Towards End-to-End Text Spotting in Natural Scenes

Peng Wang, Hui Li, and Chunhua Shen

Abstract—Text spotting in natural scene images is of great importance for many image understanding tasks. It includes two sub-tasks: text detection and recognition. In this work, we propose a unified network that simultaneously localizes and recognizes text with a single forward pass, avoiding intermediate processes such as image cropping and feature re-calculation, word separation, and character grouping. The overall framework is trained end-to-end and is able to spot text of arbitrary shapes. The convolutional features are calculated only once and shared by both the detection and recognition modules. Through multi-task training, the learned features become more discriminative and improve the overall performance. By employing a 2D attention model in word recognition, the issue of text irregularity is robustly addressed. The attention model provides the spatial location for each character, which not only helps local feature extraction in word recognition, but also indicates an orientation angle to refine text localization. Experiments demonstrate that our proposed method can achieve state-of-the-art performance on several commonly used text spotting benchmarks, including both regular and irregular datasets. Extensive ablation experiments are performed to verify the effectiveness of each module design.

Index Terms—End-to-end scene text spotting, deep neural network, attention model

1 Introduction

Text—as a fundamental tool of communicating information—scatters throughout natural scenes, e.g., street signs, product labels, license plates, *etc*. Automatically reading text in natural scene images is an important task in machine learning and gains increasing attention due to a variety of applications. For example, accessing text in images can help the visually impaired understand the surrounding environment. To enable autonomous driving, one must accurately detect and recognize every road sign. Indexing text in images would enable image search and retrieval from billions of consumer photos in internet.

End-to-end text spotting includes two sub-tasks: text detection and word recognition. Text detection aims to localize each text in images, using bounding boxes for example. Word recognition attempts to output readable transcription. Compared to traditional optical character recognition (OCR), text spotting in natural scene images is much more challenging, mainly due to the extreme diversity of text patterns and highly complicated background. Text appearing in natural scene images can be of varying fonts, sizes, shapes, orientation and layouts. Moreover, the background can be cluttered, making the task largely unsolved to date.

An intuitive approach to scene text spotting is to divide it into two separated sub-tasks. Text detection is first performed to obtain candidate text bounding boxes, and word recognition is applied subsequently on the cropped regions to output transcriptions. A few approaches were developed which solely focus on text detection [1], [2], [3], [4], [5] or word recognition [6], [7], [8], [9]. Methods are improved from only recognizing simple horizontal text to addressing complicated irregular (oriented or curved) text. We believe that these two sub-tasks are highly correlated and complementary to each other, and thus should be solved in a single framework. On one hand, image features can be shared between these two tasks so as to reducing computational cost. On the other hand, the multi-task training is likely to improve feature representation power and benefit both sub-tasks.

To this end, end-to-end approaches are proposed recently to concurrently tackle both sub-tasks [10], [11], [12], [13]. Note that most end-to-end approaches spend major effort on designing sophisticated detection modules, so as to acquire tighter bounding boxes around the text, alleviating the level of difficulty for word recognition. However, we argue that the ultimate goal of text spotting is to recognize each text in the image, rather than attaining precise bounding box locations. Thus, here we strive for a balance between detection and recognition by letting the recognition module deal with the challenge caused by text irregularity. To be more specific, our detection module is designed to output a rectangular bounding box for each word, regardless of what the text appears (horizontal, oriented or curved). A robust recognition module, which shares image features with the detection module, is devised to effectively recognize the text within the relatively loose bounding box. The overall framework of our method is illustrated in Fig. 1. We use the ResNet [14] as the backbone, with Feature Pyramid Networks (FPN) [15] used to tackle the

- *Peng Wang is with the School of Computer Science and Ningbo Institute, Northwestern Polytechnical University, Xi'an, Shaanxi 710072, China, and also with the National Engineering Laboratory for Integrated Aero-Space-Ground-Ocean Big Data Application Technology, Xi'an, Shaanxi 710072, China. E-mail: peng.wang@nwpu.edu.cn.*
- *Hui Li is with the School of Computer Science, The University of Adelaide, Adelaide, SA 5005, Australia. E-mail: huili03855@gmail.com.*
- *Chunhua Shen is with Monash University, Clayton, VIC 3800, Australia. E-mail: chunhua@me.com.*

Manuscript received 5 Jan. 2020; revised 3 May 2021; accepted 25 June 2021. Date of publication 9 July 2021; date of current version 9 Sept. 2022. (Corresponding author: Chunhua Shen.) Recommended for acceptance by N. Quadrianto. Digital Object Identifier no. 10.1109/TPAMI.2021.3095916

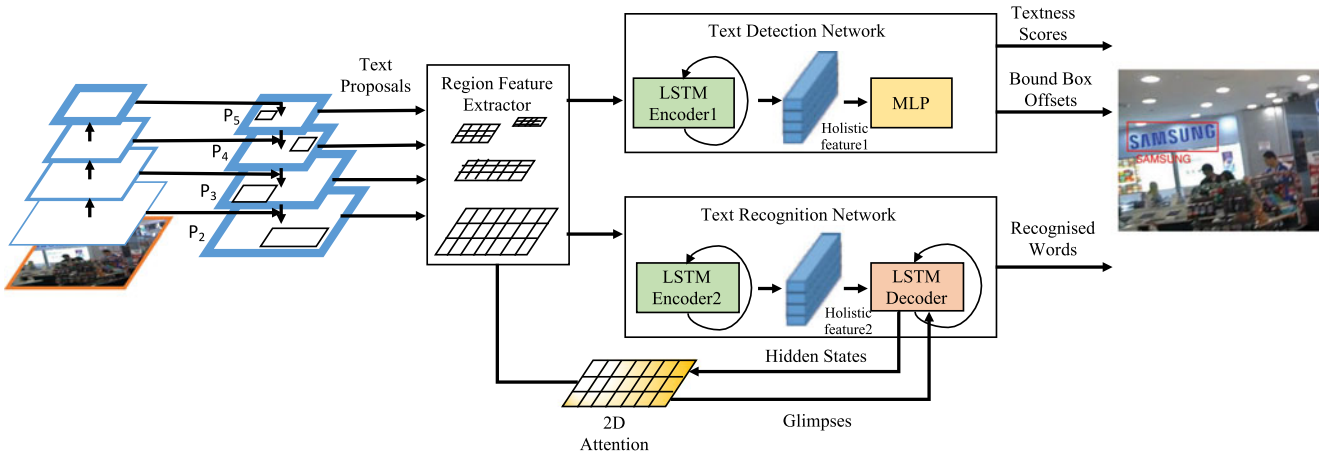


Fig. 1. *The overall architecture of our proposed model for end-to-end text spotting in natural scene images.* The network takes an image as input, and outputs both text bounding boxes and text labels in one single forward pass. The entire network is trained end-to-end.

multi-scale detection. Text Proposal network (TPN) is adapted at multiple levels of the feature pyramid so as to obtain text proposals of different scales. A RoI pooling layer is then employed to extract varying-size 2D features from each proposal, which are then used both in the text detection network and word recognition network. A 2-dimensional attention network is employed in the word recognition module. The 2D attention model attends on the local discriminative features for each individual character during decoding, which improves the recognition accuracy. At the same time, the attention model indicates the character alignment in each word bounding box, with which we can refine the loosely localized bounding box. The recognition module can also be used to reject false positives in the detection phase, thus improving the overall performance.

This work here is an extension of our previous work published in [10] and [16]. The work in Li *et al.* [10] proposed *the first end-to-end trainable framework for scene text spotting*. However, a significant drawback of [10] is that it is incapable of dealing with irregular text that is oriented or curved. In [16] we presented a 2D attention based simple baseline for irregular text recognition, which is robust to various distortions. Here, we still use the 2D attention based encoder-decoder framework in the recognition branch. Moreover, the calculated attention weights are exploited to compute the orientation angle for bounding box refinement in the end-to-end scenario. The improvements compared to [10] are as follows.

1) The work here is able to tackle text with arbitrary shapes. It is no longer restricted to horizontal text as in [10].
2) We now use ResNet with FPN as the backbone network, leading to significantly improved feature representations. We also adapt the text proposal network with pyramid feature maps. These two modifications are able to propose text instances at a wide range of scales and improve the recall for small size text.
3) The training process is made much simpler. Instead of training the detection and recognition modules separately at the early stages as in [10], the new framework is trained completely in a simple end-to-end fashion. Both detection and recognition tasks are jointly optimized in the training process. Our code is optimized, resulting in a faster computational speed compared to [10].
4) More experiments are conducted on three additional datasets to demonstrate the effectiveness of the proposed method in dealing with various text appearance.

The main contributions of this work are thus four-fold.

1) We design an end-to-end trainable network, which can localize text in natural scene images and recognize it simultaneously. The method is robust to text appearance variation in that it can detect and recognize arbitrarily-oriented text. The convolutional features are shared by both the detection and recognition modules, which reduces computational cost, comparing with approaches that need two distinct models. In addition, the multi-task optimization benefits the feature learning, and thus promotes the detection results and the overall performance. To our knowledge, ours is the first work that integrates text detection and recognition into a single end-to-end trainable network.
2) A tailored RoI pooling method is proposed, which takes the significant diversity of aspect ratios in text bounding boxes into account. The generated RoI feature maps accommodate the aspect ratios of different words and extract sufficient information that is essential for the subsequent detection and recognition.
3) We take full use of the 2D attention mechanism in both word recognition and bounding box refinement. The learned attention weights can not only select local features to boost recognition performance, but also provide character locations which can be used to refine the bounding boxes. Note that the 2D attention model is trained in a *weakly supervised* manner using the cross-entropy loss in word recognition. We do not require additional pixel-level or character-level annotations for supervision.
4) Our work provides a new approach of solving the end-to-end text spotting problem. Conventional methods have been built on the idea of obtaining accurate and tight bounding boxes around the text at the first step, so as to exclude redundant noises and make the word recognition task easier. In contrast, a strong and robust word recognition model underpins our framework,

which can compensate the detection module, leading to a simple end-to-end text spotting method. Experiments on several widely-used text spotting benchmarking datasets, including ICDAR2013, ICDAR2015, Total-Text and COCO-Text, achieve state-of-the-art results, which demonstrate the effectiveness of our method.

2 Related Work

In this section, we review some related work on text detection, word recognition and end-to-end text spotting. Comprehensive surveys on scene text detection and recognition can be found in [17], [18], [19], [20].

Text Detection With the development of deep learning, text detection in natural scene images has achieved significant progress. Methods have been developed for detecting regular horizontal text, oriented and curved text.

Pioneering methods such as [21], [22] simply use pre-trained Convolutional Neural Networks (CNNs) as classifiers to distinguish characters from background. Heuristic steps are needed to group characters into words. Zhang *et al.* [23] proposed to extract text lines by exploiting text symmetry property. Tian *et al.* [24] developed a vertical anchor mechanism, and proposed a Connectionist Text Proposal Network (CTPN) to accurately localize text lines in images. Advances in generic object detection and segmentation provide inspirations for text detection. For example, inspired by Faster-RCNN [25], Zhong *et al.* [26] designed a text detector with a multi-scale Region Proposal Network (RPN) and a multi-level RoI pooling layer which can localize word level bounding boxes directly. Gupta *et al.* [27] used a Fully-Convolutional Regression Network (FCRN) for efficient text detection and bounding box regression, motivated by YOLO [28]. Similar to SSD [29], Liao *et al.* [30] proposed "TextBoxes" by combining predictions from multiple feature maps with different resolutions. Those methods output horizontal rectangles for detecting text of regular shapes.

The authors of [31] proposed to localize text lines via salient maps. Post-processing techniques were proposed to extract text lines in multiple orientations. Ma *et al.* [32] introduced Rotation Region Proposal Networks (RRPN) to generate orientated proposals. He *et al.* [33] proposed to use an attention mechanism to identify text regions from images. The bounding box position was regressed with an angle for box orientation. These methods output rotated rectangular bounding boxes. In addition, Zhou *et al.* [2] proposed "EAST" that uses FCN to produce word level predictions which can be either rotated rectangles or quadrangles. Liu *et al.* [3] proposed Deep Matching Prior Network (DMPNet) to detect text with tighter quadrangle. Liao *et al.* [4] improved "TextBoxes" to predict orientation angles or quadrilateral bounding box offsets so as to detect oriented scene text (referred to as "TextBoxes++"). Lyu *et al.* [34] proposed to detect scene text by localizing the corner points of text bounding boxes and segmenting text regions in relative positions. Candidate boxes are generated by sampling and grouping corner points, which results in quadrangle detection.

Recently, more advanced methods are proposed to predict bounding boxes of polygons, which aim to fit text more tightly. For example, inspired by Mask R-CNN [35], Xie *et al.* [36] proposed to detect arbitrary shape text based on FPN [15] and instance segmentation. Zhang *et al.* [5] proposed to detect text via iterative refinement and shape expression. An instance-level shape expression module was introduced to generate polygons that can fit arbitrary-shape text (e.g., curved). Progressive Scale Expansion Network (PSENet) [37] performs pixel-level segmentation for localizing text instances precisely of arbitrary shapes. Tian *et al.* [38] solved text detection using instance segmentation. Pixels belonging to the same word are pulled together as connected components while pixels from different words are pushed away from each other.

The text detection module of our framework is based on the Faster R-CNN framework [25], which aims to generate word-level bounding boxes directly, eliminating intermediate steps such as character aggregation and text line separation. In order to cover text of a variety of scales and aspect ratios, FPN [15] is used to generate text proposals with both higher recall and precision. Different from other work, we use the minimal horizontal rectangle that encloses the whole word as the ground-truth. It contains sufficient information for text spotting. Besides, the overall framework can be simplified as we do not need additional modules to accommodate text orientation. In our framework, a preciser localization is obtained later by using the word recognition results.

Word Recognition. Word recognition means to recognize the cropped word image patches and output character sequences. Early work on scene text recognition often works in a bottom-up fashion [21], [39], which detects individual characters first and integrates them into a word by dynamic programming. Top-down methods [40], recognize a word patch as a whole, and formulate it as a multi-class classification problem. Considering that scene text generally appears in the form of a character sequence, recent work models it as a sequence recognition problem. Recurrent Neural Networks (RNNs) are employed for this purpose.

The work in [41] and [6] formulates word recognition as one-dimensional sequence labeling problem using RNNs. A Connectionist Temporal Classification (CTC) layer [42] is used to decode the sequence, eliminating the need of segmenting characters.

The work in [43] and [44] recognizes text using an attention-based sequence-to-sequence framework [45], in which RNNs are able to learn the character-level language model. A 1D soft-attention model was employed to select relevant local features. The RNN+CTC and sequence-to-sequence frameworks serve as two meta-algorithms that are widely used by recent approaches. Both models can be trained end-to-end and achieve considerable improvements on regular text recognition. Cheng *et al.* [46] observed that the frame-wise maximal likelihood loss, which is conventionally used to train the encoder-decoder framework, may be confused and misled by missing or superfluity of characters, and degrades the recognition accuracy. They proposed "Edit Probability" to tackle this misalignment problem.

Recognizing irregular text has also attracted much attention recently. Shi *et al.* [8], [44] rectified oriented and curved text using Spatial Transformer Network (STN) [47]. ESIR [9] employed a line-fitting transformation to estimate the pose of text, and developed a pipeline that iteratively removes perspective distortion and text line curvature in order to achieve improved recognition accuracy.

Instead of rectifying the whole distorted text image, Liu *et al.* [48] presented a Character-Aware Neural Network (Char-Net) to detect and rectify individual characters, which, however, requires expensive character-level annotations. Cheng *et al.* [49] proposed a Focusing Attention Network (FAN) that is composed of an attention network for character recognition and a focusing network to adjust the attention drift between local character features and targets. Character-level bounding box annotations are also required. Cheng *et al.* [7] applied LSTMs in four directions to encode arbitrarily-oriented text. The work in [16] depends on a tailored 2D attention mechanism to deal with the complicated spatial layout of irregular text, and shows significant flexibility and robustness. In this work, we use a 2D attention model in the recognition module, and train altogether with the detection module for end-to-end text spotting.

End-to-End Text Spotting. Most previous methods design a multi-stage pipeline to achieve text spotting. For instance, Jaderberg *et al.* [50] generated a large number of text proposals, and then trained the word classifier for recognition. Gupta *et al.* [27] employed FCRN for text detection and the word classifier in [40] for recognition. Liao *et al.* [4] combined "TextBoxes++" and "CRNN" [6] to complete the text spotting task. The work in ASTER [8] combines "TextBoxes" [30] and a rectification based recognition method for text spotting.

Preliminary results of this work, presented in [10], is among the first a few, to explore a unified end-to-end trainable framework for simultaneous text detection and recognition. Although in one single framework, the work in [51] does not share features between detection and recognition parts, which can be seen as a loose combination. He *et al.* [11] proposed an end-to-end text spotter which can compute convolutional features for oriented text instances. A 1D character attention mechanism was introduced via explicit alignment, which improves performance. Note that character level annotations are needed for supervision. Liu *et al.* [12] presented "FOTS" that proposes "RoIRotate" to share convolutional features between detection and recognition for oriented text. 1D sequential features are extracted via several layers of CNNs and RNNs, and decoded by a CTC layer. Both work may encounter difficulty in dealing with curved or distorted scene text, where the orientation is not well defined.

Lyu *et al.* [13] proposed "Mask TextSpotter" that introduced a mask branch for text instance segmentation, inspired by Mask R-CNN [35]. It can detect and recognize text of various shapes, including horizontal, oriented and curved text. Again, character-level mask information is requested for training. Its extended version [52] integrated a Spatial Attention Module in the recognition module, which mitigates the above-mentioned problem. Sun *et al.* [53] proposed "TextNet" to read irregular text. It outputs quadrangle text proposals. A perspective RoI transform is developed to extract features from an arbitrary-size quadrangle for recognition. Four directional RNNs are used to encode the irregular text instances, which results in context features for the following spatial attention mechanism in the decoding process. More recently, Qin *et al.* [54] formulated arbitrary shape text spotting as an instance segmentation problem. Xing *et al.* [55] proposed convolutional character network, which performs detection and recognition at the character level. Liu *et al.* [56] proposed ABCNet that fits cubic Bezier curves to curved text and designs a BezierAlign layer to extract curved sequence features.

In contrast to designing a sophisticated framework to accommodate the variety of text shapes, which can potentially increase model complexity, we resort to the conventional horizontal bounding box to represent text location. Text irregularity is post-processed by the outputs from 2D attention model in word recognition. The work of ASTER [8] used the control points obtained in recognition model to rectify the detection results, which is similar to our post-processing step to some extent. The main difference is that our model is end-to-end trainable, but ASTER is not.

3 Our Method

The overall architecture of our proposed model is illustrated in Fig. 1. Our goal is to design an end-to-end trainable network, which can simultaneously detect and recognize all words in natural scene images, robust to various appearances. The overall framework consists of five components: 1) a ResNet backbone with FPN embedded for feature extraction; 2) a TPN with a shared head across all feature pyramid levels for text proposal generation; 3) a Region Feature Extractor (RFE) to extract varying length 2D features that accommodate text aspect ratios and are shared by following detection and recognition modules; 4) a Text Detection Network (TDN) for proposal classification and bounding box regression; and 5) meanwhile a Text Recognition Network (TRN) with 2D attention equipped for proposal recognition.

We attempt to design a simple model. Hence, we exclude additional modules for dealing with the irregularity of text. Instead, we solely rely on a 2D attention mechanism in both word recognition and location refinement. Despite its simplicity, we shown that our mode is robust in various scenarios. In the following, we describe each component of the model in detail.

3.1 Backbone

A pre-trained ResNet [14] is used here as the backbone convolutional layers for its good performance on image recognition. It consists of 5 residual blocks with down sampling ratios of $\{2, 4, 8, 16, 32\}$ separately for the last layer of each block, with respect to the input image. We remove the final pooling and fully connected layer. Thus an input image leads to a pyramid of feature maps. In order to build high-level semantic features, FPN [15] is applied which uses a bottom-up and a top-down pathways with lateral connections to learn a strong semantic feature pyramid at all scales. It shows a significant improvement on bounding box proposals [15]. Similarly, we exclude the output from conv1 in the feature pyramid, and denote the final set of feature pyramid maps as $\{P_2, P_3, P_4, P_5\}$. The feature dimension is also fixed to $d = 256$ in all feature maps.

3.2 Text Proposal Network

In order to take full use of the rich semantic feature pyramid as well as the location information, following the work in [15], we attach a head with 3×3 convolution and two sibling 1×1 convolutions (for text/non-text classification and bounding box regression respectively) to each level of the feature pyramid, which gives rise to anchors at different

levels. Considering the relatively small sizes of text instances, we define the anchors of sizes $\{16^2, 32^2, 64^2, 128^2, 256^2\}$ pixels on $\{P_2, P_3, P_4, P_5, P_6\}$ respectively, where P_6 is a stride two subsampling of P_5. The aspect ratios are set to $\{0.125, 0.25, 0.5, 1.0\}$ by considering that text bounding boxes usually have larger width than height. Therefore, there are in total 20 anchors over the feature pyramid, which are capable of covering text instances with different scales and shapes.

The heads with 3×3 conv and two 1×1 conv's share parameters across all feature pyramid levels. They extract features with 256-d from each anchor and fed them into two sibling layers for text/non-text classification and bounding box regression. The training of TPN follows the work in FPN [15].

3.3 Region Feature Extractor

Given that text instances usually have a large variation on word length, it is unreasonable to make fixed-size RoI pooling for short words like "Dr" and long words like "congratulations". This would inevitably lead to significant distortion in the produced feature maps, which is disadvantageous for the downstream text detection and recognition networks. In this work, we propose to re-sample regions according to their perspective aspect ratios. RoI-Align [35] is also used to improve alignment between input and output features. For RoIs of different scales, we assign them to different pyramid levels for feature extraction, following the method in [15]. The difference is that, for an RoI of size $h \times w$, a spatial RoI-Align is performed with the resulting feature size of

$$H \times \max(H, \min(W_{max}, 3Hw/h)), \tag{1}$$

where the expected height H is fixed to 4, and the width is adjusted to accommodate the large variation of text aspect ratios. The resulted feature maps are denser along the width direction compared to the height direction, which reserves more information along the horizontal axis and benefits the subsequent recognition task. Moreover, the feature width is clamped by H and a maximum length W_{max} which is set to 30 in our work. The resulting 2D feature maps (denoted as $\mathbf{V}$ of size $H \times W \times D$ where $D = 256$ is the number of channels) are used: 1) to extract holistic features for the following text detection and recognition; 2) as the context for the 2D attention network in text recognition.

3.4 Text Detection Network

Text Detection Network (TDN) aims to classify whether the proposed RoIs are text or not and refine the coordinates of bounding boxes again, based on the extracted region features $\mathbf{V}$. Note that $\mathbf{V}$ is of varying sizes. To extract a fixed-size holistic feature from each proposal, RNNs with Long-Short Term Memory (LSTM) is adopted. We flatten the features in each column of $\mathbf{V}$, and obtain a sequence $\{\mathbf{q}_1, \ldots, \mathbf{q}_W\}$ where $\mathbf{q}_t \in \mathbb{R}^{D \times H}$. The sequential elements are fed into LSTMs one by one. Each time LSTMs receive one column of feature $\mathbf{q}_t$, and update their hidden state $\mathbf{h}_{\mathbf{d}t}$ by a non-linear function: $\mathbf{h}_{\mathbf{d}t} = \mathrm{f}(\mathbf{q}_t, \mathbf{h}_{\mathbf{d}t-1})$. In this recurrent fashion, the final hidden state $\mathbf{h}_{\mathbf{d}W}$ (with size $R = 1024$) captures the holistic information of $\mathbf{V}$ and is used as a RoI representation with fixed dimension. Two fully-connected layers with 1024 neurons are applied on $\mathbf{h}_{\mathbf{d}W}$, followed by two parallel layers for classification and bounding box regression respectively.

To boost the detection performance, an online hard negative mining is used during training. We first apply TDN on 1024 initially proposed RoIs. The ones that have higher textness scores but are actually negatives are re-sampled to harness TDN. In the re-sampled RoIs, we restrict the positive-to-negative ratio as $1:3$, where in the negative RoIs, we use 70 hard negatives and 30 percent random sampled ones. Through this processing, we observe that the text detection performance can be improved significantly.

3.5 Text Recognition Network

Text Recognition Network (TRN) aims to predict the text in the detected bounding boxes based on the extracted region features. Considering the irregularity of text, we apply a 2D attention mechanism based the encoder-decoder network for text recognition, following the work in [16]. The extracted RoI feature $\mathbf{V}$ is adopted directly in the recognition network, instead of cropping the text proposals out and feeding to another standalone backbone CNNs for feature extraction. Without additional transformation on the extracted RoI features, the proposed attention module is able to accommodate text of arbitrary shape, layout and orientation.

The extracted RoI feature $\mathbf{V}$ is encoded again to extract discriminative features for word recognition. 2 layers of LSTMs are employed here in the encoder, with 512 hidden states per layer. The LSTM encoder receives one column of the 2D features maps at each time step, followed by max-pooling along the vertical axis, and updates its hidden state $\mathbf{h}_t$. After W steps, the final hidden state of the second RNN layer, $\mathbf{h}_W$, is regarded as the holistic feature for word recognition.

The decoder is another 2-layer LSTMs with 512 hidden states per layer. Here the encoder and decoder do not share parameters. As illustrated in Fig. 2, initially, the holistic feature $\mathbf{h}_W$ is fed into the decoder LSTMs at time step 0. Then a "$START$" token is input into LSTMs at step 1. From time step 2, the output of the previous step is fed into LSTMs until the "END" token is received. All the inputs to LSTMs are represented by one-hot vectors, followed by a linear transformation $\Psi(\cdot)$.

During training, the inputs of decoder LSTMs are replaced by the ground-truth character sequence. The outputs are computed by the following transformation:

$$\mathbf{y}_t = \varphi(\mathbf{h}'_t, \mathbf{g}_t) = \mathrm{softmax}(\mathbf{W}_o[\mathbf{h}'_t; \mathbf{g}_t]), \tag{2}$$

where $\mathbf{h}'_t$ is the current hidden state and $\mathbf{g}_t$ is the output of the attention module. $\mathbf{W}_o$ is a linear transformation, which embeds features into the output space of 38 classes, in corresponding to 10 digits, 26 case insensitive letters, one special token representing all punctuation, and an "END" token.

The attention model $\mathbf{g}_t = \mathrm{Atten}(\mathbf{V}, \mathbf{h}'_t)$ is defined as:

$$\begin{cases} \mathbf{e}_{ij} = \tanh(\mathbf{W}_v \mathbf{v}_{ij} + \mathbf{W}_h \mathbf{h}'_t), \\ \alpha_{ij} = \mathrm{softmax}(\mathbf{w}_e^T \cdot \mathbf{e}_{ij}), \\ \mathbf{g}_t = \sum_{i,j} \alpha_{ij} \mathbf{v}_{ij}, \quad i = 1, \ldots, H, \quad j = 1, \ldots, W. \end{cases}' \tag{3}$$

where $\mathbf{v}_{ij}$ is the local feature vector at position (i, j) in the extracted region feature $\mathbf{V}$; $\mathbf{h}'_t$ is the hidden state of decoder

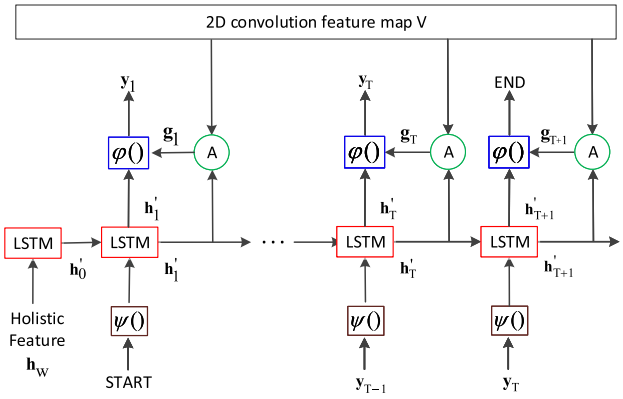


Fig. 2. The structure of the LSTM decoder used in this work. The holistic feature $\mathbf{h}_W$, a "START" token and the previous outputs are input into LSTM subsequently, terminated by an "END" token. At each time step t, the output y_t is computed by $\varphi(\cdot)$ with the current hidden state and the attention output as inputs.

LSTMs at time step t, to be used as the guidance signal; $\mathbf{W}_v$ and $\mathbf{W}_h$ are linear transformations to be learned; α_{ij} is the attention weight at location (i, j); and $\mathbf{g}_t$ is the weighted sum of local features, denoted as a *glimpse*.

The attention module is learned in a *weakly supervised* manner by the cross entropy loss in the final word recognition. *No pixel-level or character-level annotations are required for supervision in our model*. The calculated attention weights can not only extract discriminative local features for the character being decoded and help word recognition, but also provide a group of character location information. For irregular text, an orientation angle is then calculated based on the character locations in the proposal, which can be used to refine the bounding boxes afterwards. To be more specific, as shown in Fig. 3, a linear equation can be regressed based on the character locations specified by the attention weights in decoding process. The output rectangle is then rotated based on the computed slope. In practice, we remove attention weights smaller than 0.2 to reduce noise.

3.6 Loss Functions and Training

Our proposed framework is trained in an end-to-end manner, requiring only input images, the ground-truth word bounding boxes and their text labels as input during training phase. Instead of requiring quadrangle or more sophisticated polygonal coordinate annotations, in this work we are able to use the simplest horizontal bounding box which indicates the minimum rectangle encircling the word instance. In addition, no pixel-level or character-level annotations are requested for supervision. Specifically, both TPN and TDN employ the binary logistic loss L_{cls} for classification, and smooth L_1 loss L_{reg} [25] for regression. So the loss for training TPN is

$$L_{TPN} = \frac{1}{N}\sum_{i=1}^{N} L_{cls}(p_i, p_i^\star) + \frac{1}{N_+}\sum_{i=1}^{N_+} L_{reg}(\mathbf{d}_i, \mathbf{d}_i^\star), \tag{4}$$

where N is the number of randomly sampled anchors in a mini-batch and N_+ is the number of positive anchors in this batch. The mini-batch sampling and training process of TPN are similar to that used in [15].

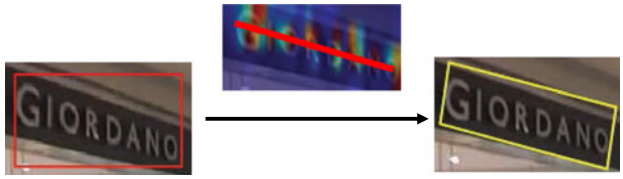


Fig. 3. Box refinement according to character alignment indexed by attention weights.

An anchor is considered as positive if its Intersection-over-Union (IoU) ratio with a ground-truth is greater than 0.7 and considered as negative if its IoU with any ground-truth is smaller than 0.3. N is set to 256 and N_+ is at most 128. p_i denotes the predicted probability of anchor i being text and $p_i^\star$ is the corresponding ground-truth label (1 for text, 0 for non-text). $\mathbf{d}_i$ is the predicted coordinate offsets $(\mathrm{dx}_i, \mathrm{dy}_i, \mathrm{dw}_i, \mathrm{dh}_i)$ for anchor i, which indicates scale-invariant translations and log-space height/width shifts relative to the pre-defined anchors, and $\mathbf{d}_i^\star$ is the associated offsets for anchor i relative to the ground-truth. Bounding box regression is only for positive anchors, as there is no ground-truth bounding box matched with negative ones.

For the final outputs of the whole system, we apply a multi-task loss for both detection and recognition:

$$\begin{aligned} L_{DRN} = &\frac{1}{\hat{N}}\sum_{i=1}^{\hat{N}} L_{cls}(\hat{p}_i, \hat{p}_i^\star) + \frac{1}{\hat{N}_+}\sum_{i=1}^{\hat{N}_+} L_{reg}(\hat{\mathbf{d}}_i, \hat{\mathbf{d}}_i^\star) \\ &+ \frac{1}{\hat{N}_+}\sum_{i=1}^{\hat{N}_+} L_{rec}(\mathbf{Y}^{(i)}, \mathbf{s}^{(i)}), \end{aligned} \tag{5}$$

where $\hat{N} \le 512$ is the number of text proposals sampled after hard negative mining, and $\hat{N}_+ \le 256$ is the number of positive ones. The thresholds for positive and negative anchors are set to 0.6 and 0.4 respectively, so as to increase the difficulty for text classification and regression, and improve the ability of TDN. $\hat{p}_i$ and $\hat{\mathbf{d}}_i$ are the outputs of TDN. $\mathbf{s}^{(i)} = \{\mathbf{s}_1^{(i)}, \ldots, \mathbf{s}_{T+1}^{(i)}\}$ is the ground-truth tokens for sample i, where $\mathbf{s}_{T+1}^{(i)}$ represents the special "END" token, and $\mathbf{Y}^{(i)} = \{\mathbf{y}_1^{(i)}, \ldots, \mathbf{y}_{T+1}^{(i)}\}$ is the corresponding output sequence of decoder LSTMs. $L_{rec}(\mathbf{Y}, \mathbf{s}) = -\sum_{t=1}^{T+1} \log \mathbf{y}_t(s_t)$ denotes the cross entropy loss on $\mathbf{y}_1, \ldots, \mathbf{y}_{T+1}$, where $\mathbf{y}_t(s_t)$ represents the predicted probability of the output being s_t at time-step t.

4 Experiments

In this section, we conduct extensive experiments to verify the effectiveness of the proposed method. We first describe the used datasets and the implementation details. Then, our model is compared against a few state-of-the-art methods on a number of standard benchmark datasets, including both regular and irregular text in natural scene images. Intermediate results are also demonstrated for ablation study. Finally, we compare the difference between the new model and the ones presented in our previous conference versions.

4.1 Datasets

The following datasets are used in our experiments for training and evaluation:

Synthetic Datasets (*SynT*) In [27], a fast and scalable engine was presented to generate synthetic images of text in clutter. A synthetic dataset with 800,000 images (denoted as "SynthText") was also released to public. It contains a large number of multi-oriented text instances, and is adopted widely in model pre-training.

ICDAR2013 (*IC13*) [57] This is the widely used dataset for scene text spotting from ICDAR2013 Robust Reading Competition. Images in this dataset explicitly focus around the text content of interest, which results in well-captured, nearly horizontal text instances. There are 229 images for training and 233 images for test. Text instances are annotated by horizontal bounding boxes with word-level transcriptions. There are 3 specific lists of words provided as lexicons for reference in the test phase, i.e., "Strong", "Weak" and "Generic". "Strong" lexicon provides 100 words per-image including all words appeared in the image. "Weak" lexicon contains all words appeared in the entire dataset, and "Generic" lexicon is a 90k word vocabulary proposed by [50].

ICDAR02015 (*IC15*) [58] This is another popular dataset from "Incidental Scene Text" of ICDAR2015 Robust Reading Competition. Images in this dataset are captured incidentally with Google Glasses, and hence most text instances are irregular (oriented, perspective and blurring). There are 1,000 images for training and 500 images for test. 3 scales of lexicons are also provided in test phase. The ground-truth for text is given by quadrangles and word-level annotations.

Total-Text (*TT*) [59] This dataset was released in ICDAR 2017, featuring curved text. More than half of its images have a combination of text instances with more than two orientations. There are 1,255 images in training set and 300 images for test. Text is annotated by polygon at the word level.

MLT [60] MLT is a large multi-lingual text dataset, which contains 7,200 training images, 1,800 validation images and 9,000 test images. Following FOTS [12], we also employ the "Latin" instances in training and validation images to enlarge our training data.

AddF2k [26] It contains 1,715 images with near horizontal text released in [26]. The images are annotated by horizontal bounding boxes and word-level transcripts.

COCO-Text (*CT*) [61] COCO-Text is by far the largest dataset for scene text detection and recognition. It consists of 43,686 images for training, 10,000 images for validation and another 10,000 for test. In our experiment, we collect all training and validation images for training. COCO-Text is created by annotating images from the MS COCO dataset, which contains images of complex scenes. As a result, this dataset is very challenging with text in arbitrary shapes. The ground-truth is given by word-level with top-left and bottom-right coordinates.

4.2 Implementation Details

In contrast to the work in our conference version [10] where the network is trained with TRN module locked initially, in this work, we train the whole network in an end-to-end fashion during the entire training process. This is achieved, we believe, with the benefit of better text proposals and RoI-Align methods. We use an approximate joint training process [25] to minimize the aforementioned two losses, i.e., L_{TPN} and L_{DRN} together, ignoring the derivatives with respect to the proposed boxes' coordinates.

The whole network is trained end-to-end on "SynthText" for 2 epochs first. Then we randomly sample 10k images from "SynthText", and combine with real training data to fine-tune the model for 20 epochs. Lastly, synthetic data is removed and the model is fine-tuned using only real data for another 15 epochs. Different real training datasets are used for different tasks, which will be discussed in the experiments.

We optimize our model using SGD with a batch size of 4, a weight decay of 0.0001 and a momentum of 0.9. The learning rate is set to 0.005 initially, with a decay rate of 0.8 every 30k iterations until it reaches 10^{-4} on the synthetic training data. When fine-tuning on real training images, the learning rate is decayed again with a rate of 0.8 every 30k iterations until it reaches 10^{-5}.

Data augmentation is also employed in the model training process. Specifically, 1) A multi-scale training strategy is used, where the shorter side of input image is randomly resized to three scales of $(600, 800, 1000)$ pixels, and the longer side is no more than 1200 pixels. 2) We randomly re-scale (with a probability of 0.5) the height of the image with a ratio from 0.8 to 1.2 without changing its width, so that the bounding boxes have more variable aspect ratios. 3) Images are rotated in the range of $[-10^{\circ}, 10^{\circ}]$ randomly with a probability of 0.4. 4) Images are randomly cropped from the input with a proportion of 0.9 and then resized to the original size.

During the test phase, we re-scale the input image into multiple sizes as well so as to cover the large range of bounding box scales. At each scale, 300 proposals with the highest textness scores are produced by TPN. Those proposals are re-identified by TDN and recognized by TRN simultaneously. A recognition score is then calculated by averaging the output probabilities. The ones with textness score larger than 0.5 and recognition score larger than 0.7 are kept and merged via NMS (non-maximum suppression) as the final output.

4.3 Evaluation Criterion

We follow the standard evaluation criterion in the end-to-end text spotting task: a bounding box is considered as correct if its IoU ratio with any ground-truth is greater than 0.5 and the recognized word also matches, ignoring the case. The ones with no longer than three characters and annotated as "do not care" are ignored. For the ICDAR2013 and ICDAR2015 datasets, there are two protocols: "End-to-End" and "Word Spotting". "End-to-End" protocol requires all words in the image to be recognized, no matter whether the string exists or not in the provided contextualised lexicon. "Word Spotting" on the other hand, only looks at the words that actually exist in the lexicon provided, ignoring all the rest that do not appear in the lexicon. There is no lexicon released in COCO-Text and Total-Text. Thus methods will be evaluated based on raw outputs, without using any prior knowledge. It should be noted that the location ground-truth is *rectangles* in ICDAR2013 and COCO-Text, *quadrangles* in ICDAR2015, and *polygons* in Total-Text.

4.4 Comparison With State-of-the-Art Methods

4.4.1 Experimental Results on ICDAR2013

The ICDAR2013 dataset is mostly used to evaluate model ability in detecting and recognizing horizontal text. We use

TABLE 1
Text Spotting Results on ICDAR2013 Dataset

Method	Backbone	Training data	ICDAR2013 Word-Spotting			ICDAR2013 End-to-End		
			Strong	Weak	Generic	Strong	Weak	Generic
Deep2Text II+ [1]	-	IC13	84.84	83.43	78.90	81.81	79.47	76.99
Jaderberg *et al.* [50]	CNN	SynT+IC13+IC03+SVT	90.49	–	76	86.35	–	–
FCRNall+multi-filt [27]	VGG16	SynT	–	–	84.7	–	–	–
TextBoxes [30]	VGG16	SynT+IC13	93.90	91.95	85.92	91.57	89.65	83.89
DeepTextSpotter [51]	GoogleNet	SynT+IC13+IC15	92	89	81	89	86	77
TextBoxes++ [4]	VGG16+CRNN	SynT+IC13	95.50	*94.79*	87.21	**92.99**	**92.16**	84.65
MaskTextSpotter* [13]	R50	SynT+IC13+IC15+TT	92.5	92.0	*88.2*	92.2	91.1	**86.5**
TextNet [53]	R50	SynT+IC13	94.59	93.48	86.99	89.77	88.80	82.96
AlignmentTextSpotter* [11]	PVA	SynT+IC13+IC15+MLT	93	92	87	91	89	*86*
FOTS [12]	R50	SynT+IC13+MLT	*95.94*	93.90	*87.76*	91.99	90.11	84.77
Ours	R50	SynT+IC13+IC15+MLT	**96.39**	**95.53**	**89.45**	*92.56*	*91.60*	85.49

We present F-measures here in percentage. Using ResNet50 as backbone, Our model achieves state-of-the-art performance on "Word-Spotting". The approaches marked with "" need to be trained with additional character-level annotations. In each column, the best result is shown in* **bold** *font, and the second best is shown in italic font.*

training images from IC13, IC15 and MLT during fine-tuning process. The text spotting results are presented and compared with other state-of-the-art methods in Table 1. Using ResNet50 as backbone, our model outperforms existing methods under "Word-Spotting" protocol, with average 1.2 percent improvement on F-measure. Results under "End-to-End" protocol are also comparable with other methods. Samples of results on ICDAR2013 are visualized in Fig. 4.

4.4.2 Experimental Results on ICDAR2015

We verify the effectiveness of the proposed model in spotting oriented text on the ICDAR2015 dataset. Real training images from IC13, IC15 and MLT are adopted during the fine-tuning process. As shown in Table 2, our model achieves state-of-the-art performance under two task settings with both protocols (excluding the "weak" one). Note that we have not used any lexicon in the "Generic" sub-task. The results are the raw outputs without using any prior knowledge about lexicon. Some qualitative results are presented in Fig. 5, with both quadrangle localizations and corresponding text labels. It can be seen that with the help of the spatial 2D attention weights, the improved framework is able to tackle irregular text.

4.4.3 Experimental Results on Total-Text

Next, we conduct experiments on the Total-Text dataset to evaluate the performance of our method for curved text. Real training images from IC13, IC15 and Total-Text are employed here for fine-tuning. As shown in Table 3, based on the same evaluation protocol as that used in [13] where the IoU is calculated by using polygon ground-truth, our model leads to an "End-to-End" performance of 58.56 percent without using any lexicon, which is about 4.5 percent higher than the best of the compared methods.

Text detection results on Total-Text are also presented here for reference. As our model is not delicately designed for text detection, the output bounding box does not enclose the text tightly, which leads to unsatisfactory detection performance under the detection evaluation criterion. However, the promising end-to-end performance exactly proves the strength and robustness of our 2D attention based text recognition model from another point of view. Compared to TextDragon [64] that leads to the best detection performance, its end-to-end result is about 10 percent lower than ours. Our method can correctly recognize text contained in loose bounding boxes. That is of practical importance from the viewpoint of text spotting. Some visualization results are presented in Fig. 6.

4.4.4 Experimental Results on COCO-Text

The COCO-text dataset is very challenging, not only because of the quantity, but also lying in the large variance of text appearance. Here, we fine-tune the pre-trained model with real training images from IC13, IC15, MLT and COCO-Text. The corresponding experimental results are shown in Table 4. It should be noted that text ground-truth in COCO-Text is given by rectangles. With this setting, our model achieves the highest text detection performance, which is 2 percent higher than the state-of-the-art. With the further integration of a strong text recognizer, our model finally achieves a surpassing text spotting performance. Several text spotting examples on COCO-Text are visualized in Fig. 7.

4.5 Ablation Experiments

In this section, a series of ablation experiments are carried out to analyze the design of each model part in detail. In ablation experiments, we use all real training images listed in Section 4.1 except COCO-Text during the fine-tuning process. Only the first and second data augmentation manners are adopted.

4.5.1 Joint Training Versus Separate Training

To validate the superiority of multi-task joint training, we build a two-stage system (denoted as "Ours (sep)") in which

Fig. 4. Examples of text spotting results on ICDAR2013. The red bounding boxes are both detected and recognized correctly. The green bounding boxes are missed words. The new model can cover more scales of text compared to the conference version [10]. For example, "SIXTH" and "EDITION" in the third image can be covered, which have a big space between characters.

the detection and recognition models are trained separately. For fair comparison, the detector in "Ours (sep)" is built by removing the recognition part from the original model and trained only with the detection loss. As for recognition, we employ our 2D-attention based text recognition network [16], but train it without extra training data. We can see from Table 5 that the two-stage system performs worse than the proposed model on ICDAR2013, ICDAR2015 and Total-Text. The results are consistent with that in our conference version [10], which prove again that the multi-task joint training would result in much better model parameters for feature extraction and lead to better end-to-end performance.

Furthermore, we compare the detection performance of the two-stage model and the jointly trained model. Note that for ICDAR2013, both detection results are achieved without referring to recognition results, while for ICDAR2015 and Total-Text, attention weights obtained during recognition phrase are used to rectify the detected bounding boxes. The detection results in Table 6 demonstrate that the recognition loss in model training can also benefit the detection performance. The proposed end-to-end model produces detection performance (F-measures) averagely 4 percent higher than that given by "Ours (sep)". The detection results on ICDAR2013 are comparable with the state-of-the-art under three evaluation criteria.

TABLE 2
Text Spotting Results on ICDAR2015 Dataset

Method	Backbone	Training data	ICDAR2015 Word-Spotting			ICDAR2015 End-to-End		
			Strong	Weak	Generic	Strong	Weak	Generic
Deep2Text-MO [1]	-	IC11	17.58	17.58	17.58	16.77	16.77	16.77
TextSpotter [62]	-	IC15	–	–	–	35.0	19.9	15.6
TextProposals + DictNet [40], [63]	CNN	SynT	56.00	52.26	49.73	53.30	49.61	47.18
DeepTextSpotter [51]	GoogleNet	SynT+IC13+IC15	58	53	51	54	51	47
TextBoxes++ [4]	VGG16+CRNN	SynT+IC15	76.45	69.04	54.37	73.34	65.87	51.90
ASTER [8]	VGG16+R50	SynT+IC15	75.2	71.3	67.6	70.6	67.3	64.0
MaskTextSpotter* [13]	R50	SynT+IC13+IC15+TT	79.3	74.5	64.2	79.3	73.0	62.4
TextNet [53]	R50	SynT+IC15	82.38	78.43	62.36	78.66	74.90	60.45
AlignmentTextSpotter* [11]	PVA	SynT+IC13+IC15+MLT	85	80	65	82	77	63
FOTS [12]	R50	SynT+IC13+IC15+MLT	87.01	**82.39**	67.97	83.55	**79.11**	65.33
TextDragon [64]	VGG16	SynT+IC15	86.22	81.62	68.03	82.54	78.34	65.15
Ours	R50	SynT+IC13+IC15+MLT	**87.80**	81.58	**68.21**	**84.23**	78.04	**65.53**

We present F-measures here in percentage. Our model achieves promising performance on both "Word-Spotting" and "End-to-End" settings, in comparison with other methods. The approaches marked with "" need to be trained with additional character-level annotations. In each column, the best performing result is shown in* **bold** *font, and the second best is shown in* *italic* *font.*

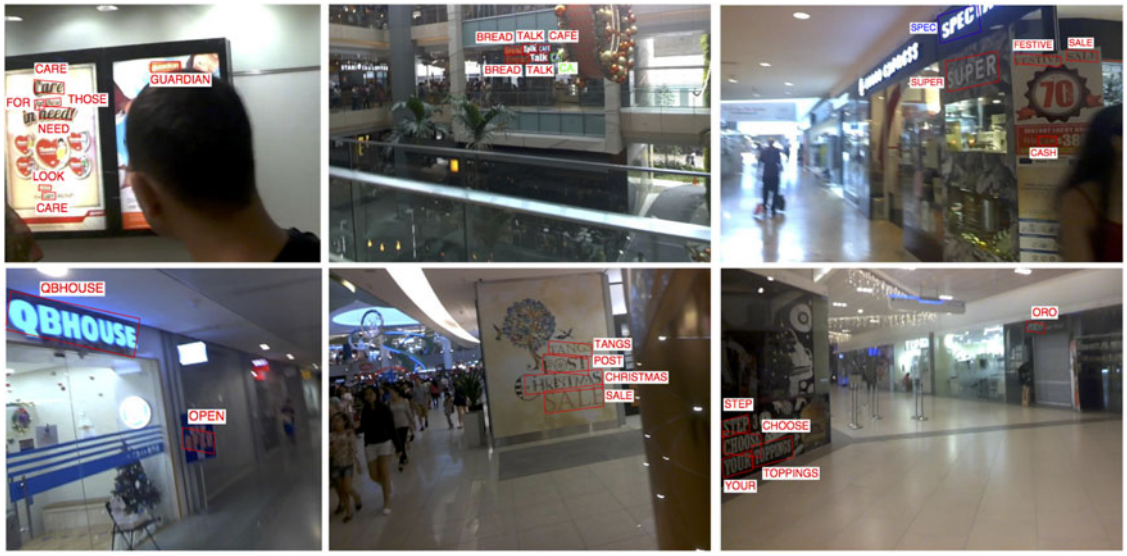


Fig. 5. Examples of text spotting results on ICDAR2015. The red bounding boxes are both detected and recognized correctly. The green bounding boxes are missed words, and the blue labels are wrongly recognized. With the employed 2D attention mechanism, our network is able to detect and recognize oriented text with a single forward pass in cluttered natural scene images.

TABLE 3
Text Detection and Text Spotting Results on Total-Text Dataset

Method	Backbone	Training data	Detection			End-to-End
			Recall	Precision	F-measure	F-measure
DeconvNet [59]	VGG16	SynT+IC13+IC15+MLT	33.0	40.0	36.0	–
TextBoxes [30]	VGG16	SynT+IC13	45.5	62.1	52.5	36.3
MaskTextSpotter [13]	R50	SynT+IC13+IC15+TT	55.0	69.0	61.3	52.9
TextNet [53]	R50	SynT+TT	59.45	68.21	63.53	54.02
TextDragon [64]	R50	SynT+TT	**75.7**	**85.6**	**80.3**	48.8
Ours	R50	SynT+IC13+IC15+TT	59.38	63.25	61.25	**58.56**

"Ours (New_R101)" achieves the best "End-to-End" performance among the compared methods. In each column, the best result is shown in **bold** *font, and the second best result is shown in italic font.*

Fig. 6. Text spotting examples on Total-Text. The red bounding boxes are both detected and recognized correctly. The blue ones are recognized incorrectly. The use of 2D attention mechanism enables our model detect and recognize curved text with a single forward pass in cluttered natural scene images.

TABLE 4
Text Detection and Text Spotting Results on COCO-Text Dataset

Method	Backbone	Training data	Detection			End-to-End
			Recall	Precision	F-measure	Average Precision
Yao *et al.* [65]	VGG16	IC13+IC15+MSRA-TD500	27.10	43.23	33.31	–
He *et al.* [33]	VGG16	IC13+IC15+ext.	31	46	37	–
EAST [2]	VGG16	ImageNet+CT	32.40	50.39	39.45	–
TO-CNN [66]	VGG16	NTU-UTOI	44	47	45	–
TextBoxes++ [4]	VGG16	SynT+CT	56.70	60.87	58.72	–
Lyu *et al.* [34]	VGG16	SynT+IC15	52.9	72.5	61.1	–
MaskTextSpotter+ [52]	R50	SynT+IC13+IC15+TT+AddF2k	**58.3**	66.8	62.3	23.9
Ours	R50	SynT+IC13+IC15+MLT+CT	56.60	**74.76**	**64.43**	**33.75**

Our method achieves state-of-the-art text detection performance, with F-measure outperforming the second best around 4 percent. The end-to-end performance is also promising.

Fig. 7. Text spotting examples on COCO-Text. The red bounding boxes are both detected and recognized correctly. The blue labels are wrongly recognized.

Nevertheless, they are worse than state-of-the-arts on irregular text datasets. Note that our work mainly focuses on the end-to-end text spotting scenario and only uses circumscribed rectangles as ground-truth bounding boxes for training. We leave the accurate text localization for future work.

4.5.2 Fixed-Size Versus Varying-Size RoI Pooling

Another contribution of this work is a varying-size RoI pooling mechanism, to accommodate the large variation of text aspect ratios. To validate its effectiveness, we compare the performance of models with varying-size RoI features ($H = 4$ and $W_{max} = 30$) and fixed-size ones. Different RoI pooling sizes are also tested for comparison.

Experimental results in Table 7 indicate that adopting varying-size RoI pooling improves the text spotting performance. After collecting statistics of the aspect ratios of bounding boxes in the training data, we set the width of RoI to the medium value of 20 in our comparison (the model is denoted as "Ours (fix4×20)"). It achieves the best text spotting performance among all fixed settings, but is still worse than using varying-size RoI pooling, with an 0.5 percent drop on F-measures. We visualize the attention heat maps based on varying-size RoI features and fixed-size RoI features

TABLE 5
Experiments on Multi-Task Learning

Model Name	Backbone	Attn.	Training	ICDAR2013 Word-Spotting			ICDAR2013 End-to-End			ICDAR2015 Word-Spotting			ICDAR2015 End-to-End			Total-Text
				Strong	Weak	Generic	Strong	Weak	Generic	Strong	Weak	Generic	Strong	Weak	Generic	
Former (sep) [10]	VGG-16	1D	Separate	92.94	90.54	84.24	88.20	86.06	81.97	-	-	-	-	-	-	-
Former (full) [10]	VGG-16	1D	Joint	94.16	92.42	88.20	91.08	89.81	84.59	-	-	-	-	-	-	-
Ours (sep)	R50+FPN	2D	Separate	95.95	94.31	88.35	91.28	90.90	84.35	82.67	77.27	63.82	79.86	75.12	61.04	57.82
Ours	R50+FPN	2D	Joint	96.35	94.87	88.90	92.13	91.25	84.74	85.64	80.45	65.84	82.21	77.14	63.55	58.72

We present F-measures here in percentage. "Former" ones show results from previous conference version [10]. Experimental results from both the conference version [10] and our proposed model proves that the joint training of the whole framework benefits the end-to-end task.

TABLE 6
Text Detection Results on ICDAR2013, ICDAR2015, and Total-Text

Model Name	ICDAR2013			ICDAR2015	Total-Text
	ICDAR standard	DetEval	IoU		
Jaderberg *et al.* [50]	-	-	76.2	-	-
FCRNall+multi-filt [27]	-	-	84.2	-	-
CTPN [24]	82.2	87.7	-	61	-
TextBoxes [30]	85	86	-	-	52.5
SSTD [33]	87	88	-	77	-
RRPN [32]	–	91	-	80.2	-
AlignmentTextSpotter_Det [11]	88	88	-	83	-
AlignmentTextSpotter [11]	90	90	-	87	-
FOTS_Det [12]	86.9	87.3	-	85.3	-
FOTS [12]	92.5	92.8	-	89.8	-
MaskTextSpotter [13]	-	91.7	-	86.0	61.3
TextNet [53]	91.3	91.4	-	87.4	63.5
Former (sep) [10]	-	-	83.4	-	-
Former (full) [10]	-	-	85.6	-	-
Ours (sep)	87.7	88.5	87.5	79.5	60.1
Ours	91.6	92.3	90.3	85.0	61.5

F-measures are presented here in percentage. We use standard detection evaluation criteria proposed in each dataset for fair comparison. The inclusion of the recognition loss in model training greatly enhances the detection performance.

TABLE 7
Ablation Experiments on RoI Pooling Manner

Model Name	Backbone	Attn.	RoI	RoI Spec.	ICDAR2013 Word-Spotting			ICDAR2013 End-to-End			ICDAR2015 Word-Spotting			ICDAR2015 End-to-End			Total-Text
					Strong	Weak	Generic	Strong	Weak	Generic	Strong	Weak	Generic	Strong	Weak	Generic	
Former (fix) [10]	VGG-16	1D	F	4×20	93.33	91.66	87.73	90.72	87.86	83.98	-	-	-	-	-	-	-
Former (full) [10]	VGG-16	1D	V	4×35	94.16	92.42	88.20	91.08	89.81	84.59	-	-	-	-	-	-	-
Ours (fix4×30)	R50+FPN	2D	F	4×30	95.27	94.24	86.83	91.45	89.95	83.27	85.24	79.24	64.92	81.72	74.75	62.40	57.42
Ours (fix4×20)	R50+FPN	2D	F	4×20	96.33	94.32	88.44	92.56	90.26	84.29	85.29	79.47	65.47	82.12	75.10	63.08	57.47
Ours (fix7×7)	R50+FPN	2D	F	7×7	96.17	94.30	86.63	92.03	90.28	83.47	81.58	77.77	62.83	78.52	73.59	60.49	50.82
Ours	R50+FPN	2D	V	4×30	96.35	94.87	88.90	92.13	91.25	84.74	85.64	80.45	65.84	82.21	77.14	63.55	58.72

*"Former" ones show results from the conference version [10]. "RoI" column means using **F**ixed-size or **V**arying-size RoI pooling. "RoI Spec." shows the RoI feature size specification, where the maximum size is presented for **V**arying-size RoI pooling. The proposed varying-size RoI pooling is more flexible in handling the large variability on text scales and aspect ratios, and leads to the best performance. Note that we use $W_{max} = 35$ in [10] but 30 here, which leads to a faster running speed but without sacrificing model performance.*

respectively. As shown in Fig. 8, fixed-size RoI pooling may lead to a large portion of information loss for long words.

To peer off the impact of RoI pooling size adopted on model performance, we also test the model that uses $W = 30$ in the fixed setting (denoted as "Ours (fix4×30)"). The F-measures drop about 1 percent comparing with the results by using "Ours (fix4×20)". This result indicates that a longer W adopted in the fixed setting may not be beneficial, as most image features will be distorted and stretched in that case. The deformation would be more serious for short words. These experiments also demonstrate the flexibility of our varying-size pooling method in dealing with the large diversity of text aspect ratios.

In addition, we test the model that pools RoI features to 7×7 (named as "Ours (fix 7×7)"), which is originally used in Faster-RCNN [25]. F-measures also decrease on all test datasets compared to the proposed model, especially on Total-Text.

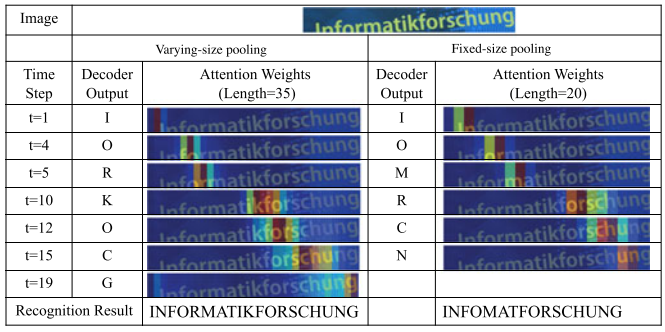


Fig. 8. Attention mechanism based sequence decoding process by varying-size and fixed-size RoI features separately. The heat maps show that at each time step, the position of the character to be decoded has higher attention weights, so that the corresponding local features are extracted and assist the text recognition. However, if we use the fixed-size RoI pooling, information may be lost during pooling, especially for a long word, which leads to an incorrect recognition result. In contrast, the varying-size RoI pooling preserves more information and leads to a correct result.

TABLE 8
Ablation Study on RoI Feature Encoding Methods

Model Name	Encoding in TDN	ICDAR2013 Word-Spotting			ICDAR2013 End-to-End			ICDAR2015 Word-Spotting			ICDAR2015 End-to-End			Total-Text
		Strong	Weak	Generic	Strong	Weak	Generic	Strong	Weak	Generic	Strong	Weak	Generic	
Ours (AP)	AvePooling	94.13	92.63	85.06	90.45	89.49	82.96	83.62	78.70	62.85	80.14	74.99	60.60	55.95
Ours (AP+FC)	AvePooling+FC	94.23	93.36	87.61	90.75	89.76	83.56	83.62	78.80	63.34	80.34	74.82	60.76	57.06
Ours	LSTMs	96.35	94.87	88.90	92.13	91.25	84.74	85.64	80.45	65.84	82.21	77.14	63.55	58.72

Compared to average pooling, LSTMs show superiority on sequential feature encoding.

TABLE 9
Ablation Experiments on Model Architecture

Model Name	Backbone	Attn.	RNN Enc.	Box Ref.	ICDAR2013 Word-Spotting			ICDAR2013 End-to-End			ICDAR2015 Word-Spotting			ICDAR2015 End-to-End			Total-Text
					Strong	Weak	Generic	Strong	Weak	Generic	Strong	Weak	Generic	Strong	Weak	Generic	
Former (full) [10]	VGG-16	1D	Share	N	94.16	92.42	88.20	91.08	89.81	84.59	-	-	-	-	-	-	-
Ours (1D)	R50+FPN	1D	Share	N	95.26	93.94	88.13	91.93	90.17	84.26	80.76	76.03	61.14	77.40	73.38	59.08	49.85
Ours (2D)	R50+FPN	2D	Share	N	96.87	95.15	88.06	92.61	91.05	84.18	82.12	77.70	61.96	78.75	74.43	60.49	50.39
Ours (Shr)	R50+FPN	2D	Share	Y	96.87	95.15	88.06	92.61	91.05	84.18	85.70	80.14	65.61	82.43	76.35	63.30	58.20
Ours	R50+FPN	2D	Sep.	Y	96.35	94.87	88.90	92.13	91.25	84.74	85.64	80.45	65.84	82.21	77.14	63.55	58.72

"Former (full) [10]" shows the results from previous conference version. "RNN Enc." shows whether TDN and TRN share 1 layer of RNN encoder or not. "Box Ref." means whether performing box refinement. Experiments are conducted not only on regular text dataset, but also on irregular ones for comprehensive evaluation.

4.5.3 Effect of RoI Encoding Manner

In the proposed framework, LSTMs are adopted to convert the varying-length RoI features into a fixed-size for the following text detection and recognition networks. Instead of using LSTMs, here, we extract a fixed-size holistic feature by average pooling across RoI features (named as "Ours (AP)"). A performance degradation of nearly 2 percent on F-measures is received compared to using LSTMs in Table 8. Furthermore, we test the model with an extra 1024D Fully-connected layer after average pooling (named as "Ours (AP +FC)") and find a slight improvement. However, the proposed model still performs significantly better than "Ours (AP+FC)". These experiments illustrate the effectiveness of LSTMs on sequential feature encoding.

4.6 Improvements Over Our Preliminary Results in [10]

To be specific, there are four major improvements on the model architecture over the previous conference version [10], i.e., backbone network, attention structures, box refinement, and the re-arrangement of RNN encoder. In this subsection, we prove the effect of each part in detail, so as to better understand our model.

4.6.1 Effect of Backbone Network

The conference version [10] used VGG-16 as the backbone network. Only the final convolutional layer is adopted for RoI feature extraction. In contrast, the new model adopts ResNet50 with FPN as the backbone, which extracts RoI features from different levels of feature pyramid according to their scales. As compared in Table 9 between "Former (full) [10]" and "Ours (1D)", the new backbone framework gives F-measures gain around 1 percent on ICDAR2013, mostly because of a higher recall.

4.6.2 1D vs. 2D Attention

By comparing the results between "Ours (1D)" and "Ours (2D)", we find that using 2D Attention instead of 1D attention gives an roughly 1 percent improvement to accuracy. It is worth to note that even for the horizontal text in ICDAR2013, the 2D attention mechanism is still better than the 1D counterpart. The reason may be caused by the more accurate character-level feature extraction during decoding process. We also visualize the 2D attention heat maps in Fig. 9. Although trained in a weakly supervised manner (which means that it is trained without character-level annotations), the attention model can approximately localize each character to be decoded, which, on one hand, extracts local feature for character recognition, on other hand, roughly indicates character alignment for bounding box refinement.

4.6.3 Box Refinement

The proposed box refinement process is used together with 2D attention to boost text localization performance. As shown in

Fig. 9. Visualization of 2D attention heat map for each word proposal by aggregating attention weights at all character decoding steps. The results show that the 2D attention model can approximately localize characters, which provides assistance in both word recognition and bounding box rectification. Images are from ICDAR2015 in the first row and Total-Text in the second row. The red bounding boxes are both detected and recognized correctly. The green bounding boxes are missed words.

TABLE 10
Ablation Experiments on Box Refinement Manner

Method	Detection			End-to-End
	Recall	Precision	F-measure	F-measure
Ours_Quad	60.23	62.76	61.47	58.72
Ours_Poly	60.75	63.05	61.88	59.11

Fitting 6-point polygons on Total-Text can bring performance improvement of 0.4 percent averagely on F-measures, compared to the quadrangle counterpart.

Fig. 10. Comparison between polygon and quadrangle fitting results on total-text.

Table 9, our model with box refinement ("Ours (shr)") significantly outperforms that without box refinement ("Ours (2D)") on the irregular text datasets (roughly 3 and 7 percent improvement on ICDAR2015 and Total-Text respectively). Box refinement is not performed for regular text, so the performance of "Ours (shr)" and "Ours (2D)" on ICDAR2013 is the same. These results demonstrate that box refinement is a simple yet effective method for improving irregular text localization.

Moreover, we have attempted to fit 6-point polygons on Total-Text, according to the attention heat map. To be specific, for words with more than 3 characters, we fit two quadrangles according to the attention weights in front and real half respectively. The two quadrangles are then connected and form a polygon for the word. The coordinates of the two junction points are the mean values of the closest coordinates from the two quadrangles. Compared to the quadrangle results, the performance is indeed improved, as demonstrated in Table 10. Visualization comparison is presented in Fig. 10.

4.6.4 Sharing of LSTM Encoders

In the conference version [10], a layer of LSTM encoder with 1024 hidden states is shared between TDN and TRN. Another layer of LSTMs with 1024 units is adopted in TRN. While in the new model, the LSTM encoders are separated in TDN and TRN. One layer of LSTM encoder with 1024 states are applied in TDN, and two layers of LSTM encoder with 512 states are used in TRN. This modification leads to almost 4M less parameters and 20ms speeding-up (on Titan X), without significantly affecting the accuracy (comparing "Ours (shr)" and "Ours" in Table 9).

4.6.5 Speed

Owing to hyper-parameter tuning (including reducing W_{max} from 35 to 30, reducing the size of two FC layers in TDN from 2048 to 1024, and the above LSTM encoder

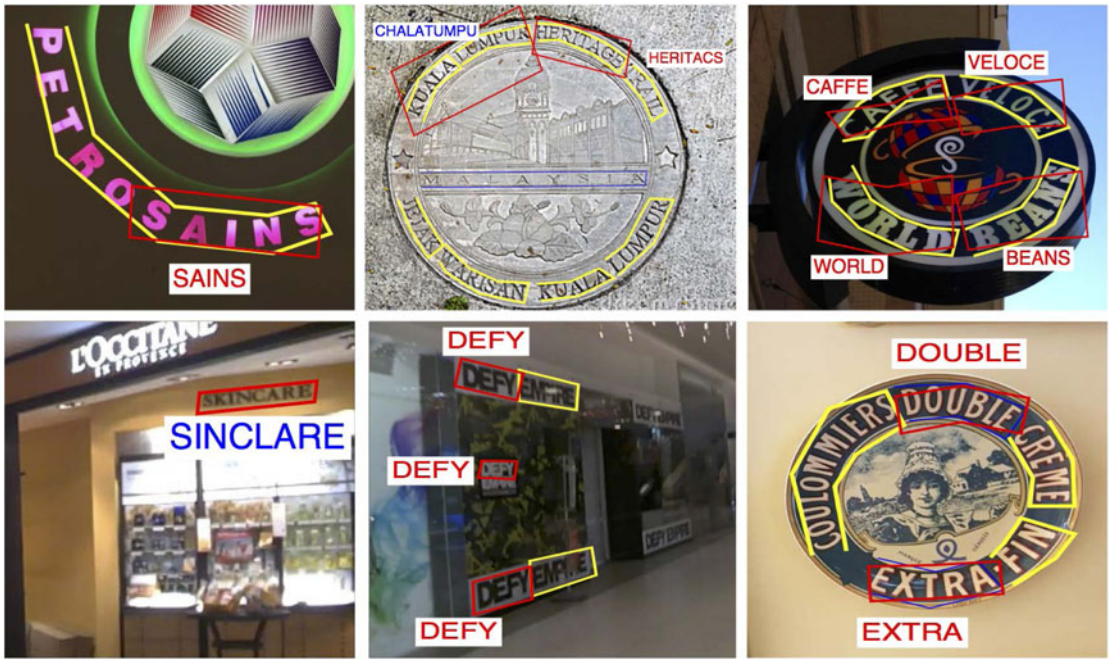


Fig. 11. Failure cases of our model. The red bounding boxes and labels are our detection and recognition results. The yellow ones are missed detection, which cannot be correctly recognized. The blue labels are wrongly recognized, although been well detected. The failure reasons are half detection, low contrast, bad refinement, small words, blurred, vertical text in sequence. The failure reasons are half detection, low contrast, bad refinement, small words, blurred, vertical text in sequence.

adjustment) and a better implementation (replacing `for` loops with GPU-friendly code), the overall running speed of the new model is around 2 times faster than the conference version (0.5s *vs*. 1s for processing a 720×1280 image on a Titan-X GPU).

4.7 Failure Cases Analysis

The failure of text spotting can be caused either by inaccurate detection results or by false recognition results. If the text is not detected or the bounding box only covers part of text, recognition is doomed to fail. In addition, it sometimes fails to recognize the word even it has been detected correctly. As presented in Fig. 11, there are a variety of reasons for the failure, such as text that appears in a low contrast against the background, text that is of a very small size, uncommon fonts or blurred. Our work is also incapable of spotting vertical text. Moreover, the heuristic box refinement process based on the attention maps is not perfect. As indicated in Fig. 9, the heat map may not hit in the middle of each character, but drifts to the upper or lower side. It may sometimes fail to rectify the bounding box. There is much room for improvement.

5 Conclusion

In this paper we have presented a simple end-to-end trainable network for simultaneous text detection and recognition in natural scene images. A novel RoI encoding method has been proposed, considering the large diversity of aspect ratios of word bounding boxes. We use a 2D attention model that is capable of indicating character locations accurately, which assists word recognition as well as text localization. Being robust to different forms of text layouts, our approach performs well for both regular and irregular scene text.

For future work, one potential direction is to use convolutions or self-attention to replace the recurrent networks used in the framework, so as to speed up the computation. Our current framework may fail to recognize text that is aligned vertically, which deserves further study.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grants U19B2037, 61876152, National Key R&D Program of China under Grant 2020AAA0106900, and Ningbo Natural Science Foundation under Grant 202003N4369. Peng Wang and Hui Li equally contributed to this work.

References

[1] X.-C. Yin, X. Yin, K. Huang, and H.-W. Hao, "Robust text detection in natural scene images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 5, pp. 970–983, May 2014.

[2] X. Zhou *et al.*, "EAST: An efficient and accurate scene text detector," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2642–2651.

[3] Y. Liu and L. Jin, "Deep matching prior network: Toward tighter multi-oriented text detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 3454–3461.

[4] 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.

[5] 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. 10552–10561.

[6] 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.

[7] Z. Cheng, Y. Xu, F. Bai, Y. Niu, S. Pu, and S. Zhou, "AON: Towards arbitrarily-oriented text recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 5571–5579.

[8] 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. 2018.

[9] F. Zhan and S. Lu, "ESIR: End-to-end scene text recognition via iterative image rectification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2059–2068.

[10] H. Li, P. Wang, and C. Shen, "Towards end-to-end text spotting with convolutional recurrent neural networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 5238–5246.

[11] 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. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 5020–5029.

[12] X. Liu, D. Liang, S. Yan, D. Chen, Y. Qiao, and J. Yan, "FOTS: Fast oriented text spotting with a unified network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 5676–5685.

[13] P. Lyu, M. Liao, C. Yao, W. Wu, and X. Bai, "Mask TextSpotter: An end-to-end trainable neural network for spotting text with arbitrary shapes," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 71–88.

[14] 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.

[15] 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.

[16] 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 Conf. Artif. Intell.*, 2019, pp. 8610–8617.

[17] S. Long, X. He, and C. Yao, "Scene text detection and recognition: The deep learning Era," *Int. J. Comput. Vis.*, vol. 129, pp. 161–184, 2021.

[18] 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.

[19] Y. Zhu, C. Yao, and X. Bai, "Scene text detection and recognition: Recent advances and future trends," *Front. Comput. Sci.*, vol. 10, no. 1, pp. 19–36, 2016.

[20] X.-C. Yin, Z.-Y. Zuo, S. Tian, and C.-L. Liu, "Text detection, tracking and recognition in video: A comprehensive survey," *IEEE Trans. Image Process.*, vol. 25, no. 6, pp. 2752–2773, Jun. 2016.

[21] M. Jaderberg, A. Vedaldi, and A. Zisserman, "Deep features for text spotting," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 512–528.

[22] W. Huang, Y. Qiao, and X. Tang, "Robust scene text detection with convolution neural network induced MSER trees," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 497–511.

[23] Z. Zhang, W. Shen, C. Yao, and X. Bai, "Symmetry-based text line detection in natural scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 2558–2567.

[24] 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.

[25] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[26] Z. Zhong, L. Jin, S. Zhang, and Z. Feng, "DeepText: A new approach for text proposal generation and text detection in natural images," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2017, pp. 1208–1212.

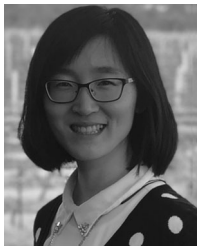
[27] A. Gupta, A. Vedaldi, and A. Zisserman, "Synthetic data for text localisation in natural images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2315–2324.

[28] J. Redmon, S. Divvala, R. B. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 779–788.

[29] W. Liu *et al.*, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 21–37.

[30] M. Liao, B. Shi, X. Bai, X. Wang, and W. Liu, "Textboxes: A fast text detector with a single deep neural network," in *Proc. AAAI Conf. Artif. Intell.*, 2017, pp. 4161–4167.
[31] Z. Zhang, C. Zhang, W. Shen, C. Yao, W. Liu, and X. Bai, "Multi-oriented text detection with fully convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4159–4167.
[32] J. Ma *et al.*, "Arbitrary-oriented scene text detection via rotation proposals," *IEEE Trans. Multimedia*, vol. 20, no. 11, pp. 3111–3122, Nov. 2018.
[33] P. He, W. Huang, T. He, Q. Zhu, Y. Qiao, and X. Li, "Single shot text detector with regional attention," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 3047–3055.
[34] P. Lyu, C. Yao, W. Wu, S. Yan, and X. Bai, "Multi-oriented scene text detection via corner localization and region segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7553–7563.
[35] P. D. R. G. Kaiming He, Georgia Gkioxari, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2018, pp. 2961–2969.
[36] 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.*, 2019, pp. 9038–9045.
[37] 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.
[38] Z. Tian, M. Shu, P. Lyu, R. Li, C. Zhou, X. Shen, and J. Jia, "Learning shape-aware embedding for scene text detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4234–4243.
[39] K. Wang, B. Babenko, and S. Belongie, "End-to-end scene text recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 1457–1464.
[40] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, "Synthetic data and artificial neural networks for natural scene text recognition," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2014.
[41] P. He, W. Huang, Y. Qiao, C. C. Loy, and X. Tang, "Reading scene text in deep convolutional sequences," in *Proc. 30th AAAI Conf. Artif. Intell.*, 2016, pp. 3501–3508.
[42] A. Graves, S. Fernandez, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2006, pp. 369–376.
[43] C.-Y. Lee and S. Osindero, "Recursive recurrent nets with attention modeling for OCR in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2231–2239.
[44] B. Shi, X. Wang, P. Lv, C. Yao, and X. Bai, "Robust scene text recognition with automatic rectification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4168–4176.
[45] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 3104–3112.
[46] F. Bai, Z. Cheng, Y. Niu, S. Pu, and S. Zhou, "Edit probability for scene text recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 1508–1516.
[47] M. Jaderberg, K. Simonyan, A. Zisserman, "Spatial transformer networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 2017–2025.
[48] W. Liu, C. Chen, and K.-Y. K. Wong, "Char-Net: A character-aware neural network for distorted scene text recognition," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 7154–7161.
[49] Z. Cheng, F. Bai, Y. Xu, G. Zheng, S. Pu, and S. Zhou, "Focusing attention: Towards accurate text recognition in natural images," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 5086–5094.
[50] 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, 2015.
[51] M. Bušta, L. Neumann, and J. Matas, "Deep TextSpotter: An end-to-end trainable scene text localization and recognition framework," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2223–2231.
[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] Y. Sun, C. Zhang, Z. Huang, J. Liu, J. Han, and E. Ding, "TextNet: Irregular text reading from images with an end-to-end trainable network," in *Proc. Asian. Conf. Comput. Vis.*, 2018, pp. 83–99.
[54] S. Qin, A. Bissacco, M. Raptis, Y. Fujii, and Y. Xiao, "Towards unconstrained end-to-end text spotting," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 4704–4714.
[55] L. Xing, Z. Tian, W. Huang, and M. R. Scott, "Convolutional character networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 9126–9136.
[56] 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. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 9809–9818.
[57] D. Karatzas *et al.*, "ICDAR 2013 robust reading competition," in *Proc. Int. Conf. Document Anal. Recognit.*, 2013, pp. 1484–1493.
[58] D. Karatzas *et al.*, "ICDAR 2015 robust reading competition," in *Proc. Int. Conf. Document Anal. Recognit.*, 2015, pp. 1156–1160.
[59] C. K. Chng and C. S. Chan, "Total-text: A comprehensive dataset for scene text detection and recognition," in *Proc. Int. Conf. Document Anal. Recognit.*, 2017, pp. 935–942.
[60] N. Nayef *et al.*, "ICDAR2017 robust reading challenge on multi-lingual scene text detection and script identification - RRC-MLT," in *Proc. Int. Conf. Document Anal. Recognit.*, 2017, pp. 1454–1459.
[61] A. Veit, T. Matera, L. Neumann, J. Matas, and S. Belongie, "COCO-Text: Dataset and benchmark for text detection and recognition in natural images," 2016, *arXiv:1601.07140*.
[62] L. Neumann and J. Matas, "Real-time lexicon-free scene text localization and recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 9, pp. 1872–1885, Sep. 2016.
[63] L. Gómez and D. Karatzas, "TextProposals: A text-specific selective search algorithm for word spotting in the wild," *Pattern Recognit.*, vol. 70, pp. 60–74, 2017.
[64] 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. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 9076–9085.
[65] C. Yao, X. Bai, N. Sang, X. Zhou, S. Zhou, and Z. Cao, "Scene text detection via holistic, multi-channel prediction," 2016, *arXiv:1606.09002*.
[66] S. Prasad and A. W. K. Kong, "Using object information for spotting text," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 540–557.

Peng Wang received the bachelor's degree in electrical engineering and automation, and the PhD degree in control science and engineering from Beihang University, China, in 2004 and 2011, respectively. He is currently a professor with the School of Computer Science, Northwestern Polytechnical University, China. His research interests include computer vision, machine learning and artificial intelligence. Then, he was with The University of Adelaide for about four years.



Hui Li received the PhD degree from the University of Adelaide, in 2018. She was a research fellow with the University of Adelaide, Australia. Her research interests include scene text detection and recognition, visual question answering, etc.

Chunhua Shen is currently an adjunct professor with Monash University, Australia. His research interests include computer vision and machine learning.