *Senior Member, IEEE*

***Abstract*— Scene text recognition, the final step of the scene text reading system, has made impressive progress based on deep neural networks. However, existing recognition methods devote to dealing with the geometrically regular or irregular scene text. They are limited to the semantically arbitrary-orientation scene text. Meanwhile, previous scene text recognizers usually learn the single-scale feature representations for various-scale characters, which cannot model effective contexts for different characters. In this paper, we propose a novel scale-adaptive orientation attention network for arbitrary-orientation scene text recognition, which consists of a dynamic log-polar transformer and a sequence recognition network. Specifically, the dynamic log-polar transformer learns the log-polar origin to adaptively convert the arbitrary rotations and scales of scene texts into the shifts in the log-polar space, which is helpful to generate the rotation-aware and scale-aware visual representation. Next, the sequence recognition network is an encoder-decoder model, which incorporates a novel character-level receptive field attention module to encode more valid contexts for various-scale characters. The whole architecture can be trained in an end-to-end manner, only requiring the word image and its corresponding ground-truth text. Extensive experiments on several public datasets have demonstrated the effectiveness and superiority of our proposed method.**

***Index Terms*— Scene text recognition, arbitrary orientation, log-polar transformation, attention mechanism, sequence-to-sequence learning.**

I. Introduction

SCENE text recognition is a fundamental and critical task in the computer vision community, as it is a key step in the scene text reading system. It plays a significant role in numerous practical applications, such as robot navigation [1], scene understanding [2], image retrieval [3], visual question answering [4], etc. Although scene text recognition has witnessed great progress in the deep learning era, it is still challenging due to the scene factors (e.g., cluttered background and uneven lighting) and the own characteristics of the scene text (e.g., various fonts and irregular layouts).

In the past years, many scene text recognition approaches have been proposed. However, most of them are not concerned with arbitrary-orientation scene texts in semantic, which extensively exist in many real-world applications. These semantically arbitrary-orientation scene texts mainly come from two aspects. One is that they are from natural images. The other is that they are generated by existing detectors. For example, the scene texts whose semantic orientations range in [90°, 270°], are not correctly detected by most scene text reading systems. As shown in the zoomed region of Fig. 1 (a), the geometric orientation α detected by [5] is inconsistent with the actual semantic orientation α'. Meanwhile, for the slant scene texts whose semantic orientations range in [0°, 90°) or (270°, 360°], existing scene text reading systems (e.g., [6]) will also generate inconsistency between the geometric orientation α and the semantic orientation α', as shown in Fig. 1 (b). When such detected texts are fed into the recognition branches of scene text reading systems, they would generate incorrect recognition results. Besides, different characters in these arbitrary-orientation scene texts may own various scales. For example, the scale of the character 'M' is different from that of 'i', as shown in Fig. 1 (a). These various scales of characters also affect the recognition accuracy of scene texts.

Recently, some methods [7]–[19] can recognize the oriented scene text, mainly including rectification-based methods and 2D-based methods. The rectification-based approaches [7]–[12] exploit rectification networks to rectify the oriented scene text to the approximately regular scene text, and then the rectified texts are fed into recognizers. However, these rectification networks need to initialize the pattern of fiducial points for the rectified image, which makes the learning of models need sophisticated skills and cannot effectively handle arbitrary-orientation scene texts. Meanwhile, the 2D-based methods [13]–[18] utilize the fully convolutional network to extract the 2D visual representations, and then decode the semantic order of the scene text from left to right in default. However, these 2D-based methods are not effective for the arbitrary-orientation scene text, especially for the oriented scene texts whose semantic orientations range in [90°, 270°]. Although an arbitrary orientation network [19] is proposed to

Manuscript received March 27, 2020; revised September 11, 2020; accepted December 7, 2020. Date of publication December 23, 2020; date of current version January 14, 2021. This work was supported in part by the National Key Research and Development Program of China under Grant 2020YFB1406704, in part by the National Natural Science Foundation of China under Grant 61733007, Grant U1736219, Grant U1803264, and Grant 62072454, in part by the Beijing Education Committee Cooperation Beijing Natural Science Foundation under Grant KZ201910005007, in part by the Peng Cheng Laboratory Project of Guangdong Province under Grant PCL2018KP004, and in part by the Beijing Natural Science Foundation under Grant 4202084. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Shuicheng Yan. *(Corresponding author: Xiaochun Cao.)*

Pengwen Dai is with the State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100195, China, and also with the School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: daipengwen@iie.ac.cn).

Hua Zhang is with the State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100195, China (e-mail: zhanghua@iie.ac.cn).

Xiaochun Cao is with the State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100195, China, also with the Peng Cheng Laboratory, Cyberspace Security Research Center, Shenzhen 518055, China, and also with the School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: caoxiaochun@iie.ac.cn).

Digital Object Identifier 10.1109/TIP.2020.3045602

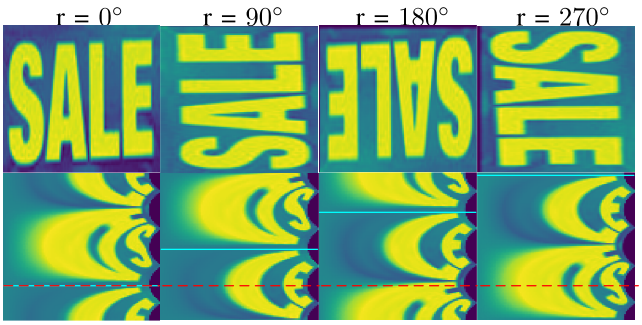
(a) (b)

Fig. 1. Examples of arbitrary-orientation scene text recognition using the scene text reading systems [5] (a) and [6] (b). The orange regions display the recognition results. The cyan boxes are the zoomed text instances. The yellow and green points are the first and fourth corner points of the detected bounding box respectively. The black arrow indicates the detected geometric orientation presented as the angle α. The blue arrow indicates the semantic orientation presented as the angle α'. The dashed line means the left-to-right horizontal direction. Note that we do not show all results for clear presentations.

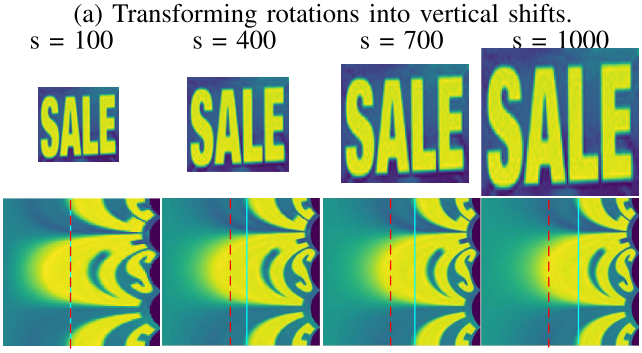
recognize the arbitrary-orientation scene text by combining the weighted four-directional features, it is not easy to learn the rotation-aware and scale-aware representation.

Additionally, for the characters with different scales in the scene texts, existing sequence-based methods [7]–[14], [19]–[27] usually do not consider these character-level scale variations. In the recognition network, the generated sequence feature is associated with the rectangle region called the receptive field [21] in the input image. Theoretically, the size of the receptive field relies on the structure of the network. It is fixed and single-scale for previous scene text recognizers. Employing such a fixed-size and single-scale receptive field is not suitable for all kinds of characters. When the receptive field is small, it may cover a part of the large character, which cannot capture the discriminative spatial contexts. When the receptive field is large, it may cover several characters, which cannot capture the discriminative details. Therefore, this fixed-size and single-scale receptive field is disadvantageous to encode the discriminative feature representations for the individual character with different scales.

Based on the above analyses, in this paper, we create a **S**ca**L**e-adaptive **O**rientation **A**ttention **N**etwork, called **SLOAN**, to detect arbitrary-orientation scene texts. In SLOAN, we introduce the log-polar transformation to solve the sequence recognition of arbitrary-orientation scene texts. The log-polar transformation can convert the arbitrary rotation and scale of the entire scene text in the Cartesian coordinate system into the vertical and horizontal shift in the log-polar space, as shown in Fig. 2. The polar transformation has been employed in many tasks (e.g., medical image segmentation [28], image registration [29], image classification [30], etc.) to flatten the circular regions or pursue the rotation/scale invariance. However, our SLOAN develops the Dynamic Log-Polar Transformer (DLPT) to attend to arbitrary orientations of scene texts by learning the rotation-aware and scale-aware feature representations, based on the predicted log-polar origin. Different from [31] that utilizes the log-polar transformation to extract artificially-designed features for recognizing the texts on the scanned image, our DLPT can handle more complicated scenarios. Besides, to further



(a) Transforming rotations into vertical shifts.



(b) Transforming scalings into horizontal shifts.

Fig. 2. Illustration of properties of the log-polar transformation. It can convert the rotation and scaling in the Cartesian coordinate system into the shifts in the log-polar space. (a) is the rotated images and the corresponding log-polar images. (b) is the scaled images and the corresponding log-polar images. Red lines denote reference lines, while cyan lines indicate the target lines shifted from the reference lines. The distances between the red and cyan lines correspond to the rotation or scaling factors.

model the scale variations of individual characters, we propose a novel Character-level Receptive Field Attention (CRFA) to learn the scale-adaptive representations for characters. The CRFA module can associate different receptive fields with the sequence features, and then adaptively focuses on useful contexts based on the learned weights. More specifically, as illustrated in Fig. 3, DLPT first converts the input image into the feature representations in the log-polar space. Then, the feature encoder incorporated with the CRFA module is utilized to generate sequence features. Finally, the sequence features are decoded to the text strings by the attention-based sequence decoder.

The contributions of this paper are summarized as follows:

i) We devote to recognizing scene texts with a wider range of orientations, and collect a new dataset to support the semantically arbitrary-orientation scene text detection, recognition and spotting.

ii) We introduce a dynamic log-polar transformer to learn the rotation-aware and scale-aware features of arbitrary-orientation scene texts with a weakly-supervised strategy, which can significantly improve the recognition performance.

iii) A character-level receptive field attention mechanism is developed to learn more discriminative representations for individual characters with various scales, which is a scale-adaptive technique and can effectively achieve higher accuracy.

The rest of the paper is organized as follows. Section II introduces the related work in detail. Section III will elaborate

on the proposed method. Then, numerous experiments are conducted and the experimental results are described in Section IV and V. Finally, Section VI concludes the paper.

II. Related Work

Scene text recognition has been widely studied for many years. Comprehensive surveys can be found in [32], [33]. In this section, we carefully review the related work on the scene text recognition.

A. Scene Text Recognition

1) Regular Scene Text Recognition: Many inspiring and novel methods for recognizing the regular scene text (e.g., horizontal and frontal texts) have emerged widely. They are roughly divided into three types: character-based methods, word-based methods and sequence-based methods.

The character-based methods [34]–[36] first localize the candidate characters, and then recognize these candidates. In the stage of localizing candidate characters, some representative approaches are proposed. For example, Neumann *et al.* exploit maximally stable extremal regions (MSERs) [34] to detect character components, while Bai *et al.* [35] localize the characters based on the strokelets. Besides, a generative shape model [36] is developed by a small number of clear images to extract characters. In the stage of recognizing candidate characters, the language models or the heuristic algorithms are usually employed to generate word-level recognition results. However, the character-based methods cannot provide accurate character locations due to the cluttered background or the insufficient space between consecutive characters.

The word-based methods [37], [38] directly formulate a 90k-class classification task based on a powerfully convolutional neural network, where each class denotes one English word. However, this model cannot recognize the words that are out of the predefined vocabularies.

The sequence-based methods [20]–[27] simultaneously perform character localization and word recognition. In such a framework, the character localization is an implicit step. These methods first encode the input image into sequence representations and then decode the representations to text strings. In the decoder network, they usually involve two major techniques: the connectionist temporal classification (CTC) and the attention mechanism. For the CTC-based methods [21]–[23], they utilize the CTC to calculate the conditional probability between the prediction and the target sequence based on all mapping paths. For the attention-based methods [24], [25], they employ the attention mechanism to learn the mapping between the encoding feature sequences and the target text strings. However, attention-based methods may cause misalignments. To alleviate the attention drift problem, Cheng *et al.* [25] introduce the focusing network to adjust the attention sequences, while Bai *et al.* [27] propose the edit probability loss for better learning the alignments. Besides, in the sequence-based methods, the recurrent neural network (RNN) is utilized, which does not be computed in parallel. Thus Fang *et al.* [26] provide no-recurrence strategies instead of RNN for computation parallelization.

2) Irregular Scene Text Recognition: Most regular scene text recognition methods are not suitable for the irregular scene text (e.g., oriented, perspective or curved texts). Thus, some specialized approaches are proposed to recognize the irregular scene text. These methods can be roughly categorized into three groups: rectification-based methods, 2D-based methods and multi-direction encoding based methods.

The rectification-based methods [7]–[12] incorporate the rectification network into the recognition network to formulate an end-to-end trainable framework. Specifically, they exploit a network to rectify the irregular scene text to the regular scene text, and then employ the regular scene text recognition network to generate final text strings. For example, Shi *et al.* [9] utilize the thin-plate-spline (TPS) transformation to rectify the irregular scene text. Yang *et al.* [11] introduce the symmetry constraint in the rectification network for better rectifying the irregular scene text. Luo *et al.* [10] formulate a multi-object rectification network by deforming the offsets of each pixel. Instead of rectifying the entire input image, Liu *et al.* [12] propose a character-aware neural network to detect and rectify individual characters.

The 2D-based methods [13]–[18] would encode the input image to the 2D representations for better keeping the spatial information. For example, Yang *et al.* [13] exploit the 2D attention mechanism to learn better contexts for each spatial position, based on character-level annotations. Similarly, Li *et al.* [14] utilize a more simple 2D attention mechanism to achieve better performance without character-level annotations. In [15] and [16], the authors formulate the irregular scene text recognition as a 2D segmentation that predicts a class for each spatial position. Besides, Wan *et al.* [17] not only perform the pixel-level semantic segmentation but also generate the position and order maps of characters in the 2D features. Different from directly performing the 2D segmentation, the scholars [18] also utilize the RNN-based decoder to generate text strings based on the 2D attention map and the semantic segmentation map.

The multi-direction encoding based method [19] first exploits the CNN-RNN framework to encode the input image into the representations in four (left-to-right, right-to-left, top-to-bottom and bottom-to-top) directions. Then, a directional attention mechanism is designed to fuse these four directional features. These fused features can represent the arbitrary-orientation scene text.

In this paper, we propose a scale-adaptive orientation attention network to recognize the arbitrarily-orientation scene text, which not only captures orientation variations but also learns scale-adaptive representations.

B. End-to-End Scene Text Recognition System

An end-to-end scene text recognition system, also called scene text spotting, consists of a detector and recognizer in a sequential manner [9]. It contains two mainstreams: two-stage methods and one-stage methods.

The two-stage methods [5], [39]–[42] first localize the scene text, and then recognize the cropped images from the detected bounding boxes. For example, Liao *et al.* [5] propose

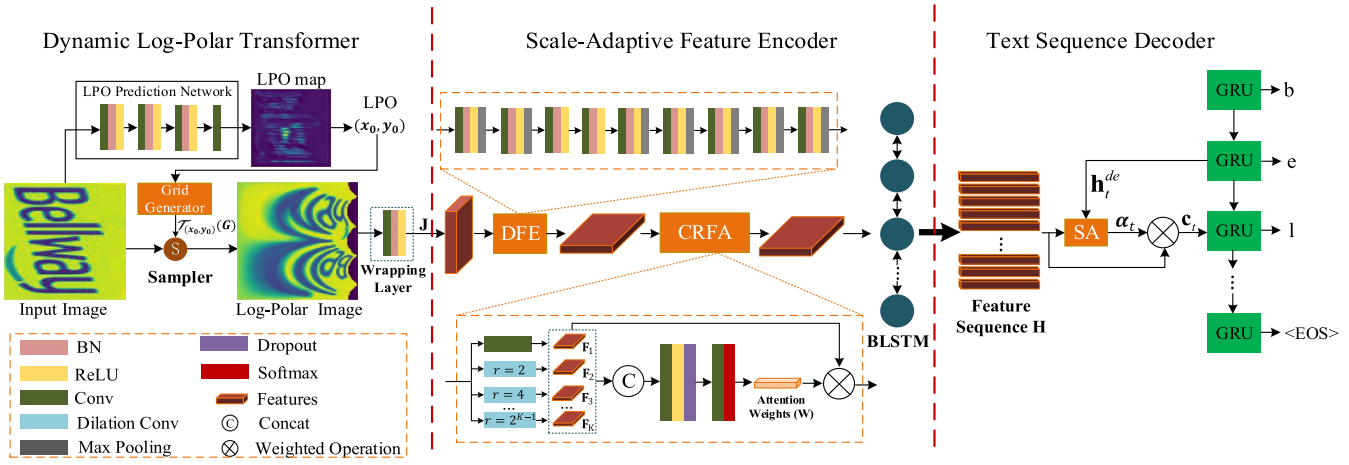


Fig. 3. The overall framework of our proposed method. Given a gray-scale input image, the log-polar origin (LPO) prediction network generates the LPO map. After obtaining LPO based on the LPO map, the grid generator converts the spatial grid G over the log-polar image to the sampling grid $\mathcal{T}_{(x_0,y_0)}(G)$ over the input image. Then, the sampler determines the value of the log-polar image at each position before generating the log-polar representation $\mathbf{J}$ via a wrapping layer. Subsequently, in the scale-adaptive feature encoder network, the deep feature extractor (DFE) takes the log-polar representation as the input to obtain deep features, and then the character-level receptive field attention (CRFA) mechanism is utilized to adaptively learn more representative features. After that, the enhanced features are fed into BLSTM to further encode contexts and form the feature sequence. Finally, based on the feature sequence $\mathbf{H}$, the text sequence decoder network employs the sequence attention (SA) mechanism to generate the character sequence step by step.

a multi-oriented scene text detection method based on the SSD framework [43], and then utilize an off-the-shelf recognition method CRNN [21] to recognize the detected bounding boxes. In effect, any detectors (e.g., [44]–[47]) and recognizers (e.g., [9], [15], [21]) can be integrated to form an end-to-end scene text recognition system. It is flexible when spotting multi-language scene texts, as the recognizers can be trained for different languages and achieve better performance under diverse language models.

The one-stage methods [6], [48]–[52] integrate detection and recognition into an end-to-end framework. The recognition network would take the corresponding features of the detected bounding boxes as the inputs, thus the mappings from the detections to the features play a crucial role. For example, Busta *et al.* [48] utilize a region-of-interest (ROI) pooling approach to map the multi-oriented proposals into the fixed-height representations, before the extracted features are fed into a CTC-based recognition network. He *et al.* [50] propose a text alignment layer to alleviate the misalignments between the multi-oriented proposals and the features, before the extracted features are fed into an attention-based recognition network. Liao *et al.* [6] employ the ROIAlign technique to perform the feature mappings, before the mapped features are fed into a 2D character-wise segmentation network and attention network to perform recognition. Besides, Liu *et al.* [51] develop a BezierAlign strategy to project the features of arbitrary-shape text into the fixed-size feature map. Different from performing feature sampling on the entire texts, Feng *et al.* [52] extract the features of arbitrary-shape texts based on the local quadrangle in a sliding manner. In effect, the detection and recognition can benefit from each other in one-stage methods, but they are not flexible for the multi-language scene text.

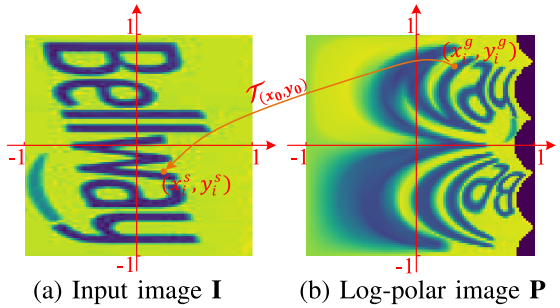
Although in this paper we focus on the scene text recognition, we show that our proposed model can help maintain the robustness of recognition for the arbitrary-orientation scene text and achieve state-of-the-art performances in the end-to-end recognition system, even without using the strongest detector. These properties make our recognition method appealing in many practical scenarios.

III. Methodology

Our proposed model consists of three modules: the dynamic log-polar transformer, the scale-adaptive feature encoder and the text sequence decoder. Specifically, given an input image, we first construct a dynamic log-polar transformer to transform the image in the gray space to the feature representations in the log-polar space. Then, the log-polar representations are fed into the scale-adaptive feature encoder to achieve the sequence representations by using the character-level receptive field attention. Finally, the sequence representations are given as input for the text sequence decoder network to recognize the text. The overall architecture of our method is presented in Fig. 3, which can be trained in an end-to-end fashion.

A. Dynamic Log-Polar Transformer

To construct the dynamic log-polar transformer (DLPT), the critical parameter is the original point, which needs to be predefined. Different from traditional methods using the center of the image as the origin, we propose a log-polar origin (LPO) prediction network to determine the original point. The advantage of introducing the dynamic log-polar original point lies that it can adaptively convert the arbitrary rotations and scales into the shifts in the log-polar space, which facilitates the visual representation extracting. Besides, the recognition of scene texts are mainly dependent on the shapes of characters, so we first convert the original RGB image into the gray-scale image $\mathbf{I} \in \mathbb{R}^{H \times W}$ (H and W are the height and width of the image). Compared with the RGB image, the gray-scale image can relieve the influence of the appearance of characters, effectively suppress the interference



(a) Input image **I** (b) Log-polar image **P**

Fig. 4. Illustration of the log-polar transformation. The orange arrow denotes the coordinate mapping $\mathcal{T}_{(x_0,y_0)}$ from the point (x_i^g, y_i^g) in the log-polar space to the point (x_i^s, y_i^s) in the Cartesian coordinate system. The value of (x_i^g, y_i^g) comes from the nearby points of (x_i^s, y_i^s) via a sampler.

of backgrounds, and reduce the cost of DLPT. Then, the LPO prediction network (LPOPN) takes **I** as the input to generate the LPO heatmap $\mathbf{O} \in \mathbb{R}^{H/2 \times W/2}$, which is formulated as:

$$\mathbf{O} = LPOPN(\mathbf{I};\ \Theta_{lpo}), \tag{1}$$

where Θ_{lpo} denotes the trainable parameters. To facilitate the calculation of log-polar transformation, we set the center coordinate of the heatmap **O** as (0, 0) and normalize the coordinates of **O** to [-1, 1], forming the x-coordinate map $\mathbf{E}^x$ and y-coordinate map $\mathbf{E}^y$. Thus, the coordinate (x_0, y_0) of LPO can be expressed as:

$$x_0 = \frac{\sum_i \mathbf{O}_i \cdot \mathbf{E}_i^x}{\sum_i \mathbf{O}_i},\ y_0 = \frac{\sum_i \mathbf{O}_i \cdot \mathbf{E}_i^y}{\sum_i \mathbf{O}_i}, \tag{2}$$

where i denotes the index of position.

After the log-polar original point (x_0, y_0) is determined, we can transform the gray-scale image **I** into the log-polar image $\mathbf{P} \in \mathbb{R}^{H \times W}$, as shown in Fig. 4. Specifically, since the coordinate ranges in the log-polar space are different from those in the gray-scale image, we first normalize the coordinates of **I** and **P** to the same scope [-1, 1]. Next, we employ the grid generator to obtain the target grid $G = \{(x_i^g, y_i^g)\}_{i=1}^{HW}$, where (x_i^g, y_i^g) is the i-th point over **P**. After that, the sampling grid $\mathcal{T}_{(x_0,y_0)}(G) = \{(x_i^s, y_i^s)\}_{i=1}^{HW}$ over **I** can be calculated as:

$$x_i^s = x_0 + \rho_i^g \cdot cos(\theta_i^g), \tag{3}$$

$$y_i^s = y_0 + \rho_i^g \cdot sin(\theta_i^g), \tag{4}$$

where x_i^s and y_i^s denote the coordinates of the input image **I** at the i-th location. ρ_i^g and θ_i^g denote the log-polar radius and angle of the log-polar image at the i-th location, which can be calculated as:

$$\rho_i^g = \frac{\rho - 1}{\gamma - 1} \cdot \frac{2\gamma}{W},\ \rho = \frac{x_i^g + 1}{2} \cdot log\gamma, \tag{5}$$

$$\theta_i^g = (y_i^g + 1) \cdot \pi, \tag{6}$$

where γ represents the maximum distance to LPO, which is set to $\sqrt{H^2 + W^2}/2$. Eq. (5) and Eq. (6) are utilized to map the range of the coordinate from [-1, 1] to $[0, 2\gamma / W]$ and $[0, 2\pi]$, respectively. Finally, the value of each point (x_i^g, y_i^g) is sampled from the input image **I**, which is formulated as:

$$\mathbf{P}_i = \sum_{n=1}^{H} \sum_{m=1}^{W} \mathbf{I}_{nm}\, \psi(\frac{x_i^s + 1}{2} W,\ m)\, \psi(\frac{y_i^s + 1}{2} H,\ n), \tag{7}$$

where $\mathbf{I}_{nm}$ denotes the value at location (n, m) of the input image. $\psi(\cdot, \cdot)$ indicates the bilinear sampling kernel, which can be expressed as:

$$\psi(a, b) = max(0, 1 - |a - b|), \tag{8}$$

where a and b are two inputs.

In the Cartesian coordinate system, the angle of rotation is periodic. It corresponds to vertical shifts on the log-polar image **P**, but will break off when the angle is 0 or 2π. Directly utilizing the traditional padding to keep the resolution in the convolution process does not model the periodicity of angle on **P**. Therefore, we employ the wrap-round padding [30] along the vertical direction to further construct a wrapping layer for generating more representative log-polar features **J**. Specifically, the bottom-most rows of **P** are first padded with the top-most rows and vice versa. Then, the padded **P** is fed into a 3×3 convolutional layer with M filters followed by the ReLU and batch normalization. Thus, the log-polar representation $\mathbf{J} \in \mathbb{R}^{H \times W \times M}$ can be expressed as:

$$\mathbf{J} = \mathcal{F}(\mathbf{P};\ \Theta_{wrap}), \tag{9}$$

where $\mathcal{F}$ denotes the wrapping process. Θ_{wrap} indicates the corresponding trainable parameters.

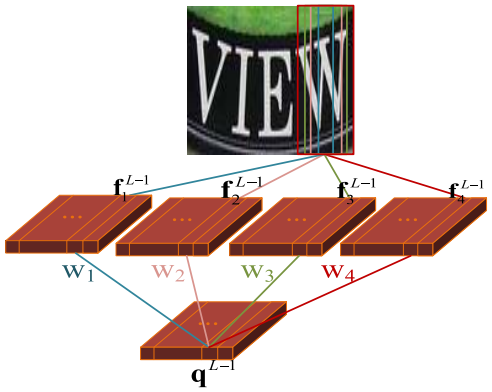


Fig. 5. Principle of CRFA. When $K = 4$, the multi-scale features $\mathbf{f}_1^{L-1}$, $\mathbf{f}_2^{L-1}$, $\mathbf{f}_3^{L-1}$ and $\mathbf{f}_4^{L-1}$ are associated with different receptive fields. They are fused to obtain a more representative feature $\mathbf{q}^{L-1}$ by the learnable weights w_1, w_2, w_3 and w_4.

B. Scale-Adaptive Feature Encoder

In the feature encoder network, the log-polar representation **J** is first fed into the deep feature extractor (DFE) to generate more representative 1D feature $\mathbf{F} \in \mathbb{R}^{1 \times L \times D}$, which is formulated as:

$$\mathbf{F} = DFE(\mathbf{J};\ \Theta_{dfe}), \tag{10}$$

where Θ_{dfe} is the trainable parameters.

To achieve the adaptive receptive field for the various-scale character, we introduce a character-level receptive field attention (CRFA) mechanism. As shown in Fig. 5, the activations in the multi-scale feature maps $\{\mathbf{F}_i\}_{i=1}^{K}$ ($\mathbf{F}_i \in \mathbb{R}^{1 \times L \times D'}$) are

associated with different sizes of receptive fields $\{R_i^0\}_{i=1}^K$, which can be calculated by the recursive formula:

$$R_i^l = (R_i^{l+1} - 1) \cdot s_i^l + r_i^l \cdot (\kappa_i^l - 1) + 1, \tag{11}$$

where R_i^l denotes the receptive field size at the l-th layer for the i-th scale. s_i^l, r_i^l and κ_i^l are the corresponding stride size, dilation rate and kernel size. For the standard convolution layer and the pooling layer, the dilation rate r_i^l is fixed to 1.

The multi-scale features are generated by using different dilation rates based on the convolution filters, and then they are fused according to the learned attention weight. Specifically, the attention weight $\mathbf{W} \in \mathbb{R}^{L \times K}$ is formulated as:

$$\mathbf{W} = CRFA(\mathbf{F}; \Theta_{crfa}), \tag{12}$$

where Θ_{crfa} means the trainable parameters.

Subsequently, we set $\mathbf{F}_i$ to $\{\mathbf{f}_i^j\}_{j=1}^L$, where $\mathbf{f}_i^j \in \mathbb{R}^{D'}$ denotes a feature vector. $\mathbf{W}$ is set to $\{\mathrm{w}_i^j\}_{j=1}^L$, where w_i^j is a scalar. $\mathbf{f}_i^j$ can be regarded as the representation corresponding to the i-th kind of receptive field at the j-th location in feature maps. When the receptive field captures the information of the character well, the corresponding weight should be larger. Therefore, the adaptively representative feature can be calculated as:

$$\mathbf{q}^j = \sum_{i=1}^{K} \mathrm{w}_i^j \cdot \mathbf{f}_i^j, \ \forall\ j = 1, \ldots, L, \tag{13}$$

where $\mathbf{q}^j \in \mathbb{R}^{D'}$ denotes the attention feature vector at the j-th location.

Based on the scale-adaptive feature $\mathbf{Q} = \{\mathbf{q}^j\}_{j=1}^L$, a single BLSTM with D' hidden units is used to encode the contexts among different locations and generate the feature sequence $\mathbf{H} \in \mathbb{R}^{L \times D'}$.

C. Text Sequence Decoder

In the sequence decoder network, we represent the feature sequence $\mathbf{H}$ as $\{\mathbf{h}_j^{en}\}_{j=1}^L$, where $\mathbf{h}_j^{en} \in \mathbb{R}^{D'}$. The sequence decoding involves T steps. At step t, the sequence attention weight $\boldsymbol{\alpha}_t \in \mathbb{R}^L$ is calculated as:

$$\boldsymbol{\alpha}_t = SA(\mathbf{h}_{t-1}^{de}, \ \mathbf{H}), \tag{14}$$

where $\mathbf{h}_{t-1}^{de}$ is the gated recurrent unit (GRU) hidden state of the decoder at the (t-1)-th step. SA denotes the sequence attention (SA) mechanism, which is formulated as:

$$\boldsymbol{e}_{t,j} = \mathbf{W}_e^\top tanh(\mathbf{W}_s \mathbf{h}_{t-1}^{de} + \mathbf{W}_h \mathbf{h}_j^{en} + \mathbf{b}), \tag{15}$$

$$\boldsymbol{\alpha}_{t,j} = \frac{\boldsymbol{e}_{t,j}}{\sum_{j=1}^{L} \boldsymbol{e}_{t,j}}, \tag{16}$$

where $\mathbf{W}_e$, $\mathbf{W}_s$, $\mathbf{W}_h$ and $\mathbf{b}$ are trainable parameters.

The sequence attention weight $\boldsymbol{\alpha}_t$ reveals the influence of each $\mathbf{h}_j^{en}$ based on the hidden state $\mathbf{h}_{t-1}^{de}$. To focus on more valid features, the weighted operation is performed on the feature sequence as:

$$\mathbf{c}_t = \sum_{j=1}^{L} \boldsymbol{\alpha}_{tj} \cdot \mathbf{h}_j^{en}, \tag{17}$$

where $\mathbf{c}_t \in \mathbb{R}^{D'}$ denotes the weighted feature, which is more representative for the sequence decoding.

Then the current hidden state $\mathbf{h}_t^{de}$ of GRU is updated as:

$$\mathbf{h}_t^{de} = GRU(\mathbf{h}_{t-1}^{de}, \mathbf{c}_t, \mathbf{y}_{t-1}), \tag{18}$$

where $\mathbf{y}_{t-1}$ means the ground-truth label in the training stage and the prediction in the inference stage.

Finally, the output $\mathbf{y}_t \in \mathbb{R}^{C+1}$ is formulated as:

$$\mathbf{y}_t = softmax(\mathbf{V}^\top \mathbf{h}_t^{de}), \tag{19}$$

where $\mathbf{V}^\top$ is the trainable parameters. $C+1$ means the number of target classes with the end-of-sequence (EOS) token.

D. Training and Inference

Our proposed framework can be trained only with the images and the corresponding text strings. The loss function of the model is formulated as:

$$\mathcal{L} = -\sum_t \ln p(\mathbf{y}_t^* | \mathbf{I}; \ \Theta), \tag{20}$$

where $\mathbf{y}_t^*$ is the ground-truth label. Θ denotes all trainable parameters in the network. The probability $p(\cdot)$ is calculated based on Eq. (19).

In the inference stage, when it meets the EOS symbol or exceeds the maximum decoding step T, the decoding will be end. Otherwise, at each step, the predicted character will be greedily generated based on the highest score. Alternatively, the heuristic algorithm beam search [53] with the beam width of β is employed to promote recognition accuracies. Specifically, at each decoding step, top-β accumulated log-likelihoods are selected instead of choosing the highest probability. However, larger beam widths will result in lower decoding speed.

IV. Experiments

In this section, extensive experiments are conducted on the public datasets to validate the effectiveness of our proposed model and the outstanding performances compared with other typical methods.

A. Datasets and Evaluation Protocols

Synth90k [37] is a synthetic text dataset that contains about 9 million word images in total. Each word image is generated by rendering the word, that comes from a set of 90k common English words, onto the natural images with some transformations. This word also acts as the annotation.

SynthText [54] is a synthetic text dataset for scene text detection originally. This dataset contains about 8k images and 4 million text instances. When the dataset is used for scene text recognition, it needs to crop the text region.

IIIT5K [55] is collected from the Internet. It contains 5,000 images, in which 3,000 images are used as the testing set and the rest serves as the training set. Each image in this dataset is equipped with two kinds of lexicons (50-word lexicon and 1k-word lexicon).

SVT [56] contains 647 testing images that are cropped from 250 full scene images. These full images come from the Google Street View, and many of them are low resolutions or are corrupted by blur and noise. Each cropped word image is also equipped with a 50-word lexicon.

Fig. 6. Examples of annotations in the dataset ASOT. The yellow and green points denote the first and fourth semantic points, respectively. Note that the annotations of word transcripts are not displayed.

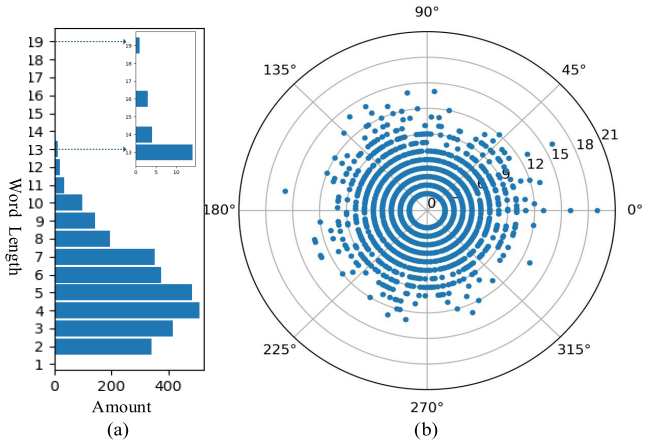


Fig. 7. (a) is the amount of text instances against word lengths. (b) is the distribution of semantic orientations and word lengths of text instances (blue) in the dataset ASOT.

ICDAR03 [57] contains 1,007 cropped word images from 251 full scene images. The texts in the scene images are focused by the camera when building this dataset. Each word image is associated with two kinds of lexicons, e.g., full-word lexicon and 50-word lexicon.

ICDAR13 [58] contains 1,095 testing word images cropped from 229 images. Most of them are from ICDAR03, and new word images are added into this dataset that is not equipped with the lexicon.

ICDAR15 [59] contains 2,077 testing word images. These images are cropped from the 500 incidental scene images that are captured by the Google Glasses without careful focusing. Thus there are many blur and low-resolution word images in this dataset associated with no lexicon. When this dataset is used for scene text spotting, it is associated with three lexicons (e.g., strong lexicon, weak lexicon and generic lexicon).

ASOT[1] is our newly collected challenging dataset about Arbitrarily Semantic-Orientation Texts in the natural image. It contains 406 images with 3,001 scene text instances, excluding the text instance marked as '###'. Each scene text instance is annotated by four corner points and the corresponding word. Some examples are shown in Fig. 6. The semantic orientations of text instances are evenly ranged in [0°, 360°), and the word lengths of text instances are also various. Their statistical information is illustrated in Fig. 7. When this dataset is used for recognition, it is associated with a full-word lexicon. For the text spotting, it involves a weak lexicon.

[1]This dataset is avilable at https://drive.google.com/open?id=1Rce7A18BhqrvWfKAOjSbQiad91Mn0ZfZ

TABLE I

SUMMARIES OF MAIN NETWORK CONFIGURATIONS. '#FILTERS', 'κ', 'S' AND 'P' DENOTE THE NUMBER OF FILTERS, KERNEL SIZE, STRIDE SIZE AND PADDING SIZE. FOR EXAMPLE, 'κ:3 × 3' MEANS 3 × 3 KERNEL SIZE. BOTH THE STRIDE SIZE AND PADDING SIZE ARE 1 × 1 IN DEFAULT. 'D-CONV' DENOTES DILATION CONVOLUTION AND 'R' IS THE DILATION RATE

Network	Type	Configurations
LPO Prediction	Conv	#filters:20, κ:3×3, s:2×2
	Conv	#filters:20, κ:3×3
	Conv	#filters:20, κ:3×3
	Conv	#filters:1, κ:1×1, p:0×0
DFE	Conv	#filters:64, κ:3×3
	MaxPooling	κ:2×2, s:2×2, p:0×0
	Conv	#filters:128, κ:3×3
	MaxPooling	κ:2×2, s:2×2
	Conv	#filters:256, κ:3×3
	Conv	#filters:256, κ:3×3
	Conv	#filters:512, κ:3×3
	MaxPooling	κ:2×2, s:2×1, p:1×0
	Conv	#filters:512, κ:3×3
	MaxPooling	κ:2×2, s:2×1, p:0×1
	Conv	#filters:512, κ:3×3
	MaxPooling	κ:2×2, s:2×1, p:1×0
	Conv	#filters:512, κ:3×3
	MaxPooling	κ:2×2, s:2×1, p:0×0
	Conv	#filters:512, κ:3×3
	MaxPooling	κ:2×2, s:2×1, p:0×0
CRFA	Conv	#filters:256, κ:1×1, p:0×0
	D-Conv	#filters:256, κ:3×3, r=2
	D-Conv	#filters:256, κ:3×3, r=4
	D-Conv	#filters:256, κ:3×3, r=8
	Conv	#filters:256, κ:3×3
	Conv	#filters:4, κ:1×1, p:0×0

In evaluation of the single recognition model, following previous methods, the case-insensitive word accuracy (CIWA) is employed as the evaluation metric. In the end-to-end recognition system, we employ two frequently-used evaluation metrics, namely, 'Word-Spotting' and 'End-to-End'. These two protocols are very similar, except that the former ignores symbols, numbers, and words whose length is less than 3. Besides, following [60], we also utilize the true positives, false positives and false negatives for evaluation.

B. Implementation Details

In our network, the original image is first converted into the gray-scale image **I**, whose height H and width W are fixed to

TABLE II

ABLATION STUDIES ON THE TESTING SET OF IIIT5K. THE REPORTED PERFORMANCE DENOTES THE CASE-INSENSITIVE WORD ACCURACY (%). 'AVG' INDICATES THE AVERAGE ACCURACY. THE BEAM WIDTH β IS FIXED TO 1. 'LEXICON@0' MEANS NO PROVIDED LEXICON WHILE 'LEXICON@1K' INDICATES THAT THE PROVIDED LEXICON SIZE IS 1K

Method		Lexicon@0						Lexicon@1k					
		0°	90°	180°	270°	avg	FPS	0°	90°	180°	270°	avg	FPS
Baseline-A		68.54	66.93	67.35	67.01	67.46	7.02	84.33	84.01	83.68	83.41	83.86	6.87
Baseline-B		74.17	74.17	74.17	74.17	74.17	1.80	87.53	87.53	87.53	87.53	87.53	1.78
AON [19]		72.84	71.55	72.45	73.69	72.63	26.96	90.02	89.71	89.83	89.83	89.85	24.81
DLPT	CRFA												
×	×	62.58	69.13	70.34	66.45	67.13	**28.91**	82.12	88.78	86.91	85.89	85.93	**27.35**
√	×	73.97	75.60	74.08	75.41	74.76	26.75	**91.54**	**92.28**	91.23	91.86	91.73	24.65
×	√	71.08	68.28	68.16	65.23	68.19	27.65	88.50	88.04	84.92	83.94	86.35	25.43
√	√	**75.06**	**76.34**	**74.47**	**75.91**	**75.45**	25.53	91.04	**92.28**	**91.62**	**92.09**	**91.75**	23.96

TABLE III

EFFECT OF THE NUMBER OF KERNEL FILTERS IN WRAPPING LAYER. THE REPORTED PERFORMANCE IS THE CASE-INSENSITIVE WORD ACCURACY (%). THE RESULTS ARE EVALUATED ON THE TESTING SET OF IIIT5K WITHOUT USING LEXICON. THE BEAM WIDTH IS SET TO $\beta = 1$

#Filters	0°	90°	180°	270°	avg
M = 0	72.72	73.50	71.51	73.38	72.78
M = 8	73.38	75.37	74.38	72.84	74.00
M = 16	**73.97**	**75.60**	**74.08**	75.41	**74.76**
M = 32	73.67	75.18	73.47	**75.45**	74.44
M = 64	73.27	75.45	74.01	75.18	74.48

100. The number of filters M is fixed to 16. The number of scales K in CRFA is set to 4. The dimensions L, D and D' are fixed to 23, 512 and 256, respectively. The number of classes C+1 is set to 37 (10 digits, 26 letters and 1 EOS symbol). The main configurations of the network are detailedly displayed in Table I. In the training, the shortest size of the original image should be larger than 15. The input image **I** is randomly rotated with the angle of 0°, 90°, 180° or 270°, to achieve the data augmentation. We train our model from scratch on the synthetic data without finetuning on other datasets, and employ the ADAELTA [61] optimization method. In each iteration, the batch size is set to 128. The model is trained for 400,000 iterations, which costs about 60 hours. The trained model is evaluated on six datasets (IIIT5K, SVT, ICDAR03, ICDAR13, ICDAR15 and ASOT). When testing, the batch size is set to 1.

Our model is implemented with the deep learning framework Tensorflow. All experiments are carried out on a workstation with a 1.70GHz Intel(R) Xeon(R) E5-2609 CPU, a single GeForce GTX 1080 Ti GPU, and 64G RAM.

C. Exploration Study

1) Ablation Studies: For comprehensive evaluations, we exploit two naive baselines to recognize the arbitrary-orientation scene text. **Baseline-A** trains an orientation classifier with ResNet-50 to classify the input image into one of the four directions (0°, 90°, 180° and 270°), and then employs CRNN [21] to recognize the scene text that is rotated to the horizontal direction based on the predicted direction. **Baseline-B** utilizes CRNN [21] to recognize the four directional images (0°, 90°, 180° and 270°) for each input image, and then selects the predicted text with the highest log-softmax score as the final prediction.

TABLE IV

COMPARISON BETWEEN THE DYNAMIC LOG-POLAR TRANSFORMER MODULE AND THE ORIENTATION ATTENTION MODULE. THE REPORTED PERFORMANCE IS THE CASE-INSENSITIVE WORD ACCURACY (%). THE RESULTS ARE EVALUATED ON THE TESTING SET OF IIIT5K WITHOUT USING LEXICON. THE BEAM WIDTH IS SET TO $\beta = 1$

module	0°	90°	180°	270°	avg
DLPT	**75.06**	76.34	74.47	75.91	75.45
OA	73.38	75.37	74.38	72.84	74.00
DLPT+OA	**75.06**	**76.42**	**75.03**	**76.32**	**75.70**

Meanwhile, we also compare our model with the most related method, arbitrary orientation network (**AON**) [19]. Since CRNN [21] is only trained on the Synth90k, we also train our model and AON on the Synth90k for a fair comparison. When evaluating the effectiveness of our proposed model, we conduct ablation studies on the testing set of IIIT5K, and set the beam width $\beta = 1$. When providing a lexicon, the final prediction should have the smallest edit distance with the word in the lexicon.

Under the unconstrained condition (the lexicon size is zero), when the dynamic log-polar transformer (DLPT) is adopted, it has a significant improvement in performance, as shown in Table II. Specifically, the case-insensitive word accuracy (CIWA) on four directions of the input image (0°, 90°, 180° and 270°) has promoted 11.39%, 5.47%, 3.74% and 8.96%, respectively. The average CIWA has increased by 7.63%. The reasons for improvements are that (i) DLPT maps the input image into the representations in the log-polar space, which helps to capture the rotation variations of scene texts; (ii) the deep feature extractor in the feature encoder would convert the height and width of lop-polar features into 1 and L, which keeps more representations in the horizontal direction, thus DLPT has more improvement for the scene text with semantically-horizontal directions (e.g., 0°) than that with vertical directions (90° and 270°). In addition, the wrapping layer in DLPT also plays positive effect on the recognition, as this layer can model the periodicity of rotation angles in the log-polar space. As shown in Table III, when removing the wrapping layer (M = 0), the model only achieves the average CIWA of 72.28%. After adding the wrapping layer (M > 0), the CIWA has obvious improvements. When M = 16, the performance achieves the optimal average CIWA.

When the feature encoder network integrates the CRFA module, the average CIWA has also elevated 1.06%, as shown

TABLE V

INFLUENCES OF PERTURBATION FOR THE LOG-POLAR ORIGIN. THE REPORTED PERFORMANCE IS THE CASE-INSENSITIVE WORD ACCURACY (%). THE RESULTS ARE EVALUATED ON THE TESTING SET OF IIIT5K WITHOUT USING LEXICON. THE BEAM WIDTH IS SET TO $\beta = 1$

perturbation	0°	90°	180°	270°	avg
$\varepsilon_{max} = 0$	72.33	73.97	72.37	74.01	73.17
$\varepsilon_{max} = 0.05$	72.84	73.56	72.64	73.27	73.08
$\varepsilon_{max} = 0.10$	**75.06**	**76.34**	**74.47**	**75.91**	**75.45**
$\varepsilon_{max} = 0.15$	72.37	74.90	72.10	74.71	73.52

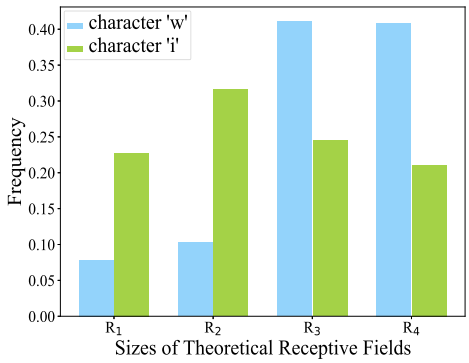


Fig. 8. The relevance of different-scale characters with the sizes of theoretical receptive fields. The frequency of characters is counted by the sampled 100 characters with small scales (e.g., 'i') and 100 characters with large scales (e.g., 'w') based on the receptive field attention weights. These characters are from the correct predictions on the testing set of IIIT5K.

in Table II. Although CRFA is harmful to the scene text with the directions of 90°, 180° and 270°, it can improve the CIWA from 62.58% to 71.08% for the scene text with the direction of 0°. The reasons could be ascribed that (i) the deep feature extractor converts the height and width of the input into 1 and L, respectively, so the generated sequence features could keep more scale information of characters for the horizontally directional scene text; (ii) CRFA would pay more attention to the incorrect representations when the normal characters are rotated by 180°.

When both DLPT and CRFA are considered, the average CIWA of our model achieves 75.45%. It has increased by 8.32% compared with that without DLPT and CRFA, as illustrated in Table II. This combination also promotes the average CIWA of 0.69% and 7.26% compared with the naive network only considering DLPT and CRFA respectively. Besides, when only incorporating CRFA, it does not work for the directions of 90°, 180° and 270°. However, DLPT can facilitate CRFA to not only work well for the direction of 0° (73.97%→75.06%), but also improve the performance for the directions of 90° (75.60%→76.34%), 180° (74.08%→74.47%) and 270° (75.41%→75.91%). It is because DLPT maps the rotation and scale in the Cartesian coordinate space into the shift in the log-polar space, which helps CRFA to learn better representations for individual characters with various scales.

When providing the lexicon, it has also revealed that DLPT and CRFA can work well. Besides, compared with the inference without lexicon, the lexicon can significantly improve the performance with a little cost of lexicon searching, as shown in Table II. Actually, in our inference, the computational

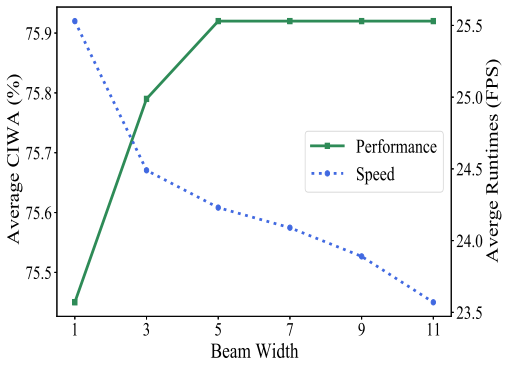


Fig. 9. Effect of the beam width β for the average case-insensitive word accuracy (CIWA) and runtimes. It achieves on the testing set of IIIT5K without using lexicon.

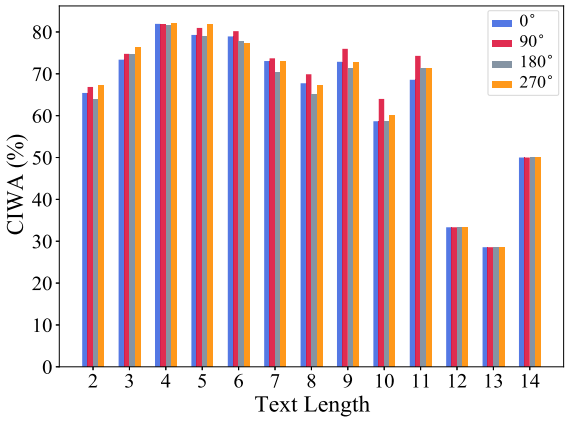


Fig. 10. Effect of the text length. It is evaluated in four semantic orientations (0°, 90°, 180° and 270°) on the testing set of IIIT5K without using lexicon.

complexity of the lexicon searching is $O(n)$ in theory, where n is the lexicon size. With the increase of n, it could distinctly burden the runtime of this post-processing.

Compared with Baseline-A, our method achieves better performance and inference time, as shown in Table II. Specifically, the average CIWA of our model has promoted from 67.46% to 75.45%. Meanwhile, the average runtime of inference per image for our method has a boost from 7.02 FPS to 25.53 FPS. It is because Baseline-A needs to classify the directions of the scene texts before recognizing them. It trains the classifier and the recognizer separately. The accuracy of the classifier would influence the performance of the recognizer. Instead, our model is trained in an end-to-end manner.

Compared with Baseline-B, the average CIWA of our method achieves the improvement of 1.28% and 4.22% without and with a lexicon, respectively. Since Baseline-B needs to test four directions for each input image, it strikingly increases runtimes. As shown in Table II, our method is faster than Baseline-B (25.53 FPS vs. 1.80 FPS) under the unconstrained condition. When providing the lexicon with the size of 1k, our method also outperforms Baseline-B (23.96 FPS vs. 1.78 FPS). Such superiority mainly depends on the inference of the recognition network. It is because the lexicon searching is very fast when the lexicon size is 1k. However, our runtime of the lexicon searching is four times faster than Baseline-B in theory. When the provided lexicon is extremely large, our method would be more superior to Baseline-B.

TABLE VI

INFLUENCES OF THE NUMBER OF SCALES FOR THE COMPUTATIONAL EFFICIENCY AND THE PERFORMANCE. '#PARAMS' DENOTES THE NUMBER OF PARAMETERS ($\times 10^6$). '#FLOPS' MEANS THE NUMBER OF FLOATING-POINT OPERATIONS ($\times 10^9$). 'AVERAGE CIWA' IS THE AVERAGE CASE-INSENSITIVE WORD ACCURACY (%) ON FOUR DIRECTIONS

Scales	DLPT		Scale-Adaptive Feature Encoder		Text Sequence Decoder		Total #Params	Total #FLOPs	Average CIWA
	#Params	#FLOPs	#Params	#FLOPs	#Params	#FLOPs			
k = 0	0.01	0.04	13.17	6.59	1.78	0.02	14.96	6.65	74.76
k = 1			13.37	6.62			15.16	6.68	74.20
k = 2			15.14	6.76			16.93	6.82	74.31
k = 3			16.91	7.02			18.70	7.08	74.87
k = 4			18.68	7.50			20.47	7.56	75.45
k = 5			20.45	8.74			22.24	8.80	74.96
k = 6			22.22	11.19			24.01	11.25	75.05

Fig. 11. Qualitative comparisons between our SLOAN (Ours) and other typical methods (GRCNN [23], ALE [26], ASTER [9], MORN-v2 [10] and SAR [14]). Each row denotes one sample. The image in (a) denotes the original image that directly comes from the testing set, while the images in (b), (c) and (d) stand for the rotation images that rotate the original image with counter-clockwise 90°, 180°, and 270°, respectively. GT means the ground-truth. Blue denotes the correct recognition and Red indicates the incorrect recognition.

Besides, Baseline-B is a kind of enumerable method, when the orientation types become larger, the runtime would become slower. However, our method is more flexible, which maps any rotations and scales into the corresponding shifts in the log-polar space.

Compared with AON [19], our method achieves better CIWA. Even though we only employ DLPT, our model is still superior to AON. It is because DLPT could be regarded as a general orientation attention mechanism. It extends the orientation attention from the discrete space to the continuous space. The representations in the log-polar space can attend to any orientations based on the learned log-polar origin. Meanwhile, DLPT is also a scale-adaptive technique, which adaptively learns the scale variation of the scene text. To further verify the scale-adaptive orientation attention ability of DLPT, we replace the DLPT module in our proposed model with the orientation attention (OA) module in [19]. As shown in Table IV, DLPT is better than OA for any directions. When DLPT and OA are simultaneously incorporated into our proposed framework, the performance only has a slight improvement than DLPT. The reason could be ascribed that DLPT is also a kind of orientation attention. When DLPT and OA are employed together, it is a redundant combination.

2) Effect of the Log-Polar Origin: The log-polar origin (LPO) plays a vital role in the DLPT module, as the transformation is dependent on LPO. In training, the stochastic perturbation of the LPO based on the predicted location (x_0, y_0), can influence the recognition accuracy. The reason is that this perturbation is helpful to learn more rotation and scale variations. We set the perturbation as ε, thus the new location of the LPO becomes ($x_0 \pm \varepsilon$, $y_0 \pm \varepsilon$), where $\varepsilon \in [0, \varepsilon_{max}]$. The tiny perturbation along the vertical direction in the log-polar space is associated with a soft rotation of the input image, which helps to capture wider ranges of semantic orientations of the scene texts. Meanwhile, the tiny perturbation along the horizontal direction in the log-polar space is equivalent to the scaling of the entire input image, which could facilitate the learning of the CRFA module, since the multi-scale receptive fields do not always cover all kinds of characters. As shown in Table V, experimental results evaluated on the testing set of IIIT5K without lexicon, have revealed that it achieves optimal recognition accuracies for any kind of rotations (0°, 90°, 180° and 270°) when ε_{max} is set to 0.1. This decent perturbation could promote the average case-insensitive word accuracy from 73.17% to 75.45%.

TABLE VII

COMPARISONS WITH RELATED METHODS. THE REPORTED PERFORMANCE DENOTES THE AVERAGE CASE-INSENSITIVE WORD ACCURACY (%). 'AVG' MEANS THE AVERAGE PERFORMANCE ON FOUR DIRECTIONS

dataset	orientation	Tesseract-OCR [63]	GRCNN [23]	ALE [26]	ASTER [9]	MORN-v2 [10]	SAR [14]	**Our SLOAN**
IIIT5K	0°	39.40	96.34	97.54	**99.06**	98.67	98.91	93.37
	90°	10.13	0.00	3.55	2.38	1.01	**95.01**	94.27
	180°	1.75	1.33	5.46	5.46	5.49	6.27	**93.45**
	270°	32.81	0.04	3.23	2.26	0.94	**95.01**	94.74
	avg	21.02	24.43	27.45	27.29	26.53	73.80	**93.96**
SVT	0°	33.69	96.45	97.22	97.68	96.75	**98.76**	93.97
	90°	9.27	2.94	0.46	6.80	6.64	**93.97**	93.66
	180°	5.56	8.35	14.68	17.00	15.61	20.87	**91.96**
	270°	32.15	2.47	4.48	5.41	7.26	**94.74**	93.66
	avg	20.17	27.55	29.21	31.72	31.57	77.08	**93.31**
ICDAR03	0°	52.33	98.14	97.91	97.67	98.14	**98.26**	95.17
	90°	10.81	0.35	1.74	1.40	0.70	**95.93**	95.23
	180°	0.81	1.74	2.33	2.09	2.09	3.02	**95.23**
	270°	40.70	0.00	0.35	0.58	0.35	95.70	**96.05**
	avg	26.16	25.06	25.58	25.44	25.32	73.23	**95.42**
ICDAR13	0°	36.52	**98.37**	97.32	98.02	97.90	**98.37**	94.28
	90°	8.17	0.00	1.40	0.47	0.00	**96.15**	94.40
	180°	10.50	1.17	1.28	1.98	1.05	1.87	**93.16**
	270°	33.26	0.00	0.82	0.35	0.12	**96.27**	93.58
	avg	22.11	24.89	25.21	25.21	24.77	73.17	**93.86**
ICDAR15	0°	15.61	83.35	82.80	84.04	84.39	**89.13**	76.53
	90°	4.47	1.20	2.44	2.29	1.14	77.68	**78.82**
	180°	0.78	0.68	0.88	0.52	0.57	5.31	**78.46**
	270°	11.50	2.13	3.85	6.55	3.07	80.18	**80.49**
	avg	8.09	21.84	22.49	23.35	22.29	63.08	**78.58**
ASOT	0°	13.58	24.56	24.60	24.52	25.00	72.96	**76.49**
	90°	12.90	27.34	28.51	28.43	28.85	69.88	**76.42**
	180°	12.11	16.40	17.41	16.87	16.96	65.14	**76.23**
	270°	12.60	19.33	19.89	19.78	19.59	60.89	**75.74**
	avg	12.80	21.91	22.60	22.40	22.60	67.22	**76.22**

TABLE VIII

SCENE TEXT SPOTTING COMPARISONS WITH TYPICAL METHODS ON THE DATASET ICDAR15. 'STRONG', 'WEAK' AND 'GENERIC' INDICATE DIFFERENT LEXICON TYPES. THE REPORTED PERFORMANCE IS THE *F-MEASURE* (%) UNDER THE EVALUATION METRIC 'WORD-SPOTTING'. 'AVG' DENOTES THE AVERAGE VALUE

method	Strong												
	0°	30°	60°	90°	120°	150°	180°	210°	240°	270°	300°	330°	avg
Deep TextSpotter [48]	58.00	28.78	13.34	1.19	1.11	0.70	0.92	1.09	0.99	1.66	20.94	32.54	13.44
EAA [50]	**85.00**	67.02	31.67	16.21	1.30	0.49	1.06	1.16	3.17	6.01	29.20	66.25	25.71
TextBoxes++ [5]	76.45	45.79	6.52	0.60	1.35	1.83	2.01	1.67	1.52	1.54	8.48	45.42	16.10
Mask TextSpotter [6]	82.40	**70.52**	37.04	4.78	2.75	3.73	3.86	4.25	4.69	17.47	48.58	**73.13**	29.43
Our spotter	75.23	62.78	**62.30**	**67.69**	**61.16**	**65.26**	**68.71**	**61.31**	**59.26**	**67.68**	**63.30**	66.74	**65.12**
method	Weak												
	0°	30°	60°	90°	120°	150°	180°	210°	240°	270°	300°	330°	avg
Deep TextSpotter [48]	53.00	30.25	12.64	1.17	0.72	0.28	0.25	0.63	0.96	2.04	21.02	32.84	12.98
EAA [50]	**80.00**	62.55	28.83	15.36	1.17	0.28	0.81	0.92	2.79	5.27	26.61	62.51	23.93
TextBoxes++ [5]	69.04	38.02	4.99	0.52	0.45	0.33	0.26	2.22	1.14	1.46	6.40	39.09	13.66
Mask TextSpotter [6]	78.10	**66.12**	33.24	3.91	1.48	1.24	1.14	2.04	3.64	14.04	43.72	**68.43**	26.43
Our spotter	67.14	56.63	**56.35**	**62.15**	**55.65**	**58.78**	**62.09**	**54.83**	**53.85**	**62.21**	**56.86**	59.57	**58.84**
method	Generic												
	0°	30°	60°	90°	120°	150°	180°	210°	240°	270°	300°	330°	avg
Deep TextSpotter [48]	51.00	21.56	9.36	0.97	0.53	0.33	0.06	0.39	0.51	1.15	14.82	23.45	10.34
EAA [50]	65.00	47.99	21.13	1.12	1.04	0.26	0.50	0.52	1.97	4.23	18.75	49.76	17.69
Mask TextSpotter [6]	**73.60**	**50.36**	24.88	2.87	1.11	0.89	0.46	1.17	2.67	10.61	30.77	**52.31**	20.98
TextBoxes++ [5]	54.37	24.45	2.56	0.43	0.37	0.00	0.00	0.00	0.76	0.94	4.03	25.76	9.47
Our spotter	52.05	44.22	**42.75**	**46.57**	**41.74**	**44.82**	**48.18**	**41.77**	**40.46**	**46.59**	**43.98**	45.84	**44.91**

3) Effect of the Multi-Scale Features: In the character-level receptive field attention (CRFA) mechanism, multi-scale features could capture much spatial information or the detailed cues of characters to model more valid contexts. When the number of scales K is set to 4, with the Eq. (11), the sizes of theoretical receptive fields on the input of the scale-adaptive feature encoder are 86, 102, 118 and 150 for the features $\mathbf{F}_1$, $\mathbf{F}_2$, $\mathbf{F}_3$ and $\mathbf{F}_4$, respectively. Although the sizes of theoretical receptive fields are larger than the scales of most characters, the recognition model still works well. It is because the effective receptive field only occupies a fraction of the theoretical receptive field [62]. With the increase of K, the features with larger sizes of theoretical receptive fields would be employed. They can contribute to the improvement of recognition

Fig. 12. Failure cases. Red indicates predicted results. Green means the ground-truth.

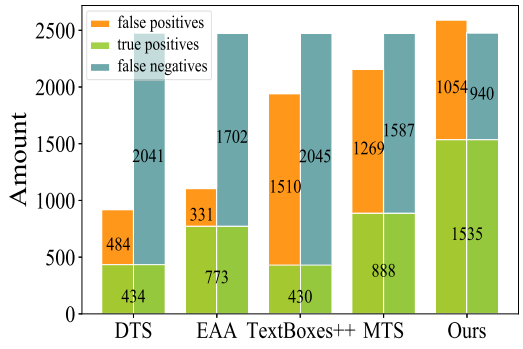


Fig. 13. Comparisons of true positives, false positives and false negatives in Deep TextSpotter (DTS) [48], EAA [50], TextBoxes++ [5], Mask TextSpotter (MTS) [6] and Ours.

accuracy. On the other hand, the number of scales K could also increase the parameters of the model and influence the computational efficiency. As shown in Table VI, when K = 4, the model increases the number of parameters (14.96×10^6 $\rightarrow$ 20.47×10^6) and floating-point operations (6.65×10^9 $\rightarrow$ 7.56×10^9), but it achieves the optimal average CIWA. In addition, the spatial information and detailed cues of characters are helpful to generate more discriminative feature representations in the sequence decoder. Their effective contexts may depend on multiple factors (e.g., scales, positions and semantics of characters). Therefore, the characters with small/large scales are not always associated with the corresponding small/large sizes of theoretical receptive fields. As shown in Fig. 8, we observe that the characters with small scales (e.g., 'i') are associated with the large sizes of receptive fields, while the characters with large scales (e.g., 'w') can also capture discriminative features from the small sizes of receptive fields.

4) Effect of the Beam Width: During inference, when adopting the beam search in the sequence decoding, different beam width β could affect the recognition accuracy and runtimes. To validate the effect of the beam width β, we conduct experiments on the testing set of IIIT5K without using the lexicon. When the beam width β is fixed to 1, it means a greedy decoding strategy that selects the predicted one with the highest score at each decoding step. Otherwise, top-β candidate labels with the highest accumulative scores are maintained at each decoding step. These top-β labels will participate in the calculation of the next decoding step, which can influence the computational efficiency of the entire sequence decoding.

TABLE IX

SCENE TEXT SPOTTING COMPARISONS WITH RELATED METHODS ON THE DATASET ASOT. 'R', 'P' AND 'F' DENOTE THE '*RECALL*', '*PRECISION*' AND '*F-MEASURE*' RESPECTIVELY

Method	Word-Spotting			End-to-End		
	R(%)	P(%)	F(%)	R(%)	P(%)	F(%)
Deep TextSpotter [48]	17.54	47.28	25.58	14.79	46.52	22.45
EAA [50]	31.23	70.02	43.20	26.12	**67.61**	37.69
TextBoxes++ [5]	17.37	22.16	19.48	15.15	21.96	17.93
Mask TextSpotter [6]	35.88	41.17	38.34	34.60	42.56	38.17
MTS-D + SLOAN	46.46	53.31	49.65	41.88	51.52	46.20
CRAFT + MTS-R	39.28	44.86	41.89	37.46	45.56	41.11
Our Spotter	**62.02**	**59.29**	**60.62**	**55.36**	57.52	**56.42**

As shown in Fig. 9, with the increase of the beam width β, the average case-insensitive word accuracy (CIWA) in four directions (0°, 90°, 180° and 270°) achieves some slight improvements. When the beam width attains to a certain value, the average CIWA becomes stable. At the same time, the inference speed gradually slows down. The reported runtimes are calculated by averaging the runtimes of three experimental results on four kinds of directions.

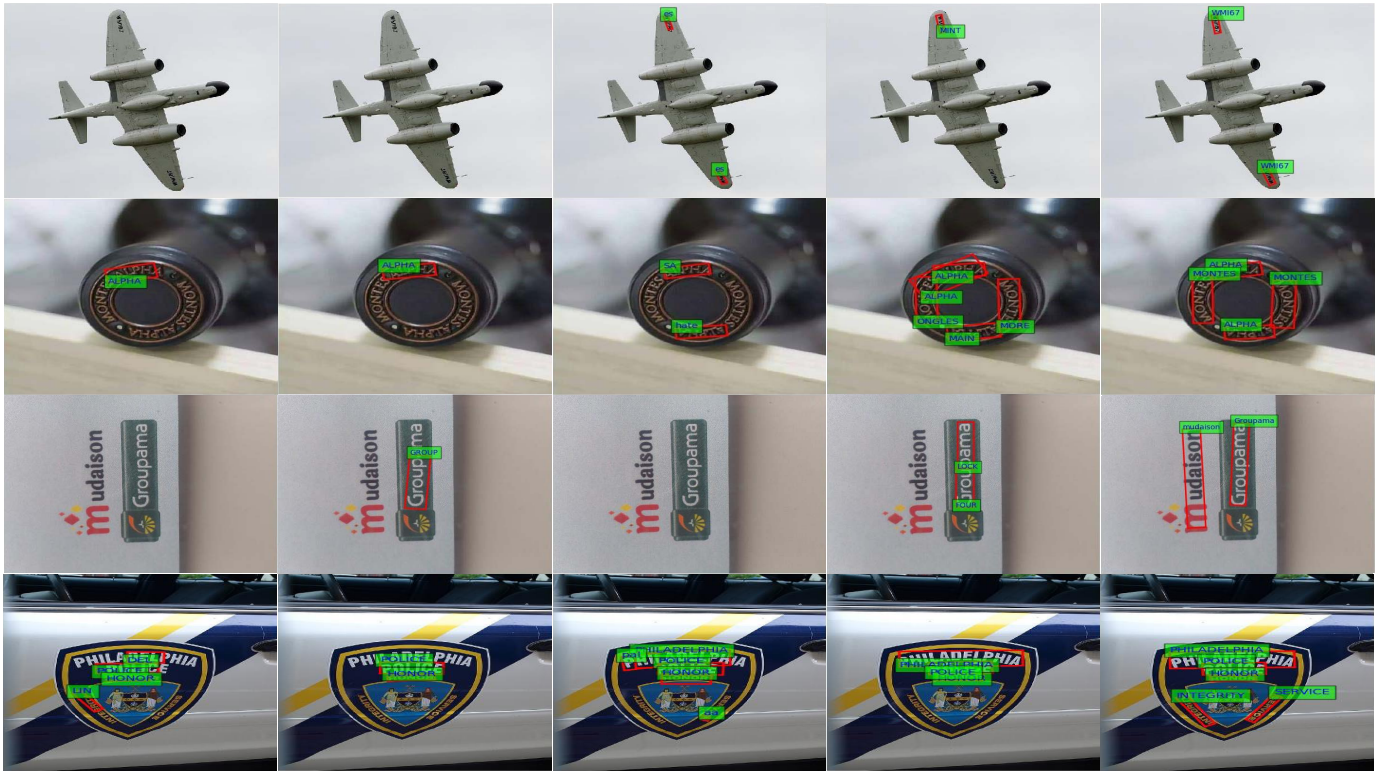
5) Effect of the Text Length: Although we resize the original image to the size of 100×100, the length of the text (namely, the number of characters) could also affect the recognition performance. As shown in Fig. 10, when the length of the text is 4, 5 or 6, the case-insensitive word accuracy (CIWA) of our model is better for all kinds of semantically-oriented scene text (0°, 90°, 180° and 270°). When the length of the text is greater than 11, we observe that the CIWA of our method drops obviously. In effect, the large length of the text is easy to make the misaligned attention in the sequence attention module, which is harder to attain the right prediction under the evaluation protocol CIWA. When the length of the text is small, the sequence attention module would have relatively weak effect on the misalignments, thus the dynamic log-polar transformer (DLPT) module would play the main role. It reveals that the length of the text could influence the prediction of the log-polar origin in the DLPT module.

D. Comparisons With Related Methods

When comparing our recognizer with other typical methods, we train our model with the synthetic datasets Synth90k and SynthText. In the inference, we set the beam width to 5 as [9], [26]. The case-insensitive word accuracy (CIWA) of our method and other related works are reported in Table VII. They are evaluated in four directions (0°, 90°, 180° and 270°) with the largest lexicon. Since ICDAR13 and ICDAR15 do not provide the lexicons for the standalone recognition task, we create the full-word lexicon (containing all word labels in the testing set) for them. For our released challenging dataset ASOT, according to the annotations, we crop the subregion of the scene text via the strategy[2] that uses the function $cv2.minAreaRect$ to generate the minimum circumscribed rectangle and then rotate the rectangle based on the calculated angle.

As shown in Table VII, it is easy to perceive that our method is inferior at the direction of 0° on most datasets,

[2]Refer to https://jdhao.github.io/2019/02/23/crop_rotated_rectangle_opencv



(a) Deep TextSpotter [48] *(b) EAA [50]* *(c) TextBoxes++ [5]* *(d) Mask TextSpotter [6]* *(e) Ours*

Fig. 14. Qualitative comparisons between our scene text spotter and other spotters. Red bounding boxes denote the detection results. Green regions indicate the recognition results.

compared with some typical approaches. It can be ascribed to two main reasons: i) The log-polar origin is learned by a weakly-supervised strategy, which does not always ensure to convert the scene text around 0° into the log-polar space well and thus generate some false predictions. ii) We randomly rotate the input images in the training process, and have downsampled the height of the input image to 1 for generating feature sequences, so it probably has learned some incorrect alignments in the sequence decoder. However, our method is robust to variously directional scene texts. Specifically, when comparing with the excellent open optimal character recognition (OCR) engine, Tesseract-OCR [63], our method achieves striking performance. The reason lies that our model is more suitable for complex scenarios. When comparing with GRCNN [23] and ALE [26], our method also achieves better performance on the non-horizontal direction. It is because GRCNN and ALE focus on recognizing the semantically-horizontal scene text. When comparing with the irregular scene text recognizers ASTER and MORN-v2 [10], our model is still more outstanding. Even though ASTER and MORN-v2 can rectify the irregular text to the horizontal direction for better recognition, they still fail to rectify these extremely-rotated scene texts (e.g., 90°, 180° and 270°). It can be ascribed that the rectification abilities of them are limited. Besides, ASTER and MORN-v2 employ two directional decoders, the left-to-right decoder and the right-to-left decoder, to generate text sequences. Then, the text sequence with the highest score is selected as the final prediction. However, such a decoding strategy is not robust to arbitrary-orientation scene texts. When comparing with SAR [14] that employs the 2D attention to recognize the irregular scene text, the average CIWA of our method also outperforms SAR. In effect, SAR inputs several directional images into the network in the inference and then selects the predicted text with the highest score, but SAR will cost more runtimes. Some qualitative recognition samples are displayed in Fig. 11. It shows that our method can recognize the arbitrary-orientation scene texts better.

E. Limitations

Although our recognition model can work well for most scenarios, it still fails to recognize the scene text in some situations. First, when the character orders of scene texts are top-down, our model cannot recognize these texts, as shown in Fig. 12 (a). It could be ascribed to two reasons: (i) Our training data does not include the scene texts with the top-down character orders. (ii) Our dynamic log-polar transformer is designed to capture the rotation variance of the entire scene text, so it cannot effectively model the rotation of the individual character. Second, our method is unsuccessful in recognizing the scene text with long lengths, as shown in Fig. 12 (b). It is because our proposed method adopts a naive sequence decoding network, which may generate some attention drifts. Third, the extremely-blurred scene texts also fail to be recognized, as shown in Fig. 12 (c). The reason is that the blurred characters are obscured with each other and the background, so it degenerates the discrimination of the learned features.

V. End-to-End Scene Text Recognition System

In practical applications, the scene text recognizer is usually combined with the scene text detector to form an end-to-end scene text recognition system, which also can be called as scene text spotting. In our spotting system, we first employ CRAFT [45] as the detector, which is only finetuned on the dataset ICDAR15. Then we utilize our proposed scene text recognition method to recognize the subregion image that is cropped from the detected bounding box and rotated to the horizontal direction based on the detected angle.

To verify the robustness of the rotation for different scene text spotters, we rotate different angles (from 0° to 360° with the interval of 30°) for the full input image when evaluating on the testing set of ICDAR15. We mainly compare our method with four well-known spotting systems, e.g., Deep TextSpotter [48], EAA [50], TextBoxes++ [5] and Mask TextSpotter [6]. As shown in Table VIII, the results have revealed that our method achieves a better average *F-measure* than other methods. To further verify the generalization ability of our spotter, we evaluate our spotter on the dataset ASOT with the 'weak' lexicon. As shown in Fig. 13, our text spotting system achieves better true positives, false positives and false negatives, compared with other typical spotters. Moreover, the *Recall*, *Precision* and *F-measure* are also reported in Table IX, which further reveals the superiority of our spotter. Some qualitative results are presented in Fig. 14, which have demonstrated that our spotter would not only localize the scene text better but also recognize the detected scene text better.

Although our text spotting system employs an excellently external detector [45], our proposed method SLOAN aims to facilitate the recognition of semantically arbitrary-orientation scene texts. Therefore, whatever the external detector is, it could be helpful when existing large gaps between the detected orientations and the actual semantic orientations. To make fair comparisons, we combine the detector of Mask TextSpotter [6] (termed as MTS-D) with our proposed recognizer (SLOAN) to form a text spotting system (MTS-D + SLOAN). Meanwhile, we also combine the detector CRAFT [45] with the recognizer of Mask TextSpotter (termed as MTS-R) to construct a new text spotting system (CRAFT + MTS-R). As shown in Table IX, it is easy to perceive that our proposed SLOAN helps the text spotting system to achieve better performance.

VI. Conclusion

In this paper, we have presented a novel approach to recognize the arbitrary-orientation scene text. To that end, the dynamic log-polar transformer is first introduced to map the original input image into the log-polar representations that are rotation-aware and scale-aware for the scene text recognition. After that, the character-level receptive field attention is developed to adaptively associate various-scale characters with suitable receptive fields, which could learn more discriminative feature representations. Finally, an attention-based text sequence decoder is employed to generate character sequences. We only require the word images and the corresponding text strings to train the model in an end-to-end fashion. Experiments over several benchmarks validate the effectiveness and superiority of our method. In the future, we will research how to learn the rotation variations of the entire scene texts and individual characters simultaneously. We will also further combine the proposed scale-adaptive and rotation-aware recognizer with the scene text detector to form a one-stage rotation-robust scene text spotter.

References

[1] I. Posner, P. Corke, and P. Newman, "Using text-spotting to query the world," in *Proc. IROS*, Oct. 2010, pp. 3181–3186.

[2] J. I. Olszewska, "Active contour based optical character recognition for automated scene understanding," *Neurocomputing*, vol. 161, pp. 65–71, Aug. 2015.

[3] S. Karaoglu, R. Tao, T. Gevers, and A. W. M. Smeulders, "Words matter: Scene text for image classification and retrieval," *IEEE Trans. Multimedia*, vol. 19, no. 5, pp. 1063–1076, May 2017.

[4] A. Singh *et al.*, "Towards VQA models that can read," in *Proc. CVPR*, Jun. 2019, pp. 8317–8326.

[5] M. Liao, B. Shi, and X. Bai, "TextBoxes++: A single-shot oriented scene text detector," *IEEE Trans. Image Process.*, vol. 27, no. 8, pp. 3676–3690, Aug. 2018.

[6] 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.*, early access, Aug. 26, 2019, doi: 10.1109/TPAMI.2019.2937086.

[7] B. Shi, X. Wang, P. Lyu, C. Yao, and X. Bai, "Robust scene text recognition with automatic rectification," in *Proc. CVPR*, Jun. 2016, pp. 4168–4176.

[8] F. Zhan and S. Lu, "ESIR: End-to-end scene text recognition via iterative image rectification," in *Proc. CVPR*, Jun. 2019, pp. 2059–2068.

[9] 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.

[10] C. Luo, L. Jin, and Z. Sun, "MORAN: A multi-object rectified attention network for scene text recognition," *Pattern Recognit.*, vol. 90, pp. 109–118, Jun. 2019.

[11] M. Yang *et al.*, "Symmetry-constrained rectification network for scene text recognition," in *Proc. ICCV*, Oct. 2019, pp. 9147–9156.

[12] W. Liu, C. Chen, and K. K. Wong, "Char-Net: A character-aware neural network for distorted scene text recognition," in *Proc. AAAI*, 2018, pp. 7154–7161.

[13] X. Yang, D. He, Z. Zhou, D. Kifer, and C. L. Giles, "Learning to read irregular text with attention mechanisms," in *Proc. IJCAI*, Aug. 2017, pp. 3280–3286.

[14] H. Li, P. Wang, C. Shen, and G. Zhang, "Show, attend and read: A simple and strong baseline for irregular text recognition," in *Proc. AAAI*, 2019, pp. 8610–8617.

[15] M. Liao *et al.*, "Scene text recognition from two-dimensional perspective," in *Proc. AAAI*, 2019, pp. 8714–8721.

[16] Z. Xie, Y. Huang, Y. Zhu, L. Jin, Y. Liu, and L. Xie, "Aggregation cross-entropy for sequence recognition," in *Proc. CVPR*, Jun. 2019, pp. 6538–6547.

[17] Z. Wan, M. He, H. Chen, X. Bai, and C. Yao, "TextScanner: Reading characters in order for robust scene text recognition," in *Proc. AAAI*, 2020, pp. 12120–12127.

[18] T. Wang *et al.*, "Decoupled attention network for text recognition," in *Proc. AAAI*, 2020, pp. 12216–12224.

[19] Z. Cheng, Y. Xu, F. Bai, Y. Niu, S. Pu, and S. Zhou, "AON: Towards arbitrarily-oriented text recognition," in *Proc. CVPR*, Jun. 2018, pp. 5571–5579.

[20] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep structured output learning for unconstrained text recognition," in *Proc. ICLR*, 2015, pp. 1–10.

[21] B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 11, pp. 2298–2304, Nov. 2017.

[22] P. He, W. Huang, Y. Qiao, C. C. Loy, and X. Tang, "Reading scene text in deep convolutional sequences," in *Proc. AAAI*, 2016, pp. 3501–3508.

[23] J. Wang and X. Hu, "Gated recurrent convolution neural network for OCR," in *Proc. NeurIPS*, 2017, pp. 334–343.
[24] C.-Y. Lee and S. Osindero, "Recursive recurrent nets with attention modeling for OCR in the wild," in *Proc. CVPR*, Jun. 2016, pp. 2231–2239.
[25] Z. Cheng, F. Bai, Y. Xu, G. Zheng, S. Pu, and S. Zhou, "Focusing attention: Towards accurate text recognition in natural images," in *Proc. ICCV*, Oct. 2017, pp. 5086–5094.
[26] S. Fang, H. Xie, Z.-J. Zha, N. Sun, J. Tan, and Y. Zhang, "Attention and language ensemble for scene text recognition with convolutional sequence modeling," in *Proc. ACM-MM*, Oct. 2018, pp. 248–256.
[27] F. Bai, Z. Cheng, Y. Niu, S. Pu, and S. Zhou, "Edit probability for scene text recognition," in *Proc. CVPR*, Jun. 2018, pp. 1508–1516.
[28] H. Fu, J. Cheng, Y. Xu, D. W. K. Wong, J. Liu, and X. Cao, "Joint optic disc and cup segmentation based on multi-label deep network and polar transformation," *IEEE Trans. Med. Imag.*, vol. 37, no. 7, pp. 1597–1605, Jul. 2018.
[29] R. Matungka, Y. F. Zheng, and R. L. Ewing, "Image registration using adaptive polar transform," *IEEE Trans. Image Process.*, vol. 18, no. 10, pp. 2340–2354, Oct. 2009.
[30] C. Esteves, C. Allen-Blanchette, X. Zhou, and K. Daniilidis, "Polar transformer networks," in *Proc. ICLR*, 2018, pp. 1–14.
[31] M. S. Khorsheed and W. F. Clocksin, "Multi-font Arabic word recognition using spectral features," in *Proc. ICPR*, 2000, pp. 4543–4546.
[32] Q. Ye and D. Doermann, "Text detection and recognition in imagery: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 7, pp. 1480–1500, Jul. 2015.
[33] S. Long, X. He, and C. Yao, "Scene text detection and recognition: The deep learning era," *Int. J. Comput. Vis.*, early access, Aug. 27, 2020, doi: 10.1007/s11263-020-01369-0.
[34] L. Neumann and J. Matas, "A method for text localization and recognition in real-world images," in *Proc. ACCV*, 2010, pp. 770–783.
[35] X. Bai, C. Yao, and W. Liu, "Strokelets: A learned multi-scale mid-level representation for scene text recognition," *IEEE Trans. Image Process.*, vol. 25, no. 6, pp. 2789–2802, Jun. 2016.
[36] X. Lou *et al.*, "Generative shape models: Joint text recognition and segmentation with very little training data," in *Proc. NeurIPS*, 2016, pp. 2793–2801.
[37] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, "Synthetic data and artificial neural networks for natural scene text recognition," in *Proc. NeurIPS Workshop*, 2014, pp. 1–10.
[38] M. Jaderberg, A. Vedaldi, and A. Zisserman, "Deep features for text spotting," in *Proc. ECCV*, 2014, pp. 512–528.
[39] A. Bissacco, M. Cummins, Y. Netzer, and H. Neven, "PhotoOCR: Reading text in uncontrolled conditions," in *Proc. ICCV*, Dec. 2013, pp. 785–792.
[40] J. J. Weinman, Z. Butler, D. Knoll, and J. Feild, "Toward integrated scene text reading," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 2, pp. 375–387, Feb. 2014.
[41] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, "Reading text in the wild with convolutional neural networks," *Int. J. Comput. Vis.*, vol. 116, no. 1, pp. 1–20, Jan. 2016.
[42] F. Borisyuk, A. Gordo, and V. Sivakumar, "Rosetta: Large scale system for text detection and recognition in images," in *Proc. SIGKDD*, Jul. 2018, pp. 71–79.
[43] W. Liu *et al.*, "SSD: Single shot multibox detector," in *Proc. ECCV*, 2016, pp. 21–37.
[44] W. He, X.-Y. Zhang, F. Yin, and C.-L. Liu, "Multi-oriented and multi-lingual scene text detection with direct regression," *IEEE Trans. Image Process.*, vol. 27, no. 11, pp. 5406–5419, Nov. 2018.
[45] Y. Baek, B. Lee, D. Han, S. Yun, and H. Lee, "Character region awareness for text detection," in *Proc. CVPR*, Jun. 2019, pp. 9365–9374.
[46] Y. Liu, L. Jin, and C. Fang, "Arbitrarily shaped scene text detection with a mask tightness text detector," *IEEE Trans. Image Process.*, vol. 29, pp. 2918–2930, 2020.
[47] P. Dai, H. Zhang, and X. Cao, "Deep multi-scale context aware feature aggregation for curved scene text detection," *IEEE Trans. Multimedia*, vol. 22, no. 8, pp. 1969–1984, Aug. 2020.
[48] M. Busta, L. Neumann, and J. Matas, "Deep TextSpotter: An end-to-end trainable scene text localization and recognition framework," in *Proc. ICCV*, Oct. 2017, pp. 2223–2231.
[49] X. Liu, D. Liang, S. Yan, D. Chen, Y. Qiao, and J. Yan, "FOTS: Fast oriented text spotting with a unified network," in *Proc. CVPR*, Jun. 2018, pp. 5676–5685.
[50] T. He, Z. Tian, W. Huang, C. Shen, Y. Qiao, and C. Sun, "An end-to-end TextSpotter with explicit alignment and attention," in *Proc. CVPR*, Jun. 2018, pp. 5020–5029.
[51] Y. Liu, H. Chen, C. Shen, T. He, L. Jin, and L. Wang, "ABCNet: Real-time scene text spotting with adaptive bezier-curve network," in *Proc. CVPR*, Jun. 2020, pp. 9806–9815.
[52] W. Feng, W. He, F. Yin, X.-Y. Zhang, and C.-L. Liu, "TextDragon: An end-to-end framework for arbitrary shaped text spotting," in *Proc. ICCV*, Oct. 2019, pp. 9075–9084.
[53] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. NeurIPS*, 2014, pp. 3104–3112.
[54] A. Gupta, A. Vedaldi, and A. Zisserman, "Synthetic data for text localisation in natural images," in *Proc. CVPR*, Jun. 2016, pp. 2315–2324.
[55] A. Mishra, K. Alahari, and C. Jawahar, "Scene text recognition using higher order language priors," in *Proc. BMVC*, 2012, pp. 1–11.
[56] K. Wang, B. Babenko, and S. Belongie, "End-to-end scene text recognition," in *Proc. ICCV*, Nov. 2011, pp. 1457–1464.
[57] S. M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, and R. Young, "ICDAR 2003 robust reading competitions," in *Proc. ICDAR*, 2003, pp. 682–687.
[58] D. Karatzas *et al.*, "ICDAR 2013 robust reading competition," in *Proc. ICDAR*, Aug. 2013, pp. 1484–1493.
[59] D. Karatzas *et al.*, "ICDAR 2015 competition on robust reading," in *Proc. ICDAR*, Aug. 2015, pp. 1156–1160.
[60] J. Olszewska, "Designing transparent and autonomous intelligent vision systems," in *Proc. ICAART*, 2019, pp. 850–856.
[61] M. D. Zeiler, "ADADLTA: An adaptive learning rate method," Aug. 2012, *arXiv:1212.5701*. [Online]. Available: http://arxiv.org/abs/1212.5701
[62] W. Luo, Y. Li, R. Urtasun, and R. S. Zemel, "Understanding the effective receptive field in deep convolutional neural networks," in *Proc. NeurIPS*, 2016, pp. 4898–4906.
[63] *Tesseract-OCR v4.0*. Accessed: Jan. 20, 2020. [Online]. Available: https://github.com/tesseract-ocr/tesseract/releases

Pengwen Dai received the B.E. degree from the College of Computer Science, Chongqing University, China, in 2014. He is currently pursuing the Ph.D. degree with the Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China. His current research interests include scene text detection and recognition.

Hua Zhang received the Ph.D. degree in computer science from the School of Computer Science and Technology, Tianjin University, Tianjin, China, in 2015. He is currently an Associate Professor with the Institute of Information Engineering, Chinese Academy of Sciences. His research interests include computer vision, multimedia, and machine learning.

Xiaochun Cao (Senior Member, IEEE) received the B.E. and M.E. degrees in computer science from Beihang University, China, and the Ph.D. degree in computer science from the University of Central Florida, Orlando, USA. After graduation, he spent over three years at ObjectVideo Inc., as a Research Scientist. From 2008 to 2012, he was a Professor with Tianjin University, Tianjin, China. He has been a Professor with the Institute of Information Engineering, Chinese Academy of Sciences, since 2012. He is also with the Peng Cheng Laboratory, Cyberspace Security Research Center, China, and the School of Cyber Security, University of Chinese Academy of Sciences, China. In 2004 and 2010, he was a recipient of the Piero Zamperoni Best Student Paper Award at the International Conference on Pattern Recognition. He is on the Editorial Boards of the IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON MULTIMEDIA, and IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY.