# Towards End-to-End Text Spotting in Natural Scenes

Hui Li*, Peng Wang*, 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.**

**In contrast to existing approaches that consider text detection and recognition as two distinct tasks and tackle them one by one, the proposed framework settles these two tasks concurrently. The whole framework can be trained end-to-end and is able to handle text of arbitrary shapes. The convolutional features are calculated only once and shared by both detection and recognition modules. Through multi-task training, the learned features become more discriminate and improve the overall performance. By employing the 2D attention model in word recognition, the irregularity of text can be robustly addressed. It 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. We show that our proposed method can achieve state-of-the-art performance on several widely-used text spotting benchmarks, including both regular and irregular datasets.**

***Index Terms*—End-to-end scene text spotting, Deep neural network, Attention model**

## I. 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 potential applications. For example, accessing text in images can help blind person understand the environment they are involved, understanding road signs will make automatic vehicles work securely; indexing text within 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 obtain the localization of text in images, in terms of bounding boxes, while word recognition attempts to output human readable text transcriptions. Compared to traditional OCR, text spotting in natural scene images is even more challenging because of the extreme diversity of text patterns and highly complicated background. Text appearing in natural scene images can be of varying fonts, sizes, shapes and layouts. It may be distorted by strong lighting, occlusion, blurring or orientation. The background usually contains a large amount of noise and text-like outliers, such as windows, railings, bricks.

An intuitive approach to scene text spotting is to divide it into two separated sub-tasks. Text detection is carried out firstly to obtain candidate text bounding boxes, and word recognition is performed subsequently on the cropped regions to output transcriptions. Numerous approaches have been developed which solely focus on text detection [1], [2], [3], [4], [5] or word recognition [6], [7], [8], [9]. Methods are improved from only handling simple horizontal text to addressing complicated irregular (oriented or curved) text. However, these two sub-tasks are highly correlated and complementary. On one hand, the feature information may be shared between them to save computation. On the other hand, the multi-task training can improve feature representation power and benefit both sub-tasks.

To this end, some end-to-end approaches are proposed recently to concurrently tackle both sub-tasks [10], [11], [12], [13]. It should be noted that most end-to-end approaches pay more attention on designing a sophisticated detection module, so as to acquire tighter bounding boxes around the text, which would alleviate the challenges for word recognition. Nevertheless, the ultimate goal of text spotting is to let the machine know what is on the image, instead of struggling on exact bounding box locations. Hence, in this work, we leave the challenge of text irregularity to the recognition part. To be more specific, the detection module is designed to output a rectangular bounding box for each word, no matter what text appearance is (horizontal, oriented or curved). A robust recognition module, which shares image features with the detection module, is devised to recognize the text within the relatively loose bounding box. The overall framework of our method is presented in Figure 1. It makes use of ResNet-101 [14] as the backbone, with Feature Pyramid Networks (FPN) [15] embedded for strong semantic feature learning. Text Proposal network (TPN) is adapted to multiple levels on feature pyramid so as to obtain text proposals at different scales. A RoI pooling layer is then employed to extract varying-size 2D features from each proposal, which are then concurrently used in text detection network and word recognition network. A 2-dimensional attention network is employed in the word recognition module. On one hand, it is able to select local features for individual character during decoding process so as to improve recognition accuracy. On the other hand, it indicates the character alignment in word bounding

H. Li and C. Shen are with School of Computer Science, The University of Adelaide, Adelaide, SA, 5005, Australia; and Australian Centre for Robotic Vision. Correspondence should be addressed to C. Shen (e-mail: chunhua.shen@adelaide.edu.au).

P. Wang is with the School of Computer Science, Northwestern Polytechnical University, China.

*The first two authors equally contributed to this work.

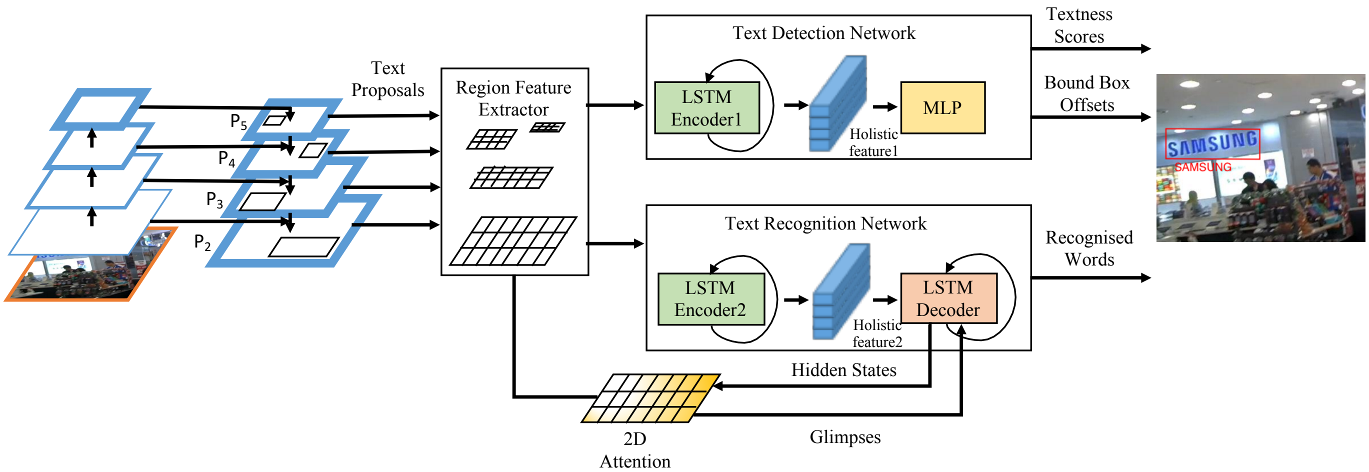


**Fig. 1** – The overall architecture of our proposed model for end-to-end text spotting in natural scene image. The network takes an image as input, and outputs both text bounding boxes and text labels in one single forward pass. The entire network is trained end-to-end.

box, which can be used to refine the loose bounding box. The recognition module can also help reject false positives in detection phase, thus improving the overall performance.

Preliminary results of this study appeared in Li *et al.* [10], which is *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. This work here is an extension of [10]. 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 by horizontal text as in [10].
2) We now use ResNet with FPN as the backbone network, leading to significantly better feature representations. We also adapt the text proposal network with pyramid feature maps. The two modifications are able to propose text instances at a wide range of scales and improve the recall of small size text.
3) The training process is simplified. 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 whole 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 three-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 the appearance of the text in that it can handle arbitrary-oriented text. The convolutional features are shared by both detection and recognition modules, which saves computation in comparison with addressing them separately by two distinct models. In addition, the multi-task optimization benefits the feature learning, and thus promotes the detection results as well as 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 keep sufficient information which is valuable for the following 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 to refine the bounding boxes. It should be noted 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 to solving the end-to-end text spotting problem. Conventional methods have been built on the idea of accurate and tight bounding boxes around the text being the first-step output, so as to exclude redundant noise and benefit word recognition. Our work grounds on a strong and robust word recognition model, which, in turn, can complement the detection results and finally lead to an intact end-to-end text spotting framework. Our model achieves the state-of-the-art experimental results on several standard text spotting benchmarks, including ICDAR2013, ICDAR2015, Total-Text and COCO-Text.

## II. Related Work

In this section, we introduce some related work on text detection, word recognition and end-to-end text spotting methods. There are comprehensive surveys for scene text detection and recognition in [16], [17], [18], [19].

**Text Detection** With the development of deep learning techniques, text detection in natural scene images achieves significant progress. Methods are springing up rapidly, from detecting regular horizontal text to multi-oriented or even curved text. The location annotation is also more delicate, from horizontal rectangle to quadrangle and polygon.

Methods in the early stage including [20], [21] 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.* [22] proposed to extract text lines by exploiting text symmetry property compared to background. Tian *et al.* [23] developed a vertical anchor mechanism, and proposed a Connectionist Text Proposal Network (CTPN) to accurately localize text lines in image. The developments on general object detection and segmentation provide a lot of inspirations for text detection. Inspired by Faster-RCNN [24], Zhong *et al.* [25] 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.* [26] used a Fully-Convolutional Regression Network (FCRN) for efficient text detection and bounding box regression, motivated by YOLO [27]. Similar to SSD [28], Liao *et al.* [29] proposed "TextBoxes" by combining predictions from multiple feature maps with different resolutions. Those methods are mainly for regular text, which output horizontal rectangles.

In [30], the authors proposed to localize text lines via salient maps that are calculated by Fully Convolutional Networks (FCN). Post-processing techniques are proposed to extract text lines in multiple orientations. Ma *et al.* [31] introduced Rotation Region Proposal Networks (RRPN) to generate inclined proposals with text orientation angle. A Rotation Region-of-Interest (RRoI) pooling layer was designed for feature extraction. He *et al.* [32] proposed to use an attention mechanism to identify text regions from image. A hierarchical inception module was developed to aggregate multi-scale inception features. 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 or text-line 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. Quadrilateral sliding windows were used to recall text and a sequential protocol was designed for relative regression of compact quadrangle. Liao *et al.* [4] improved "TextBoxes" to produce additional orientation angle or quadrilateral bounding box offsets so as to detect oriented scene text (referred to as "TextBoxes++"). Lyu *et al.* [33] 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.

Most recently, more advanced methods are proposed to produce polygons which aim to fit text appearance even better. For example, inspired by Mask R-CNN [34], Xie *et al.* [35] proposed to detect arbitrary shape text based on FPN [15] and instance segmentation. A supervised pyramid context network was introduced to precisely locate text regions. 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) [36] is to perform pixel-level segmentation for precisely locating text instance with arbitrary shape. The PSE algorithm was introduced to generate different scales of kernels and expend to complete shape. Tian *et al.* [37] treated text detection as an instance-level segmentation. Pixels belonging to the same word are pulled together as connected component while pixels from different words are pushed away from each other.

Our work on text detection part is based on Faster R-CNN framework [24], 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 at a variety of scales and aspect ratios, FPN [15] is adopted here to generate text proposals with both higher recall and precision. Since our ultimate target is end-to-end text spotting, we also use the horizontal rectangle that encloses the whole word as the ground-truth. Horizontal rectangles already contain sufficient information to text spotting. Besides, the whole framework can be simplified as we do not need additional modules to handle text orientation. A more preciser bounding box can be obtained according to word recognition results.

**Word Recognition** Word recognition means to recognize the cropped word image patches into character sequences. Early work for scene text recognition adopts a bottom-up fashion [38], [20], which detects individual characters firstly and integrates them into a word by means of dynamic programming, or a top-down manner [39], which treats the word patch as a whole and recognizes it as a multi-class image 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 usually employed for sequential feature learning. Recognition methods have also been developed significantly, from only handling horizontal text to recognizing arbitrary shape text.

The work in [40] and [6] considered word recognition as one-dimensional sequence labeling problem. RNNs are employed to model the sequential features. A Connectionist Temporal Classification (CTC) layer [41] is adopted to decode the whole sequences, eliminating character separation. Wang and Hu [42] proposed a Gated Recurrent Convolutional Neural Network (GRCNN) with CTC for regular text recognition. The works in [43] and [44] were proposed to recognize text using an attention-based sequence-to-sequence framework [45]. In this manner, RNNs are able to learn the character-level language model hidden in the word strings from the training data. A 1D soft-attention model was adopted to select relevant local features during decoding characters. The RNN+CTC and sequence-to-sequence frameworks serve as two meta-algorithms that are widely used by subsequent text recognition 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 degrade the recognition accuracy. They proposed "Edit Probability" to tackle this misalignment problem.

The rapid progress on regular text recognition has given rise to increasing attention on recognizing irregular ones. Shi *et al.* [8], [44] rectified oriented or curved text based on Spatial Transformer Network (STN) [47] and then performed recognition using a 1D attentional sequence-to-sequence model. ESIR [9] employed a line-fitting transformation to estimate the pose of text, and developed a pipline that iteratively removes perspective distortion and text line curvature to drive a better recognition performance. 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 extra character-level annotations. Yang *et al.* [49] introduced an auxiliary dense character detection task into the encoder-decoder network to handle the irregular text. Pixel-level character annotations are required to train the network. Cheng *et al.* [50] 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 feature and target. Character-level bounding box annotations is also requested in this work. Cheng *et al.* [7] applied LSTMs in four directions to encode arbitrarily-oriented text. A filtering mechanism was designed to integrate these redundant features and reduce irrelevant ones. The work in [51] 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 adopt it in the recognition module, and train together with the detection parts towards an end-to-end text spotting system.

**End-to-End Text Spotting** Most previous methods design a multi-stage pipeline to achieve text spotting. For instance, Jaderberg *et al.* [52] generated a large number of text proposals using ensemble models, and then adopted the word classifier in [39] for recognition. Gupta *et al.* [26] employed FCRN for text detection and the word classifier in [39] for recognition. Liao *et al.* [4] combined "TextBoxes++" and "CRNN" [6] to complete the text spotting task. The work in [8] combines "TextBoxes" [29] and a rectification based recognition method for text spotting.

Preliminary results of the work here, presented in [10], may be the first, in parallel with [53] to explore a unified end-to-end trainable framework for concurrent text detection and recognition. Although in one single framework, the work in [53] does not share any features between detection and recognition parts, which can be seen as a loose combination. Our previous work [10] shares the RoI features for both detection and recognition, which saves computation. At the same time, the joint optimization of multi-task loss can also improve feature learning, thus boosting detection performance in return. Nevertheless, one drawback of [10] is that the method can only process horizontal scene text. 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 greatly. However, character level annotations are needed for supervision. Contemporaneously, Liu *et al.* [12] presented "FOTS" that applies "RoIRotate" to share convolutional features between detection and recognition for oriented text. 1D sequential features are extracted via several sequential convolutions and bi-directional RNNs, and decoded by the CTC layer. Both work may encounter difficulty in dealing with curved or distorted scene text, which do not have obvious text orientation. Lyu *et al.* [13] proposed "Mask TextSpotter" that introduces a mask branch for character instance segmentation, inspired by Mask R-CNN [34]. It can detect and recognize text of various shapes, including horizontal, oriented and curved text, but character-level mask information is needed for training. Sun *et al.* [54] proposed "TextNet" to read irregular text. It outputs quadrangle text proposals. A perspective RoI transform was developed to extract features from arbitrary-size quadrangle for recognition. Four directional RNNs are adopted to encode the irregular text instances, and worked as context feature for the following spatial attention mechanism in decoding process.

In contrast to designing a sophisticated framework to handle the variety of text shape and expression form, which, potentially, increases the model complexity, we resort to the conventional horizontal bounding box for text location representation in our model. It not only provides sufficient information to complete the text spotting task, but also leads to a considerably simpler model. We postpone the processing of text irregularity to the flexible yet strong 2D attention model in word recognition—the second module of the proposed end-to-end framework.

## III. Model

The overall architecture of our proposed model is illustrated in Figure 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 5 components: 1) a ResNet CNN working as 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 for proposal recognition.

Simplicity is at the core of our design. Hence, we exclude additional modules for handling the irregularity of text shapes. 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 part of the model in detail.

### A. Backbone

A pre-trained ResNet-101 [14] is used here as the backbone convolutional layers for its state-of-the-art 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 gives rise 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.

### B. 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 $3 \times 3$ convolution and two sibling $1 \times 1$ 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 size 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 totally 20 anchors over the feature pyramid, which are capable of covering text instances with different shapes.

The heads with $3 \times 3$ conv and two $1 \times 1$ 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] exactly.

### C. 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 [34] 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 following 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 resulted 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.

### D. 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 once 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 $\mathbf{h_d}_t$ by a non-linear function: $\mathbf{h_d}_t = \mathrm{f}(\mathbf{q}_t, \mathbf{h_d}_{t-1})$. In this recurrent fashion, the final hidden state $\mathbf{h_d}_W$ (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 $\mathbf{h_d}_W$, followed by two parallel layers for classification and bounding box regression respectively.

To boost the detection performance, an online hard negative mining is adopted during the training stage. We firstly 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 $30\%$ random sampled ones. Through this processing, we observe that the text detection performance can be improved significantly.

### E. 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 encoder-decoder network for text recognition. 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 discriminate 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 Figure 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

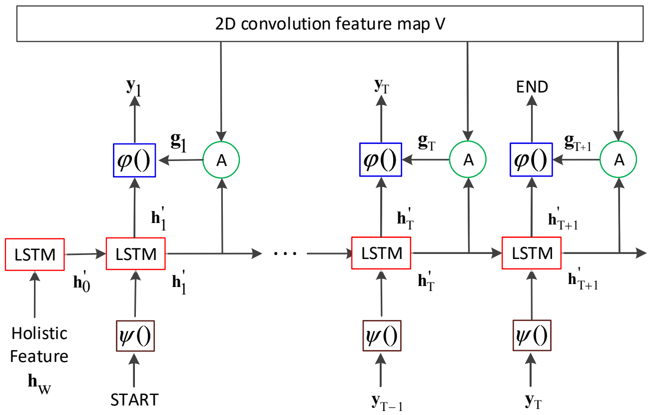


**Fig. 2** – The structure of the LSTM decoder used in this work. The holistic feature $\mathbf{h}_W$, a "START" token and the previous outputs are input into LSTM subsequently, terminated by an "END" token. At each time step $t$, the output $y_t$ is computed by $\varphi(\cdot)$ with the current hidden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

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 follows:

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

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 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; $\alpha_{ij}$ 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 discriminate 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 Figure 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.

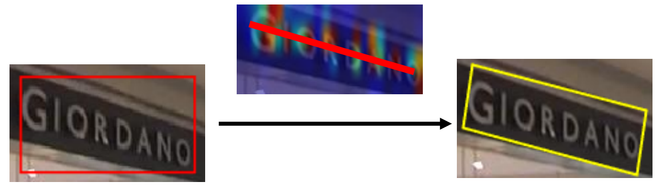


**Fig. 3** – Box refinement according to character alignment indexed by attention weights.

## F. 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 $L_{reg}$ [24] 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].

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 $\hat{N} \leq 512$ is the number of text proposals sampled after hard negative mining, and $\hat{N}_+ \leq 256$ is the number of positive ones. The thresholds for positive and negative anchors are set to 0.6 and 0.4 respectively, which are less strict than those used for training TPN. $\hat{p}_i$ 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 $t$.

## IV. Experiments

In this section, we perform extensive experiments to verify the effectiveness of the proposed method. We first introduce a few datasets and present the implementation details. Some intermediate results are also demonstrated for ablation study. Our model is evaluated on a number of standard benchmark datasets, including both regular and irregular text in natural scene images.

### A. Datasets

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

**Synthetic Datasets** In [26], 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 for public. Considering the complexity of our model, we follow the idea of curriculum learning [55], and generate another $48,000$ images (denoted as "Synth-Simple") using the engine, with words randomly placed on simple *pure colour backgrounds* (10 words per image on average). The words are sampled from the "Generic" lexicon [52] of size 90k.

**ICDAR**2013 [56] This is the widely used dataset for scene text spotting, from the "Focused Scene Text" of 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 [52].

**ICDAR**2015 [57] 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** [58] This dataset was released in ICDAR2017, featuring curved-oriented 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 in test set. Text is annotated by polygon at the word level.

**MLT** [59] MLT is a large multi-lingual text dataset, which contains $7,200$ training images, $1,800$ validation images and $9,000$ test images. As introduced in FOTS [12] to enlarge the training data, we also employ the "Latin" instances in training and validation images during training phase. Because our proposed model is only for reading English words, we cannot test the model on MLT test dataset.

**AddF2k** [25] It contains $1,715$ images with near horizontal text instances released in [25]. The images are annotated by horizontal bounding boxes and word-level transcripts. All images are used in training phase.

**COCO-Text** [60] 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 everyday 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. Images in this dataset are only used to fine-tune the model.

### B. Implementation Details

In contrast to the work in our conference version [10] where the network is trained with the 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 [24] 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 "Synth-Simple" for 20k iterations firstly and on "SynthText" for 200k iterations secondly. Then real training data excluding COCO-Text is adopted to fine-tune the model for 50k iterations and another 80k iterations including COCO-Text training data.

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 $0.001$ initially, with a decay rate of $0.8$ every 10k iterations until it reaches $5 \times 10^{-5}$ 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 20k iterations until it reaches $10^{-5}$.

Data augmentation is also adopted 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 rescale (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.

During the test phase, we rescale 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.

### C. Experimental Results

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 that all words in the image are 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 the evaluation in COCO-Text and Total-Text. Thus methods are 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.

*1) Experimental Results on ICDAR2013:* The end-to-end text spotting results on ICDAR2013 are presented in Table I. *Our new proposed model outperforms existing methods by a large margin under "Word-Spotting" protocol, and achieves comparable performance under "End-to-End" protocol.*

The superiority is even more obvious when using a general lexicon. Some text spotting examples are presented in Figure 4. As compared with the results in [10], the new model can cover more text size and appearance.

Our former work [10] is the first attempt to solve text spotting in a unified, end-to-end trainable framework, with both text detection and recognition accomplished simultaneously. It is inspired by the basic Faster R-CNN [24] system, with VGG-16 without FPN employed as the backbone. The anchors are of multiple pre-defined scales and aspect ratios. TPN is only working on top of a single-scale convolutional feature map, as well as the region feature extractor. 1D attentions model is employed in TRN for text recognition. The one using varying length RoI pooling is denoted as "Ours-Former (Ours Atten+Vary)", and the one using fixed-size RoI pooling is denoted as "Ours-Former (Ours Atten+Fixed)". We also build a two-stage system (denoted as "Ours-Former (Two-stage)") in order to demonstrate the superiority of end-to-end jointed training. Some insights can be obtained from the experimental results.

**Joint Training vs. Separate Training**

Most previous works [52], [26], [29] on text spotting typically perform in a two-stage manner, where detection and recognition are trained and processed by two unrelated models separately. The text bounding boxes detected by a model need to be cropped from the image and then recognized by another model. In contrast, our proposed model is trained jointly by a multi-task loss for both detection and recognition. With multi-task loss supervision, the learned features are more discriminate and lead to better performance for both tasks.

To validate the superiority of multi-task joint training, we build a two-stage system (denoted as "Ours-Former (Two-stage)") in which detection and recognition models are trained separately. For fair comparison, the detector in "Ours-Former (Two-stage)" is built by removing the recognition part from model "Ours-Former (Atten+Vary)" and trained only with the detection objective (denoted as "Ours DetOnly"). As for recognition, we employ CRNN [6] that produces state-of-the-art performance on text recognition. Model "Ours-Former (Two-stage)" firstly adopts "Ours DetOnly" to detect text with the same multi-scale inputs. CRNN is then followed to recognize the detected bounding boxes. We can see from Table I that model "Ours-Former (Two-stage)" performs worse than "Ours-Former(Atten+Vary)" on both settings on ICDAR2013.

Furthermore, we also compare the detection-only performance of these two systems. Note that "Ours DetOnly" and the detection part of "Ours-Former (Atten+Vary)" share the same architecture, but they are trained with different strategies: "Ours DetOnly" is optimized with only the detection loss, while "Ours-Former (Atten+Vary)" is trained with a multi-task loss for both detection and recognition.

In consistent with the "End-to-End" evaluation criterion, a detected bounding box is considered to be correct if its IoU ratio with any ground-truth is greater than 0.5. The detection results are presented in Table II. Without any lexicon used, "Ours-Former (Atten+Vary)" produces a detection performance with F-measures of 85.6% on ICDAR2013, which is 2% higher than that given by "Ours DetOnly". *This result illustrates that detector performance can be improved via joint training*.

**Fixed-size vs. 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 "Ours-Former (Atten+Vary)" (RoI features of size $H = 4$ and $W_{max} = 35$) and "Ours-Former (Atten+Fixed)" (RoI features of fixed-size $4 \times 20$).

Experimental results in Table I indicate that adopting varying-size RoI pooling increases the F-measures by around 1%, compared to using fixed-size pooling. We also visualize the attention heat maps based on varying-size RoI features and fixed-size RoI features respectively. As shown in Figure 5, fixed-size RoI pooling may lead to a large portion of information loss for long words.

*2) Experimental Results on ICDAR2015:* We verify the effectiveness of the new proposed model in detecting and recognizing oriented text on the ICDAR2015 dataset. Based on the improved backbone and 2D attention model, our method is now able to spotting oriented text effectively. As presented in Table III, *our method achieves state-of-the-art performance under three task settings with both protocols*. Actually, we have not used any lexicon in the "Generic" sub-task. The result is the raw output without using any prior knowledge.


**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 text size and appearance 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.

**TABLE I** – Text spotting results on ICDAR2013 dataset. We present the F-measure here in percentage. "Ours-Former" indicates the model presented in the previous conference version, which use VGG-Net without FPN as backbone and 1D attention in TRN. "Ours-New" denotes the current model. "Ours-Former(Two-stage)" uses separate models for detection and recognition, while other "Ours" models are end-to-end trained. "Ours-New" achieves the best performance on "Word-Spotting" setting and the second best on "End-to-End" setting, in comparing with both other methods and our former method. 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 result is shown in *italic* font.

| Method | ICDAR2013 Word-Spotting | | | ICDAR2013 End-to-End | | |
|---|---|---|---|---|---|---|
| | Strong | Weak | Generic | Strong | Weak | Generic |
| Deep2Text II+ [1] | 84.84 | 83.43 | 78.90 | 81.81 | 79.47 | 76.99 |
| Jaderberg *et al.* [52] | 90.49 | – | 76 | 86.35 | – | – |
| FCRNall+multi-filt [26] | – | – | 84.7 | – | – | – |
| TextBoxes [29] | 93.90 | 91.95 | 85.92 | 91.57 | 89.65 | 83.89 |
| DeepTextSpotter [53] | 92 | 89 | 81 | 89 | 86 | 77 |
| TextBoxes++ [4] | 95.50 | *94.79* | 87.21 | **92.99** | **92.16** | 84.65 |
| MaskTextSpotter* [13] | 92.5 | 92.0 | 88.2 | 92.2 | 91.1 | 86.5 |
| TextNet [54] | 94.59 | 93.48 | 86.99 | 89.77 | 88.80 | 82.96 |
| AlignmentTextSpotter* [11] | 93 | 92 | 87 | 91 | 89 | 86 |
| FOTS [12] | *95.94* | 93.90 | *87.76* | 91.99 | 90.11 | *84.77* |
| Ours-Former(Two-stage) [10] | 92.94 | 90.54 | 84.24 | 88.20 | 86.06 | 81.97 |
| Ours-Former(Atten+Fixed) [10] | 93.33 | 91.66 | 87.73 | 90.72 | 87.86 | 83.98 |
| Ours-Former(Atten+Vary) [10] | 94.16 | 92.42 | 88.20 | 91.08 | 89.81 | 84.59 |
| Ours-New | **97.70** | **96.05** | **89.05** | *92.53* | *91.17* | **84.86** |

**TABLE II** – Text detection results on different datasets. Precision (P) and Recall (R) at maximum F-measure (F) are reported in percentage. The jointly trained model ("Ours-Former (Atten+Vary)") gives better detection results than the one trained with detection loss only ("Ours DetOnly").

| Method | ICDAR2013 | | |
|---|---|---|---|
| | R | P | F |
| Jaderberg *et al.* [52] | 68.0 | 86.7 | 76.2 |
| FCRNall+multi-filt [26] | 76.4 | **93.8** | 84.2 |
| Ours DetOnly | 78.5 | 88.9 | 83.4 |
| Ours Atten+Vary | **80.5** | 91.4 | **85.6** |

However, our model shows an even better performance, which demonstrates the practicality of our proposed approach.

Some qualitative results are presented in Figure 7, with both quadrangle localizations and corresponding text labels shown. It can be seen that with the help of the spatial 2D attention weights, the improved framework is able to tackle irregular cases well.

We also visualize the 2D attention heat maps for some images in Figure 6. Although trained in a weakly supervised manner, the well-trained attention model can approximately localize each character to be decoded, which, on one hand, extracts local feature for character recognition, on the other hand, indicates character alignment for refining word bounding boxes.

*3) Experimental Results on Total-Text:* Next, we conduct experiments on the Total-Text dataset to demonstrate the results of our method in detecting and recognizing curved text. As shown in Table IV, our method leads to an "End-to-End" performance of 57.46% without using any lexicon, which is about 3.5% higher than the state-of-the-art.

**TABLE III** – Text spotting results on ICDAR2015 dataset. We present the F-measure here in percentage. "Ours-New" achieves the best performance on "Word-Spotting" setting and the second best on "End-to-End" setting, in comparing 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 result is shown in *italic* font.

| Method | ICDAR2015 Word-Spotting | | | ICDAR2015 End-to-End | | |
|---|---|---|---|---|---|---|
| | Strong | Weak | Generic | Strong | Weak | Generic |
| Deep2Text-MO [1] | 17.58 | 17.58 | 17.58 | 16.77 | 16.77 | 16.77 |
| TextSpotter [61] | – | – | – | 35.0 | 19.9 | 15.6 |
| TextProposals + DictNet [62], [39] | 56.00 | 52.26 | 49.73 | 53.30 | 49.61 | 47.18 |
| DeepTextSpotter [53] | 58 | 53 | 51 | 54 | 51 | 47 |
| TextBoxes++ [4] | 76.45 | 69.04 | 54.37 | 73.34 | 65.87 | 51.90 |
| ASTER [8] | 75.2 | 71.3 | 67.6 | 70.6 | 67.3 | 64.0 |
| MaskTextSpotter* [13] | 79.3 | 74.5 | 64.2 | 79.3 | 73.0 | 62.4 |
| TextNet [54] | 82.38 | 78.43 | 62.36 | 78.66 | 74.90 | 60.45 |
| AlignmentTextSpotter* [11] | 85 | 80 | 65 | 82 | 77 | 63 |
| FOTS [12] | *87.01* | **82.39** | *67.97* | *83.55* | **79.11** | *65.33* |
| Ours-New | **87.67** | *82.33* | **68.73** | **84.36** | *78.89* | **66.06** |

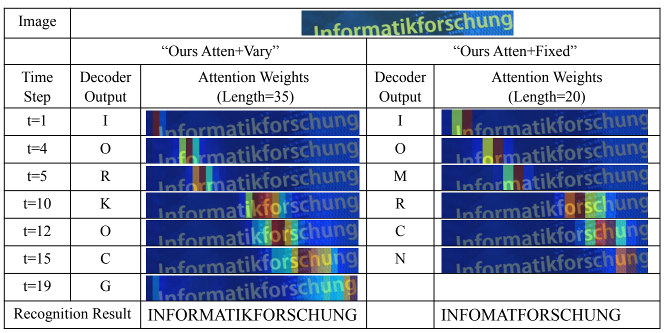


**Fig. 5** – Attention mechanism based sequence decoding process by "Ours-Former (Atten+Vary)" and "Ours-Former (Atten+Fixed)" separately. The heat maps show that at each time step, the position of the character to be decoded has higher attention weights, so that the corresponding local features are 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.

Some visualization results are presented in Figure 8. In fact, our model is not delicately designed for curved text, but the promising result proves the robustness of our 2D attention based model again. Although our method outputs rectangles initially, the contained text can be correctly recognized. That is adequate from the viewpoint of text spotting. Moreover, if we use rectangle ground-truth bounding boxes, the end-to-end F-measure can be increased to 60%.

**TABLE IV** – Text detection and text spotting results on Total-Text dataset. "Ours-New" achieves the best "End-to-End" performance, which is 3.5% higher than the second best. In each column, the best performing result is shown in **bold** font, and the second best result is shown in *italic* font.

| Method | Detection | | | End-to-End |
|---|---|---|---|---|
| | Recall | Precision | F-measure | F-measure |
| DeconvNet [58] | 33.0 | 40.0 | 36.0 | – |
| TextBoxes [29] | 45.5 | 62.1 | 52.5 | 36.3 |
| MaskTextSpotter* [13] | 55.0 | **69.0** | 61.3 | 52.9 |
| TextNet [54] | *59.45* | *68.21* | **63.53** | *54.02* |
| Ours-New | **59.79** | 64.76 | *62.18* | **57.80** |

*4) Experimental Results on COCO-Text:* The COCO-text dataset contains $10,000$ images for test without any lexicon provided. It is very challenging, not only because of the quantity, but also lying in the large variance of text appearance. Actually the COCO data is not originally proposed by text, hence images were not collected with text in mind and thus contain a broad variety of text instances. As there are not many results reported on this dataset, we set up a baseline for the following work. In addition, we find that our model achieves state-of-the-art text detection performance, compared with published results.

*5) Speed:* Using an NVIDIA Titan X GPU, the new proposed model takes approximately 0.7s to process an input image of $720 \times 1280$ pixels, which is 1.3 times faster than the previous conference version although we use a deeper backbone. However, it is slower than current methods such as [12], [13]. We further analyze the computation speed of each stage and find the about 36% of the computation time is used for RoI pooling because of the implementation, which


**Fig. 6** – 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.

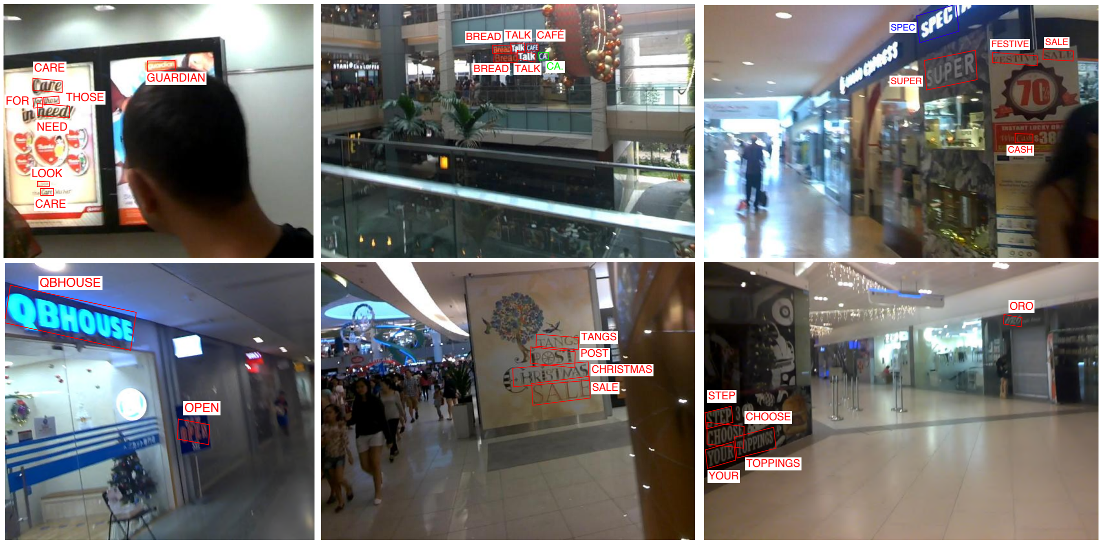


**Fig. 7** – Examples of text spotting results on ICDAR2015. The red bounding boxes are both detected and recognized correctly. The green bounding boxes are missed words, and the blue labels are wrongly recognized. With the employed 2D attention mechanism, our network is able to detect and recognize oriented text with a single forward pass in cluttered natural scene images.


**Fig. 8** – Examples of text spotting results on Total-Text. The red bounding boxes are both detected and recognized correctly. The blue ones are recognized incorrectly. With the employed 2D attention mechanism, our network is able to detect and recognize curved text with a single forward pass in cluttered natural scene images.

**TABLE V** – Text detection and text spotting results on COCO-Text dataset. Our method achieves state-of-the-art text detection performance, with F-measure outperforming the second best around 6%.

| Method | Detection | | | End-to-End |
|---|---|---|---|---|
| | Recall | Precision | F-measure | Average Precision |
| Yao *et al.* [63] | 27.1 | 43.23 | 33.31 | – |
| He *et al.* [32] | 31 | 46 | 37 | – |
| EAST [2] | 32.4 | 50.39 | 39.45 | – |
| TO-CNN [64] | 44 | 47 | 45 | – |
| TextBoxes++ [4] | 56.7 | 60.87 | 58.72 | – |
| Ours-New | **58.36** | **76.55** | **66.23** | **34.01** |

is unreasonable. We leave the code optimization as our future work.

## V. Conclusions

In this paper we have presented a unified end-to-end trainable network for simultaneous text detection and recognition in natural scene images. Based on an improved backbone with feature pyramid network, text proposals can be generated with a much higher recall. A novel RoI encoding method has been proposed, considering the large diversity of aspect ratios of word bounding boxes. The 2D attention model is capable of indicating character locations accurately, which assists word recognition as well as text localization. Being


**Fig. 9** – Examples of text spotting results on COCO-Text. The red bounding boxes are both detected and recognized correctly. The blue labels are wrongly recognized.

robust to different forms of text layouts, our approach performs very well for both regular and irregular scene text.

For future work, one potential direction is to use convolutions or self-attention to take place of the recurrent networks used in the framework, so as to speed up the computation. Another direction is to explore context information in the image, such as object, scene, etc., to help text detection and recognition. How to recognize text aligned vertically also deserves further study.

## 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, 2014.

[2] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, and J. Liang, "EAST: An efficient and accurate scene text detector," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2017.

[3] Y. Liu and L. Jin, "Deep matching prior network: Toward tighter multi-oriented text detection," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2017.

[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, 2018.

[5] C. Zhang, B. Liang, Z. Huang, M. En, J. Han, E. Ding, and X. Ding, "Look more than once: An accurate detector for text of arbitrary shapes," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2019.

[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, 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. Comp. Vis. Patt. Recogn.*, 2018.

[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.*, pp. 1–1, 2018.

[9] F. Zhan and S. Lu, "ESIR: End-to-end scene text recognition via iterative image rectification," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2019.

[10] H. Li, P. Wang, and C. Shen, "Towards end-to-end text spotting with convolutional recurrent neural networks," in *Proc. IEEE Int. Conf. Comp. 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. Comp. Vis. Patt. Recogn.*, 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. Comp. Vis. Patt. Recogn.*, 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. Comp. 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. Comp. Vis. Patt. Recogn.*, 2016.

[15] T.-Y. Lin, P. Dollr, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2017.

[16] S. Long, X. He, and C. Yao, "Scene text detection and recognition: The deep learning era," *CoRR*, vol. abs/1811.04256, 2018.

[17] 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, 2015.

[18] Y. Zhu, C. Yao, and X. Bai, "Scene text detection and recognition: recent advances and future trends," *Frontiers of Computer Science*, vol. 10, no. 1, pp. 19–36, 2016.

[19] 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, 2016.

[20] M. Jaderberg, A. Vedaldi, and A. Zisserman, "Deep features for text spotting," in *Proc. Eur. Conf. Comp. Vis.*, 2014.

[21] W. Huang, Y. Qiao, and X. Tang, "Robust scene text detection with convolution neural network induced mser trees," in *Proc. Eur. Conf. Comp. Vis.*, 2014.

[22] Z. Zhang, W. Shen, C. Yao, and X. Bai, "Symmetry-based text line detection in natural scenes," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2015.

[23] 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. Comp. Vis.*, 2016.

[24] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015.

[25] 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. Acoustics, Speech & Signal Processing*, 2017.

[26] A. Gupta, A. Vedaldi, and A. Zisserman, "Synthetic data for text localisation in natural images," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2016.

[27] J.Redmon, S. Divvala, R. B. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2016.

[28] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *Proc. Eur. Conf. Comp. Vis.*, 2016.

[29] 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. Artificial Intell.*, 2017.

[30] 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. Comp. Vis. Patt. Recogn.*, 2016.

[31] J. Ma, W. Shao, H. Ye, L. Wang, H. Wang, Y. Zheng, and X. Xue, "Arbitrary-oriented scene text detection via rotation proposals," *IEEE Trans. Multimedia*, vol. 20, no. 11, pp. 3111–3122, 2018.

[32]

[33] 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. Comp. Vis. Patt. Recogn.*, 2018.

[34] P. D. R. G. Kaiming He, Georgia Gkioxari, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comp. Vis.*, 2018.

[35] 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. Artificial Intell.*, 2019.

[36] W. Wang, E. Xie, X. Li, W. Hou, T. Lu, G. Yu, and S. Shao, "Shape robust text detection with progressive scale expansion network," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2019.

[37] 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. Comp. Vis. Patt. Recogn.*, 2019.

[38] K. Wang, B. Babenko, and S. Belongie, "End-to-end scene text recognition," in *Proc. IEEE Int. Conf. Comp. Vis.*, 2011.

[39] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, "Synthetic data and artificial neural networks for natural scene text recognition," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014.

[40] P. He, W. Huang, Y. Qiao, C. C. Loy, and X. Tang, "Reading scene text in deep convolutional sequences," in *Proc. AAAI Conf. Artificial Intell.*, 2016.

[41] 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.

[42] J. Wang and X. Hu, "Gated recurrent convolution neural network for ocr," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017.

[43] C.-Y. Lee and S. Osindero, "Recursive recurrent nets with attention modeling for ocr in the wild," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2016.

[44] B. Shi, X. Wang, P. Lv, C. Yao, and X. Bai, "Robust scene text recognition with automatic rectification," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2016.

[45] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Adv. 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. Comp. Vis. Patt. Recogn.*, 2018.

[47] M. Jaderberg, K. Simonyan, A. Zisserman *et al.*, "Spatial transformer networks," in *Proc. Adv. 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. Artificial Intell.*, 2018.

[49] X. Yang, D. He, Z. Zhou, D. Kifer, and C. L. Giles, "Learning to read irregular text with attention mechanisms," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, 2017, pp. 3280–3286.

[50] 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. Comp. Vis.*, 2017, pp. 5086–5094.

[51] 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. Artificial Intell.*, 2019.

[52] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, "Reading text in the wild with convolutional neural networks," *Int. J. Comp. Vis.*, vol. 116, no. 1, pp. 1–20, 2015.

[53] M. Buta, L. Neumann, and J. Matas, "Deep textspotter: An end-to-end trainable scene text localization and recognition framework," in *Proc. IEEE Int. Conf. Comp. Vis.*, 2017, pp. 2223–2231.

[54] 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. Asi. Conf. Comp. Vis.*, 2018.

[55] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proc. Int. Conf. Mach. Learn.*, 2009.

[56] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L. G. i Bigorda, S. R. Mestre, J. Mas, D. F. Mota, J. A. Almazan, and L. P. de las Heras, "ICDAR 2013 robust reading competition," in *Proc. Int. Conf. Doc. Anal. Recog.*, 2013.

[57] D. Karatzas, L. Gomez-Bigorda, A. Nicolaou, S. Ghosh, A. Bagdanov, M. Iwamura, J. Matas, L. Neumann, V. R. Chandrasekhar, S. Lu, F. Shafait, S. Uchida, and E. Valveny, "ICDAR 2015 robust reading competition," in *Proc. Int. Conf. Doc. Anal. Recog.*, 2015.

[58] C. K. Chng and C. S. Chan, "Total-text: A comprehensive dataset for scene text detection and recognition," in *Proc. Int. Conf. Doc. Anal. Recog.*, 2017, pp. 935–942.

[59] N. Nayef, F. Yin, I. Bizid, H. Choi, Y. Feng, D. Karatzas, Z. Luo, U. Pal, C. Rigaud, J. Chazalon, W. Khlif, M. Luqman, J.-C. Burie, C.-L. Liu, and J.-M. Ogier, "Icdar2017 robust reading challenge on multi-lingual scene text detection and script identification - rrc-mlt," in *Proc. Int. Conf. Doc. Anal. Recog.*, 2017, pp. 1454–1459.

[60] A. Veit, T. Matera, L. Neumann, J. Matas, and S. Belongie, "Coco-text: Dataset and benchmark for text detection and recognition in natural images," *CoRR*, vol. abs/1601.07140, 2016.

[61] 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, 2017.

[62] L. Gmez and D. Karatzas, "Textproposals: A text-specific selective search algorithm for word spotting in the wild," *Pattern Recogn.*, vol. 70, pp. 60–74, 2017.

[63] C. Yao, X. Bai, N. Sang, X. Zhou, S. Zhou, and Z. Cao, "Scene text detection via holistic, multi-channel prediction," *CoRR*, vol. abs/1606.09002, 2016.

[64] S. Prasad and A. W. K. Kong, "Using object information for spotting text," in *Proc. Eur. Conf. Comp. Vis.*, 2018.