# An End-to-End TextSpotter with Explicit Alignment and Attention

Tong He[1,*], Zhi Tian[1,4,*], Weilin Huang[3], Chunhua Shen[1,†], Yu Qiao[4], Changming Sun[2]

[1]University of Adelaide, Australia [2]CSIRO Data61, Australia
[3]Malong Technologies Co., Ltd, Shenzhen, China
[4]Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China

## Abstract

*Text detection and recognition in natural images have long been considered as two separate tasks that are processed sequentially. Jointly training two tasks is non-trivial due to significant differences in learning difficulties and convergence rates. In this work, we present a conceptually simple yet efficient framework that simultaneously processes the two tasks in a united framework. Our main contributions are three-fold: (1) we propose a novel* text-alignment *layer that allows it to precisely compute convolutional features of a text instance in arbitrary orientation, which is the key to boost the performance; (2) a character attention mechanism is introduced by using character spatial information as explicit supervision, leading to large improvements in recognition; (3) two technologies, together with a new RNN branch for word recognition, are integrated seamlessly into a single model which is end-to-end trainable. This allows the two tasks to work collaboratively by sharing convolutional features, which is critical to identify challenging text instances. Our model obtains impressive results in end-to-end recognition on the ICDAR 2015 [19], significantly advancing the most recent results [2], with improvements of F-measure from (0.54, 0.51, 0.47) to (0.82, 0.77, 0.63), by using a strong, weak and generic lexicon respectively. Thanks to joint training, our method can also serve as a good detector by achieving a new state-of-the-art detection performance on related benchmarks. Code is available at* `https://github.com/tonghe90/textspotter`.


Figure 1: Illustrations of the results on ICDAR 2015 by our proposed method, which can detect all possible text regions and recognize relevant transcriptions in a unified framework.

## 1. Introduction

The goal of text spotting is to map an input natural image into a set of character sequences or word transcripts and corresponding locations. It has attracted increasing attention in the vision community, due to its numerous potential applications. It has made rapid progress riding on the wave of recent deep learning technologies, as substantiated by recent works [17, 8, 2, 23, 12, 35, 43, 31, 33, 22, 26]. However, text spotting in the wild still remains an open problem, since text instances often exhibit vast diversity in font, scale and orientation with various illumination effects, which often come with a highly complicated background.

Past works in text spotting often consider it as two separate tasks: text detection and word recognition, which are implemented sequentially. The goal of text detection is to precisely localize all text instances (e.g., words) in a natural image, and then a recognition model is processed repeatedly through all detected regions for recognizing corresponding text transcripts. Recent approaches for text detection are mainly extended from general object detectors (such as Faster R-CNN [29] and SSD [25]) by directly regressing a bounding box for each text instance, or from semantic segmentation methods (e.g., Fully Convolutional Networks (FCN) [27]) by predicting a text/non-text probability at each pixel. With careful model design and development, these approaches can be customized properly towards this highly

The first two authors contribute equally. C. Shen is the corresponding author (e-mail: chunhua.shen@adelaide.edu.au).

domain-specific task, and achieve the state-of-the-art performance [8, 12, 35, 43, 31, 42]. The word recognition can be cast into a sequence labeling problem where convolutional recurrent models have been developed recently [31, 9]. Some of them were further incorporated with an attention mechanism for improving the performance [21, 1]. However, training two tasks separately does not exploit the full potential of convolutional networks, where the convolutional features are not shared. It is natural for us to make a more reliable decision if we clearly understand or recognize the meaning of a word and all characters within it. Besides, it is also possible to introduce a number of heuristic rules and hyper-parameters that are costly to tune, making the whole system highly complicated.

Recent Mask R-CNN [7] incorporates an instance segmentation task into the Faster R-CNN [29] detection framework, resulting in a multi-task learning model that jointly predicts a bounding box and a segmentation mask for each object instance. Our work draws inspiration from this pipeline, but has a different goal of learning a direct mapping between an input image and a set of character sequences. We create a recurrent sequence modeling branch for word recognition within a text detection framework, where the RNN based word recognition is processed in parallel to the detection task.

However, the RNN branch, where the gradients are back-propagated through time, is clearly much more difficult to optimize than the task of bounding box regression in detection. This naturally leads to significant differences in learning difficulties and convergence rates between two tasks, making the model particularly hard to be trained jointly. For example, the magnitude of images for training a text detection model is about $10^3$ (e.g., 1000 training images in the ICDAR 2015 [19]) , but the number is increased significantly by many orders of magnitude when an RNN based text recognition model is trained, such as the 800K synthetic images used in [6]. Furthermore, simply using a set of character sequences as direct supervision may be too abstractive (high-level) to provide meaningful detailed information for training such an integrated model effectively, which will make the model difficult to convergence. In this work, we introduce strong spatial constraints in both word and character levels, which allows the model to be optimized gradually by reducing the search space at each step.

**Contributions.** In this work, we present an end-to-end textspotter capable of learning a direct mapping between an input image and a set of character sequences or word transcripts. We propose a solution that combines a *text-alignment* layer tailed for multi-orientation text detection, together with a character attention mechanism that explicitly encodes strong spatial information of characters into the RNN branch, as shown in Figure 1. These two technologies faithfully preserve the exact spatial information in both text instance and character levels, playing a key role in boosting the overall performance. We develop a principled learning strategy that allows the two tasks to be trained collaboratively by sharing convolutional features. Our main contributions are described as follows.

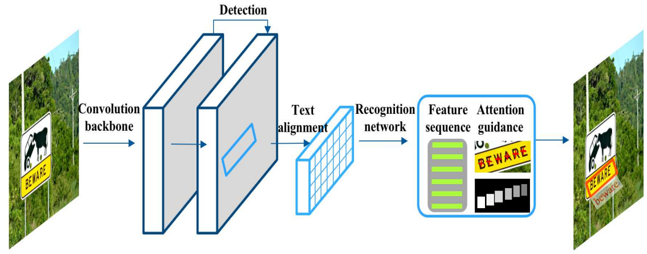


Figure 2: The framework of our method. The *text-alignment* layer is proposed to extract accurate sequence features within a detected quadrilateral of multi-orientation. A novel character attention mechanism is applied to guide the decoding process with explicit supervision. The whole framework can be trained in an end-to-end manner.

Firstly, we develop a text-alignment layer by introducing a grid sampling scheme instead of conventional RoI pooling. It computes fixed-length convolutional features that precisely align to a detected text region of arbitrary orientation, successfully avoiding the negative effects caused by orientation and quantization factor of the RoI pooling.

Secondly, we introduce a character attention mechanism by using character spatial information as an addition supervision. This explicitly encodes strong spatial attentions of characters into the model, which allows the RNN to focus on current attentional features in decoding, leading to performance boost in word recognition.

Thirdly, both approaches, together with a new RNN branch for word recognition, are integrated elegantly into a CNN detection framework, resulting in a single model that can be trained in an end-to-end manner. We develop a principled and intuitive learning strategy that allows the two tasks to be trained effectively by sharing features, with fast convergence.

Finally, we show by experiments that word recognition can significantly improve detection accuracy in our model, demonstrating strong complementary nature of them, which is unique to this highly domain-specific application. Our model achieves new state-of-the-art results on the ICDAR2015 in end-to-end recognition of multi-orientation texts, largely outperforming the most recent results in [2], with improvements of F-measure from $(0.54, 0.51, 0.47)$ to $(0.82, 0.77, 0.63)$ in terms of using a strong, weak and generic lexicon.

## 2. Related work

Here we briefly introduce some related works on text detection, recognition and recent method for end-to-end wordspotting.

**Scene text detection.** Recent approaches cast previous character-based text detection [14, 15, 11, 37] into direct text region estimation [24, 43, 42, 10, 8, 41, 38], which avoids multiple bottom-up post-processing steps by taking word or text-line as a whole detection unite. Built on advances of recent object detector Faster-RCNN [29], Tian *et al*. [35] proposed a novel Connectionist Text Proposal Network (CTPN) which detects a text line in a sequence of fine-scale text proposals from convolutional layers, with a new RNN design incorporated. Liao *et al*. [24] extended single-shot object detector (SSD) [25] to text detection. He *et al*. [8] proposed a text attention module capable of automatically learning the attention of rough text regions from the convolutional features. This allows the model to produce single-shot detection of text with a high accuracy. In [12], the author proposed a method to generate arbitrary quadrilaterals, by calculating offsets between every point of text region and vertex coordinates. A weakly supervised text detector, WeText, was proposed to learn from un-annotated or weakly annotated data [33]. Zhou *et al*. applied Intersection-over-Union (IoU) loss to text detection, which regresses text bounding boxes densely at each spatial location.

**Scene text recognition.** With the success of recurrent neural networks on handwriting recognition and speech translation, sequence modelling has recently been applied to scene text recognition. For example, He *et al*. [9] cast the task of word recognition into a sequence labelling problem, where an encoding-decoding process is introduced by incorporating LSTM [3] and connectionist temporal classification (CTC) [4] into a unified framework. Similar work has been developed by Shi *et al*. in [31], and spatial transformer networks [17] was incorporated for automatic rectification [32]. Lee *et al*. [21] proposed an attention-based LSTM for word recognition. However, these attention weights are learned completely from the distribution of data, without any clear supervision that guides the learning process.

**End-to-end wordspotting.** End-to-end wordspotting is an emerging research area. Previous methods usually try to solve it by splitting the whole process into two independent problems: training two cascade models, one for detection and one for recognition. Detected text regions are firstly cropped from original image, followed by affine transformation and rescaling. Corrected images are repeatedly processed by a recognition model to obtain corresponding transcripts. However, training errors will be accumulated due to cascading models without sharable features. Li *et al*. [23] proposed a unified network that simultaneously localizes and recognizes text in one forward pass by sharing convolution features under a curriculum strategy. But the existing RoI pooling operation limits it to detect and recognize only horizontal examples. Busta *et al*. [2] brought up deep text spotter, which can solve wordspotting of multi-orientation problem. However, the method does not have sharable feature, meaning that the recognition loss of the later stage has no influence on the former localization results.

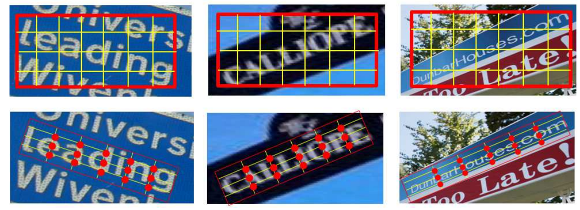

Figure 3: Standard RoI pooling (Top) and *text-alignment* layer (Bottom). Our method can avoid encoding irrelevant texts and complicated background, which is crucial for the accuracy of text recognition.

## 3. End-to-End TextSpotter

In this section, we present the details of the proposed textspotter which learns a direct mapping between an input image and a set of word transcripts with corresponding bounding boxes with arbitrary orientations. Our model is a fully convolutional architecture built on the PVAnet framework [13]. As shown in Figure 2, we introduce a new recurrent branch for word recognition, which is integrated into our CNN model in parallel with the existing detection branch for text bounding box regression. The RNN branch is composed of a new *text-alignment* layer and a LSTM-based recurrent module with a novel character attention embedding mechanism. The *text-alignment* layer extracts precise sequence feature within the detected region, preventing encoding irrelevant texts or background information. The character attention embedding mechanism regulates the decoding process by providing more detailed supervisions of characters. Our textspotter directly outputs final results in an end-to-end manner, without any post-processing step except for a simple non-maximum suppression (NMS).

**Network architecture** Our model is a fully convolutional architecture inspired by [43], where a PVA network [13] is utilized as backbone due to its significantly low computational cost. Unlike generic objects, texts often have a much larger variations in both sizes and aspect ratios. Thus it not only needs to preserve local details for small-scale text instances, but also should maintain a large receptive field for very long instances. Inspired by the success in semantic segmentation [30], we exploit feature fusion by combining convolutional features of conv5, conv4, conv3 and conv2 layers gradually, with the goal of maintaining both local detailed features and high-level context information. This results in more reliable predictions on multi-scale text instances. The size of the top layer is $\frac{1}{4}$ of the input image.

**Text detection** This branch is similar to that of [43], where a multi-task prediction is implemented at each spa-

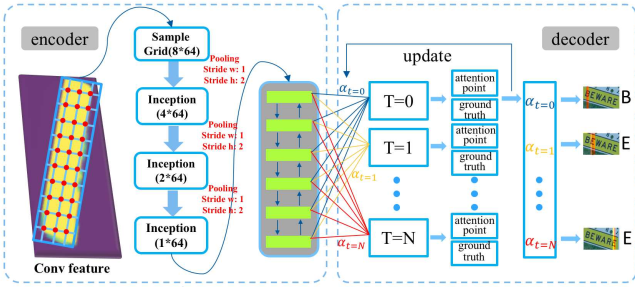


Figure 4: Our proposed sub-net structure for recognition branch, which provides guidance during the decoding process by using character spatial information as supervision.

tial location on the top convolutional maps, by adopting an Intersection over Union (IoU) loss described in [40]. It contains two sub-branches on the top convolutional layer designed for joint text/non-text classification and multi-orientation bounding boxes regression. The first sub-branch returns a classification map with an equal spatial size of the top feature maps, indicating the predicted text/non-text probabilities using a softmax function. The second sub-branch outputs five localization maps with the same spatial size, which estimate five parameters for each bounding box with arbitrary orientation at each spatial location of text regions. The five parameters represent the distances of the current point to the top, bottom, left and right sides of an associated bounding box, together with its inclined orientation. With these configurations, the detection branch is able to predict a quadrilateral of arbitrary orientation for each text instance. The feature of the detected quadrilateral region is then feed into the RNN branch for word recognition via a *text-alignment* layer which is described below.

### 3.1. Text-Alignment Layer

We create a new recurrent branch for word recognition, where a text-alignment layer is proposed to precisely compute fixed-size convolutional features from a quadrilateral region of arbitrary size. The text-alignment layer is extended from RoI pooling [5] which is widely used for general objects detection. The RoI pooling computes a fixed-size convolutional features (e.g., $7 \times 7$) from a rectangle region of arbitrary size by performing quantization operation. It can be integrated into the convolutional layers for in-network region cropping, which is a key component for training an end-to-end framework. However, directly applying the RoI pooling to a text region will lead to a significant performance drop in word recognition due to the issue of misalignment. Reasons are described below.

Firstly, unlike object detection and classification where the RoI pooling computes global features of a RoI region for discriminating an object, word recognition requires more detailed and accurate local features and spatial information for predicting each character sequentially. As pointed out in [7], the RoI pooling performs quantizations which inevitably introduce misalignments between the original RoI region and the extracted features. Such misalignments have a significant negative effect on predicting characters, particularly on some small-scale ones such as 'i', 'l'.

Secondly, RoI pooling was designed for a rectangle region which is only capable of localizing horizontal instances. It will make larger misalignments when applied to multi-orientation text instances. Furthermore, a large amount of background information and irrelevant texts are easily encoded when a rectangle RoI region is applied to a highly inclined text instance, as shown in Figure 3. This severely reduces the performance on the RNN decoding process for recognizing sequential characters.

Recent Mask R-CNN considers explicit per-pixel spatial correspondence by introducing RoIAlign pooling [7]. This inspires current work that develops a new text-alignment layer tailored for text instance which is a quadrilateral shape with arbitrary orientation. It provides strong word-level alignment with accurate per-pixel correspondence, which is of critical importance to extract exact text information from the convolutional maps, as shown in Figure 3.

Specifically, given a quadrilateral region, we first build a sampling grid with size of $h \times w$ on the top convolutional maps. The sampled points are generated with equidistant interval within the region, and the feature vector ($\mathbf{v}_p$) for a sampled point ($p$) at spatial location $(p_x, p_y)$, is calculated via a bilinear sampling [7] as follows,

$$\mathbf{v}_p = \sum_{i=1}^{4} \mathbf{v}_{pi} * g(p_x, p_{ix}) * g(p_y, p_{iy}) \tag{1}$$

where $\mathbf{v}_{pi}$ refers to four surrounding points of point $p$, $g(m, n)$ is the bilinear interpolation function and $p_{ix}$ and $p_{iy}$ refer to the coordinates of point $p_i$. As presented in [7], an appealing property of the bilinear sampling is that gradients of the sampled points can be back-propagated through the networks, by using Eq. 2.

$$\frac{\partial grad}{\partial v_{pi}} = \sum g(p_x, p_{ix}) * g(p_y, p_{iy}) \tag{2}$$

Grid sampling, by generating a fixed number of sampling points (e.g., $w = 64$, $h = 8$ in our experiments), provides an efficient way to compute fixed-size features from a quadrilateral region with arbitrary size and orientation. The bilinear sampling allows for exacting per-pixel alignment, successfully avoiding the quantization factor.

### 3.2. Word Recognition with Character Attention

The word recognition module is built on the text-alignment layer, as shown in Figure 2. Details of this module is presented in Figure 4, where the input is fixed-size convolutional features output from the text-align pooling layer with size of $w \times h \times C$. The convolutional features

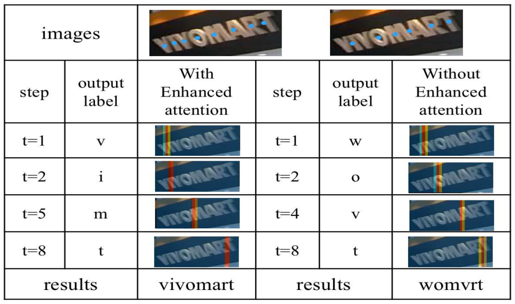

Figure 5: A comparison of the proposed method with traditional attention LSTM. The heat map indicates the focusing location at each time step.

are fed into multiple inception modules and generate a sequence of feature vectors, e.g., 64 $\times$C-dimensional features, as shown in Figure 4. In the next part, we will briefly introduce the attention mechanism and three strategies to enhance attention alignment.

### 3.2.1 Attention Mechanism

Recently, attention mechanism has been developed for word recognition [21, 1], where an implicit attention is learned automatically to enhance deep features in the decoding process. In the encoding process, a bi-direction LSTM layer is utilized to encode the sequential vectors. It outputs hidden states $\{h^e_1, h^e_2, ..., h^e_w\}$ of the same number, which encode strong sequential context features from both past and future information. Unlike previous work [31, 9] which decode a character (including a non-character label) using each hidden state, the attention mechanism introduces a new decoding process where an attention weights ($\alpha_t \in \mathbb{R}^w$) is learned automatically at each decoding iteration, and the decoder predicts a character label ($y_t$) by using this attention vector,

$$y_t = Decoder(h^d_t, g_t, y_{t-1}) \tag{3}$$

where $h^d_t$ is the hidden state vector of the decoder at time $t$, computed by:

$$h^d_t = f(y_{t-1}, h^d_{t-1}, g_t) \tag{4}$$

$g_t$ is the context vector, which is calculated as a weighted sum of the input sequence: $g_t = \sum_{j=1}^{w} \alpha_{t,j} h^e_j$. The decoder is ended until it encounters an end-of-sequence ($EOS$). The attention vector is calculated by $\alpha_{t,j} = softmax(e_{t,j})$, where $e_{t,j} = z(h^d_{t-1}, h^e_j)$ is an alignment factor measuring matching similarity between the hidden state and encoding features $h^e_j$. However, these attention vectors are learned automatically in the training process without an explicit guidance, giving rise to misalignment problem which severely reduces recognition performance, as shown in Figure 5. To address this problem, we propose new attention alignment and enhancement methods that explicitly encode strong attention of each character.

### 3.2.2 Attention Alignment and Enhancement

We introduce a new method which enhance the attention of characters in word recognition. We develop character-alignment mechanism that explicitly encodes strong character information, together with a mask supervision task which provides meaningful local details and spatial information of character for model learning. Besides, an attention position embedding is also presented. It identifies the most significant spot from the input sequence which further enhances the corresponding text features in inference. These technical improvements are integrated seamlessly into a unified framework that is end-to-end trainable. Details of each module are described as follows.

**Attention alignment** To deal with misalignment issue raised by existing implicit attention models, we propose an attention alignment which explicitly encodes spatial information of characters, by introducing an additional loss as supervision.

Specifically, assuming that $p_{t,1}, p_{t,2}, ..., p_{t,w}$ are central points in each column of the sampling grid. At $t$-th time step, these central points can be calculated by Eq. 5,

$$\delta_t = \sum_{j=1}^{w} \alpha_{t,j} \times p_{t,j} \tag{5}$$

Ideally, $\delta_t$ should be close to the center of the current character, $y_t$. Without supervision, it is likely to result in misalignment and therefore incorrect sequence labels. Intuitively, we can construct a loss function (Eq. 6) to describe whether the attention points is focusing on the right place.

$$\ell_{align} = \sum_{t=0}^{T} \left\| \frac{\delta_t - k_t}{0.5 * \bar{w}_t} \right\|^2 \tag{6}$$

where $k_t$ is ground truth (GT) coordinates, and $\bar{w}_t$ is the GT width of current character, $y_t$. Both of them are projected onto the axis of text orientation. $T$ is the number of characters in a sequence. Notice that the distance between the prediction and GT should be normalized by character width, which we found is useful for model convergence.

**Character mask** To further enhance character attention, we introduce another additional supervision by leveraging character mask, which provides more meaningful information, including both local details and spatial location of a character. A set of binary masks are generated, with the same spatial size of the last convolutional maps. The number of the masks is equal to the number of character labels. A softmax loss function is applied at each spatial location, which is referred as mask loss $\ell_{mask}$. This explicitly encoding strong detailed information of characters into the attention module. Both $\ell_{mask}$ and $\ell_{align}$ losses are optional during the training process, and can be ignored on those images where character level annotations are not provided.

**Position embedding** Position embedding was first introduced in [36], aiming to make the model 'location aware' by encoding a one-hot coordinate vector. This is equivalent to adding a varying bias terms. It is difficult to directly apply it to our task, as the size of the feature maps changes according to the size of input image. Instead, we generate a one-hot vector from the attention vector, $u_k = \arg\min_j \alpha_{t,j}$, which is a fixed-size binary vector (e.g., 64-D). Then, we directly concatenate the one-hot vector with the context vector ($g_t$), which forms a new feature representation with additional one-hot attention information. Then the decoder computed in Eq. 3 can be modified as,

$$y_t = Decoder(h_t^d, g_t, y_{t-1}, u_t) \tag{7}$$

Finally, by integrating all these modules into an end-to-end model, we obtains an overall loss function including four components,

$$L = \ell_{loc} + \ell_{word} + \lambda_1 \ell_{align} + \lambda_2 \ell_{mask} \tag{8}$$

where $\ell_{word}$ is a softmax loss for word recognition, $\ell_{loc}$ is the loss function for text instance detection, and $\lambda_1$ and $\lambda_2$ are corresponding loss weights (both are set to 0.1).

### 3.3. Training Strategy

Training our model in an end-to-end manner is challenging due to a number of difficulties. First, largely different nature of them, e.g., significant differences in learning difficulties and convergence rates. Second, the extremely unbalanced distribution of image data. Our methods require character-level bounding boxes for generating character coordinates and masks. These detailed character annotations are not provided in the standard benchmarks, such as the ICDAR2013 [20] and ICDAR2015 [19]. Although Gupta *et al.* [6] developed a fast and scalable engine to generate synthetic images of text, providing both word-level and character-level informations, there is still a large gap between realistic and synthesized images, making the trained model difficult to generalize well to real-world images.

We fill this gap by developing a principled training strategy which includes multiple steps. It is able to train multiple tasks collaboratively in our single model, allowing for excellent generalization capability from the synthesized images to real-world data.

**Step One**: We randomly select 600k images from the 800k synthetic images. Word recognition task is firstly trained by fixing the detection branch. We provide the ground truth bounding boxes of word instances to the text-align layer. Three losses: $\ell_{word}$, $\ell_{align}$ and $\ell_{mask}$ are computed. The training process takes 120k iterations with a learning rate $2 \times 10^{-3}$.

**Step Two**: For the next 80k iterations, we open the detection branch, but still use the GT bounding boxes for the text-align layer, as the detector performs poorly at first, which will be harmful to the already trained recognition branch. The learning rate is set to $2 \times 10^{-4}$. During the next 20k iterations, sampling grid is generated from detection results. The model is trained end-to-end in this stage.

**Step Three**: About 3,000 real-world images from the ICDAR 2013 [20], ICDAR 2015 [19] and Multi-lingual[1] datasets are utilized in the next 60k iterations. To enhance the generalization ability, data augmentation is employed. We re-scale the images by keeping the aspect ratio unchanged, followed by random rotation ranging from $-20°$ to $20°$, and random cropping $800 \times 800$ patches for training. To utilize the character-level supervision, we set the batch size to 4, where an image from the synthetic dataset is included. The learning rate remained at $2 \times 10^{-4}$. The whole system is implemented by using Caffe [18].

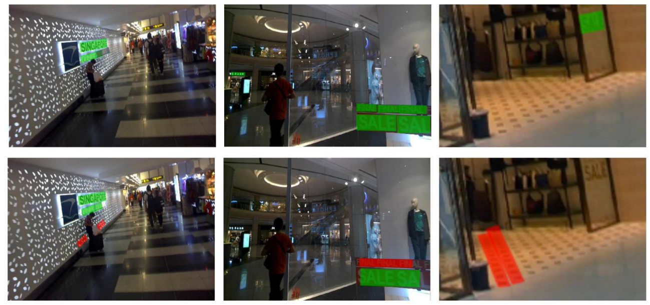

Figure 6: A comparison of detection performance between joint training (Top) and separate training (Bottom). Joint training makes it more robust to find out text regions as two tasks are highly correlated, where detection can benefit from training of recognition.

| roi pooling? | roi alignment? | text alignment? | supervision? | position embedding? | Accuracy (%) |
|---|---|---|---|---|---|
| ✓ | × | × | × | × | 60.7 |
| × | ✓ | × | × | × | 61.9 |
| × | × | ✓ | × | × | 67.6 |
| × | × | ✓ | ✓ | × | 68.8 |
| × | × | ✓ | × | ✓ | 68.2 |
| × | × | ✓ | ✓ | ✓ | **69.5** |

Table 1: Ablations for the proposed method. We test our model on ICDAR2015. The detection part is replaced with ground truth for fair comparison.

## 4. Experiments

In this section, we first briefly introduce the datasets we use and the evaluation protocols, followed by thorough comparison of the proposed method with the state-of-the-art along with comprehensive ablation experiments.

**Datasets** The ICDAR2013 dataset focuses more on horizontal text instances, which contains 229 images for training and 233 images for testing with word-level annotation.

The ICDAR2015 dataset is collected by Google glasses, which has 1,000 images for training and 500 images for

[1] http://rrc.cvc.uab.es/?ch=8&com=introduction

| Dataset | Method | Year | Word-Spotting | | | End-to-end | | |
|---|---|---|---|---|---|---|---|---|
| | | | Strong | Weak | Generic | Strong | Weak | Generic |
| ICDAR2013 | Deep2Text II+ [39] | 2014 | 0.85 | 0.83 | 0.79 | 0.82 | 0.79 | 0.77 |
| | Jaderberg *et al.* [16] | 2015 | 0.90 | – | 0.76 | 0.86 | – | – |
| | FCRNall+multi-filt [6] | 2016 | – | – | 0.85 | – | – | – |
| | TextBoxes [24] | 2017 | 0.94 | 0.92 | 0.86 | 0.92 | 0.90 | 0.84 |
| | YunosRobot1.0 | 2017 | 0.87 | – | 0.87 | 0.84 | – | 0.84 |
| | Li *et al.* [23] | 2017 | **0.94** | 0.92 | **0.88** | 0.91 | **0.90** | 0.85 |
| | Deep text spotter [2] | 2017 | 0.92 | 0.89 | 0.81 | 0.89 | 0.86 | 0.77 |
| | **Proposed Method** | - | 0.93 | **0.92** | 0.87 | **0.91** | 0.89 | **0.86** |
| ICDAR2015 | Method | Year | Word-Spotting | | | End-to-end | | |
| | | | Strong | Weak | Generic | Strong | Weak | Generic |
| | Stradvision [19] | 2013 | 0.46 | – | – | 0.44 | – | – |
| | TextSpotter [28] | 2016 | 0.37 | 0.21 | 0.16 | 0.35 | 0.20 | 0.16 |
| | Deep TextSpotter [2] | 2017 | 0.58 | 0.53 | 0.51 | 0.54 | 0.51 | 0.47 |
| | **Proposed Method** | - | **0.85** | **0.80** | **0.65** | **0.82** | **0.77** | **0.63** |

Table 2: Comparisons of the end-to-end task with state-of-the-art on ICDAR2013 and ICDAR2015. The results are reported with three different level lexicons, namely, strong, weak and generic.

testing. Different from previous datasets which are well-captured horizontal English text, it contains texts with more scales, blurring, and orientation.

The multi-lingual scene text dataset[2] is built for developing script-robust text detection methods, which contains about 9,000 images with 9 different kinds of transcriptions. We choose about 2000 of them, identified with 'Latin', to train the end-to-end task.

## 4.1. Evaluation Protocols

**Detection.** There are two standard protocols for evaluating detection results: DetEval and ICDAR2013 standard [20]. The main difference between the two protocols is that the latter one stresses more on individual words while the former one can achieve a higher score even when many words are connected into a line.

**End-to-end for Detection and Recognition.** The criterion has been used in competition: the evaluation of the results will be based on a single IoU criterion, with a threshold of 50%, and correct transcription. Besides, three dictionaries are also provided for testing reference, i.e., 'strong', 'weak' and 'generic'. 'Strong' lexicon has 100 entries for every image, and most words appeared in that image are included. 'Weak' lexicon contains all the words that appeared in the testing dataset. 'Generic' lexicon has 90K words. One thing should be noticed that the length of all the words in dictionaries are greater than 3 with symbols and numbers excluded. There are two protocols for evaluation: end-to-end and word-spotting. End-to-end needs to recognize all the words precisely, no matter whether the dictionary contains these strings. On the other hand, word-spotting only examines whether the words in the dictionary appear in images, making it less strict than end-to-end for ignoring symbols, numbers and words whose length is less than 3.

[2] http://rrc.cvc.uab.es/?ch=8&com=introduction


Figure 7: Examples of textspotting results of the proposed method on ICDAR2013 and ICDAR2015.

## 4.2. Text-alignment vs. RoI Pooling

We compare the proposed *text-alignment* with standard RoI pooling. To make fair comparison, the detection part is fixed with ground truth and recognition performance is evaluated on ICDAR2015. Due to encoding background information and irrelevant text instances, RoI pooling results in mis-alignment and inaccurate representation of feature sequences. As shown in Table 1, the accuracy of recognition

| ICDAR2013 dataset | | | | | | | | ICDAR2015 dataset | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Method | Year | ICDAR standard | | | DetEval | | | Method | Year | R | P | F |
| | | R | P | F | R | P | F | | | | | |
| TextFlow [34] | 2015 | 0.76 | 0.85 | 0.80 | - | - | - | StradVision2 | 2015 | 0.37 | 0.77 | 0.50 |
| Text-CNN [11] | 2016 | 0.73 | 0.93 | 0.82 | 0.76 | 0.93 | 0.84 | MCLAB_FCN [42] | 2016 | 0.43 | 0.71 | 0.54 |
| FCRN [6] | 2016 | 0.76 | **0.94** | 0.84 | 0.76 | 0.92 | 0.83 | EAST [43] | 2016 | 0.78 | 0.83 | 0.81 |
| CTPN [35] | 2016 | 0.73 | 0.93 | 0.82 | 0.83 | **0.93** | 0.88 | CTPN [35] | 2016 | 0.52 | 0.74 | 0.61 |
| He *et al.* [8] | 2017 | 0.86 | 0.88 | 0.87 | 0.86 | 0.89 | 0.88 | He *et al.* [8] | 2017 | 0.73 | 0.80 | 0.77 |
| He *et al.* [12] | 2017 | 0.81 | 0.92 | 0.86 | - | - | - | He *et al.* [12] | 2017 | 0.82 | 0.80 | 0.81 |
| Proposed wo recog | - | 0.87 | 0.88 | 0.88 | 0.87 | 0.88 | 0.88 | Proposed wo recog | - | 0.83 | 0.84 | 0.83 |
| **Proposed** | - | **0.88** | 0.91 | **0.90** | **0.89** | 0.91 | **0.90** | **Proposed** | - | **0.86** | **0.87** | **0.87** |

Table 3: Comparison of detection results with the state-of-the-art methods on ICDAR2013 and ICDAR2015. The results are reported Recall (R), Precision (P) and F-measure (F). For fair comparison, the detection performance is achieved without referring to recognition results.

with the proposed method surpasses standard RoI pooling by a large margin, boosting from 60.7% to 67.6%. All results are evaluated without referring to any lexicon in one single scale.

### 4.3. Character Attention

Different from traditional attention-based recognition models, where attention weights are automatically learned, we propose a method to regulate the learning process to prevent mis-alignment in the decoding stage. To demonstrate the effectiveness of our proposed method, we conduct two experiments with the detection part fixed. The first one is on VGG synthetic data [6], where we select 600K for training and 200K for testing. The accuracies of character-level and word-level are evaluated. The method with supervision has an accuracy of 0.95 and 0.88 on two protocols, comparing to 0.93 and 0.85 on traditional attention-based method. The other experiment is tested on ICDAR2015 dataset. As is shown in Figure 5, the proposed method gives more accurate character localization than attentional LSTM, leading to about 2% boosting in accuracy.

### 4.4. Joint Training vs. Separate Models

We believe that text detection and recognition are not two standalone problems, but highly correlated where each task can benefit from the training of the other. Joint training of two tasks in a unified framework avoids error accumulations among cascade models. As shown in Table 3, the task of recognition greatly enhances the performance of detection in terms of recall and precision, leading to a 3% improvement on F-Measure (note: the detection performances are achieved without referring to recognition results). As can be seen from Figure 6, joint training makes it more robust to text-like background and complicated text instances. We also provide a comparison with other detection approaches, indicating that our method achieved new state-of-the-art performance on ICDAR2013 and ICDAR2015 datasets.

### 4.5. Proposed Method vs. State-of-the-art Methods

End-to-end results on some extremely challenging images are presented in Figure 7. As can be seen in Figure 7, our method can correctly detect and recognize both small text instances and those with large inclined angles.

**ICDAR2015.** The effectiveness to multi-orientation texts is testified on ICDAR2015 dataset. Our method achieved an F-measure of 0.82, 0.77 and 0.63 respectively in terms of referencing 'Strong', 'Weak' and 'Generic' lexicon under the end-to-end protocol, which surpasses the state-of-the-art performance of 0.54, 0.77 and 0.63 by a large margin.

**ICDAR2013.** The dataset is well-captured for horizontal text instances. The result is shown in Table 2, which is comparable to the state-of-the-art result [23].

## 5. Conclusion

In this paper we have presented a novel framework that combines detection and recognition in a unified network with sharable features.

We have proposed a novel text-alignment layer that can extract precise sequence information without encoding irrelevant background or texts. We also improve the accuracy of traditional LSTM by enhancing the attention of characters during the decoding process. Our proposed method achieves state-of-the-art performance on two open benchmarks: ICDAR2013 and ICDAR2015 and outperforms previous best methods by a large margin.

## 6. Acknowledgments

C. Shen's participation was in part supported by an ARC Future Fellowship. Y. Qiao's participation was in part supported by Industrial Collaboration Project (Y5Z0371001), National Natural Science Foundation of China (U1613211, 61633021) and External Cooperation Program of BIC Chinese Academy of Sciences (172644KYSB20150019, 172644KYSB20160033).

## References


[1] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2016. 2, 5

[2] M. Busta, 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. 1, 2, 3, 7

[3] A. Graves and N. Jaitly. Towards end-to-end speech recognition with recurrent neural networks. In *Proc. Int. Conf. Mach. Learn.*, 2014. 3

[4] A. Graves, M. Liwicki, S. Fernandez, R. Bertolami, H. Bunke, and J. Schmidhuber. A novel connectionist system for unconstrained handwriting recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31:855–868, 2009. 3

[5] R. Grishick. Fast R-CNN. In *Proc. IEEE Int. Conf. Comp. Vis.*, 2015. 4

[6] A. Gupta, A. Vedaldi, and A. Zisserman. Synthetic data for text localisation in natural images. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2016. 2, 6, 7, 8

[7] K. He, G. Gkioxari, P. Dollar, and R. Grishick. Mask R-CNN. In *Proc. IEEE Int. Conf. Comp. Vis.*, 2017. 2, 4

[8] 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. Comp. Vis.*, 2017. 1, 2, 3, 8

[9] 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. 2, 3, 5

[10] T. He, W. Huang, Y. Qiao, and J. Yao. Accurate text localization in natural image with cascaded convolutional text network. *arXiv preprint arXiv:1603.09423*, 2016. 3

[11] T. He, W. Huang, Y. Qiao, and J. Yao. Text-attentional convolutional neural networks for scene text detection. *IEEE Trans. Image Process.*, 25:2529–2541, 2016. 3, 8

[12] W. He, X. Zhang, F. Yin, and C. Liu. Deep direct regression for multi-oriented scene text detection. *arXiv preprint arXiv:1703.08289*, 2017. 1, 2, 3, 8

[13] S. Hong, B. Roh, K. Kim, Y. Cheon, and M. Park. PVANet: Lightweight deep neural networks for real-time object detection. *arXiv preprint arXiv:1611.08588*, 2016. 3

[14] W. Huang, Z. Lin, J. Yang, and J. Wang. Text localization in natural images using stroke feature transform and text covariance descriptors. In *Proc. IEEE Int. Conf. Comp. Vis.*, 2013. 3

[15] W. Huang, Y. Qiao, and X. Tang. Robust scene text detection with convolutional neural networks induced MSER trees. In *Proc. Eur. Conf. Comp. Vis.*, 2014. 3

[16] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman. Reading text in the wild with convolutional neural networks. *Int. J. Comput. Vision*, 116:1–20, 2016. 7

[17] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu. Spatial transformer networks. In *Proc. Advances in Neural Inf. Process. Syst.*, 2015. 1, 3

[18] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *ACM Conf. Multimedia*, 2014. 6

[19] 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 competition on robust reading. In *Proc. Int. Conf. Document Analysis & Recognition*, 2015. 1, 2, 6, 7

[20] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L. Gomez, S. Robles, J. Mas, D. Fernandez, J. Almazan, and L. de las Heras. ICDAR 2013 robust reading competition. In *Proc. Int. Conf. Document Analysis and Recognition*, 2013. 6, 7

[21] C. Lee and S. Osindero. Recursive recurrent nets with attention modeling for OCR in the wild. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2016. 2, 3, 5

[22] H. Li, P. Wang, and C. Shen. Towards end-to-end car license plates detection and recognition with deep neural networks. *arXiv preprint arXiv:1709.08828*, 2017. 1

[23] 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. 1, 3, 7, 8

[24] 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. 3, 7

[25] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu, and A. C. Berg. SSD: Single shot multibox detector. In *Proc. Eur. Conf. Comp. Vis.*, 2016. 1, 3

[26] X. Liu, D. Liang, S. Yan, D. Chen, Y. Qiao, and J. Yan. Fots: Fast oriented text spotting with a unified network. *arXiv preprint arXiv:1801.01671*, 2018. 1

[27] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2015. 1

[28] L. Neumann and J. Matas. Real-time lexicon-free scene text localization and recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 38:1872–1885, 2016. 7

[29] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region. In *Proc. Advances in Neural Inf. Process. Syst.*, 2015. 1, 2, 3

[30] O. Ronneberger, P. Fischer, and T. Brox. U-Net: Convolutional networks for biomedical image segmentation. In *Proc. Int. Conf. Medical Image Computing & Computer-Assisted Intervention*, 2015. 3

[31] 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.*, 39:2298–2304, 2017. 1, 2, 3, 5

[32] B. Shi, X. Wang, P. Lyu, C. Yao, and X. Bai. Robust scene text recognition with automatic rectification. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2016. 3

[33] S. Tian, S. Lu, and C. Li. Wetext: Scene text detection under weak supervision. In *Proc. IEEE Int. Conf. Comp. Vis.*, 2017. 1, 3

[34] S. Tian, Y. Pan, C. Huang, S. Lu, K. Yu, and C. L. Tan. Text flow: A unified text detection system in natural scene images. In *Proc. IEEE Int. Conf. Comp. Vis.*, 2015. 8

[35] 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. 1, 2, 3, 8

[36] Z. Wojna, A. Gorban, D. Lee, K. Murphy, Q. Yu, Y. Li, and J. Ibarz. Attention-based extraction of structured information

from street view imagery. *arXiv preprint arXiv:1704.03549*, 2017. 6

[37] C. Yao, X. Bai, W. Liu, Y. Ma, and Z. Tu. Detecting texts of arbitrary orientations in natural images. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2012. 3

[38] C. Yao, X. Bai, N. Sang, X. Zhou, S. Zhou, and Z. Cao. Scene text detection via holistic, multi-channel prediction. *arXiv preprint arXiv:1606.09002*, 2016. 3

[39] X. Yin, X. Yin, K. Huang, and H. Hao. Robust text detection in natural scene images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36:970–983, 2014. 7

[40] J. Yu, Y. Jiang, Z. Wang, Z. Cao, and T. Huang. UnitBox: An advanced object detection network. In *ACM Conf. Multimedia*, 2016. 4

[41] 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. 3

[42] 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. 2, 3, 8

[43] 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. 1, 2, 3, 8