MANGO: A Mask Attention Guided One-Stage Scene Text Spotter

Liang Qiao[1*], Ying Chen[2*], Zhanzhan Cheng[31†], Xunlu Xu[1], Yi Niu[1], Shiliang Pu[1‡], Fei Wu[3]
[1]Hikvision Research Institute, China; [2]Tongji University, China; [3]Zhejiang University, China
{qiaoliang6, chengzhanzhan, xuyunlu, niuyi, pushiliang}@hikvision.com 1833019@tongji.edu.cn wufei@cs.zju.edu.cn

Abstract

Recently end-to-end scene text spotting has become a popular research topic due to its advantages of global optimization and high maintainability in real applications. Most methods attempt to develop various region of interest (RoI) operations to concatenate the detection part and the sequence recognition part into a two-stage text spotting framework. However, in such framework, the recognition part is highly sensitive to the detected results (*e.g.*, the compactness of text contours). To address this problem, in this paper, we propose a novel Mask AttentioN Guided One-stage text spotting framework named MANGO, in which character sequences can be directly recognized without RoI operation. Concretely, a position-aware mask attention module is developed to generate attention weights on each text instance and its characters. It allows different text instances in an image to be allocated on different feature map channels which are further grouped as a batch of instance features. Finally, a lightweight sequence decoder is applied to generate the character sequences. It is worth noting that MANGO inherently adapts to arbitrary-shaped text spotting and can be trained end-to-end with only coarse position information (*e.g.*, rectangular bounding box) and text annotations. Experimental results show that the proposed method achieves competitive and even new state-of-the-art performance on both regular and irregular text spotting benchmarks, i.e., ICDAR 2013, ICDAR 2015, Total-Text, and SCUT-CTW1500.

1 Introduction

Scene text spotting has attracted much attention due to its various practical applications such as key entities recognition in invoice/receipt understanding, product name identification in the e-commerce system, and license plate recognition in the intelligent transportation system. Traditional scene text spotting systems are usually in three steps: localizing text regions, cropping text regions from the original image, and recognizing them as character sequences (Wang et al. 2012; Jaderberg, Vedaldi, and Zisserman 2014; Neumann and Matas 2015; i Bigorda and Karatzas 2017). While such text spotting model brings many considerable problems, such as (1) errors will be accumulated among the multiple individual tasks, (2) it is costly to maintain multiple separate models, and (3) the model is hard to adapt to various applications.

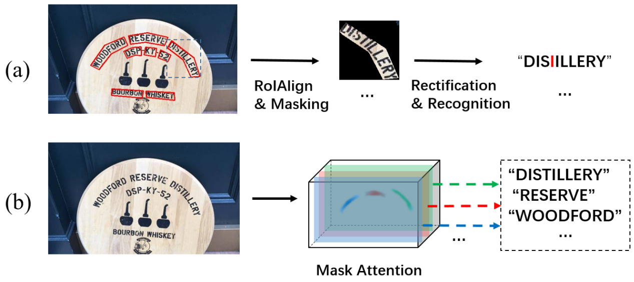

Figure 1: Illustration of the traditional two-stage text spotting process and the proposed MANGO. Sub-figure (a) shows the two-stage text spotting strategy by using RoI operations to connect the detection and recognition parts. Sub-figure (b) is the proposed one-stage text spotting approach, which can directly output the final character sequences.

Therefore, many works (Li, Wang, and Shen 2017; Bartz, Yang, and Meinel 2018; He et al. 2018; Sun et al. 2018; Liao et al. 2019) are proposed to optimize the text spotting process in an end-to-end manner. These methods usually use various Region of Interest (RoI) operations to bridge the text detection and recognition parts in a differentiable way, which form the *two-stage* framework. Roughly speaking, the early end-to-end methods (Li, Wang, and Shen 2017; Busta, Neumann, and Matas 2017; Liu et al. 2018; He et al. 2018) used the axis-aligned rectangular RoIs as the connecting modules. These methods are limited to cope with irregular (*e.g.*, perspective, or curved) text instances since such kind of RoIs might bring interferences from background or other texts. To solve this problem, the later methods (Feng et al. 2019; Qiao et al. 2020; Wang et al. 2020a; Qin et al. 2019; Liu et al. 2020) designed some shape-adaptive RoI mechanisms to extract the irregular text instances and rec-

[*]Authors contribute equally. Chen did this work during an internship in Hikvision Research Institute.

[†]This work is completed under the supervision of Zhanzhan Cheng (contact email: chengzhanzhan@hikvision.com).

[‡]Corresponding author

tify them into regular shapes.

In *two-stage* methods, the recognition part highly depends on the localization results, which needs the detection part must be capable of capturing accurate text boundaries to eliminate the background interference. Thus, training a robust text detection model relies on accurate detection annotations, such as polygonal or mask annotations used in irregular text spotting. Naturally, labeling such kind of annotations is laborious and costly. On the other hand, it is not easy to ensure that the tightly enclosed text regions (supervised by detection annotations) are the best form for the following recognition task. For example, in Figure 1(a), tight text boundaries may erase the edge texture of characters and lead to erroneous results. These tight detection results often need to be expanded manually for adapting to recognition well in real applications. Besides, performing complex RoI operations with non-maximum suppression (NMS) after proposals is also time-consuming, especially for arbitrary-shaped regions. Though (Xing et al. 2019) proposed a *one-stage* character-level spotting framework with its character segmentation strategy, it is difficult to extend to the situations with more character classes (*e.g.*, Chinese characters). It also loses crucial context information among characters.

In fact, when people read, they do not need to depict the accurate contours of text instances. It is enough to identify text instance via rough text position attended by visual attention. Here, we rethink the scene text spotting as an attending and reading problem, i.e., directly reading out the text contents of the coarsely attended text regions all at once.

In this paper, we propose a Mask Attention Guided One-stage text spotter called MANGO, a compact and powerful *one-stage* framework that directly predicts all texts simultaneously from an image without any RoI operation. Specifically, we introduce a position-aware mask attention (PMA) module to generate spatial attention over text regions, which contains both the instance-level mask attention (IMA) part and the character-level mask attention (CMA) part. IMA and CMA are responsible for perceiving the positions of text and characters in an image, respectively. Text instances' features can be directly extracted by the position-aware attention maps rather than explicit cropping operation, which reserves the global spatial information as much as possible. Here, different text instances' features will be mapped into different feature map channels using dynamic convolutions (Wang et al. 2020c), as shown in Figure 1(b). After that, a lightweight sequence decoder is applied to generate character sequences in a batch all at once.

Note that MANGO can be end-to-end optimized with only rough position information (*e.g.*, a rectangular bounding box, or even the center point of the text instance) as well as sequence annotations. Benefiting from PMA, this framework can adaptively spot various irregular text without any rectification mechanism, and is also capable of learning the reading order for arbitrary-shaped text.

The major contributions of this paper are as follows: (1) We propose a compact and robust *one-stage* text spotting framework named MANGO that can be trained in an end-to-end manner. (2) We develop the position-aware mask attention module to generate the text instance features into a batch, and build the one-to-one mapping with final character sequences. The module can be trained with only rough text position information and text annotations. (3) Extensive experiments show that our method achieves competitive and even state-of-the-art results on both regular and irregular text benchmarks.

2 Related Works

We divide existing scene text spotting methods into the following two categories.

2.1 Two-stage End-to-end Scene Text Spotting

Early scene text spotting methods (Liao, Shi, and Bai 2018; Liao et al. 2017; Wang et al. 2012) usually first localize each text with a trained detector such as (Liao et al. 2017; Zhou et al. 2017; He et al. 2017; Ma et al. 2018; Xu et al. 2019; Baek et al. 2019) and then recognize the cropped text region with a sequence decoder (Shi et al. 2016; Shi, Bai, and Yao 2017; Cheng et al. 2017; Zhan and Lu 2019; Luo, Jin, and Sun 2019). To sufficiently exploit the complementarity between text detection and text recognition, some works have been proposed to optimize the scene text spotting framework in an end-to-end manner, in which module connectors (*e.g.*, RoI Pooling (Ren et al. 2015a) used in (Li, Wang, and Shen 2017; Wang, Li, and Shen 2019), RoI-Align used in (He et al. 2018) and RoI-Rotate used in (Liu et al. 2018)) are developed to bridge the text detection and text recognition parts. Notice that these methods are incapable of spotting arbitrarily shaped text.

To address the irregular problems, many recent works have been proposed to design various adaptive RoI operations to spot arbitrary-shape text. (Sun et al. 2018) adopted a perspective RoI transforming module to rectify perspective text, but this strategy still has difficulty in handling heavily curved text. (Liao et al. 2019) proposed the mask textspotter inspired by the two-stage Mask-RCNN for detecting arbitrarily shaped text character-by-character, but this method loses the context information of characters and requires character-level location annotations. (Qin et al. 2019) directly adopted Mask-RCNN and an attention-based text recognizer using an RoI-Masking module to remove the background interferences before recognition. (Feng et al. 2019) treated a text instance as a group of feature pieces and adopted the RoI-Slide operation to reconstruct a straight feature map. Both (Qiao et al. 2020) and (Wang et al. 2020a) detected the key points around text and applied the thin-plate-spline transformation (Bookstein 1989) to rectify irregular instances. To obtain the smooth feature of the curved text, (Liu et al. 2020) used a Bezier curve to represent the top and bottom boundaries of text instances, and proposed a Bezier-Align operation to obtain the rectified feature maps.

The above methods achieve the end-to-end scene text spotting in a two-stage framework, in which the RoI-based connectors (*e.g.*, RoI-Align, RoI-Slide and Bezier-Align, etc.) need to be designed to explicitly crop the feature map. In two-stage frameworks, the performance highly depends on the text boundary accuracy acquired by the RoI operations. However, these complicated polygonal annotations

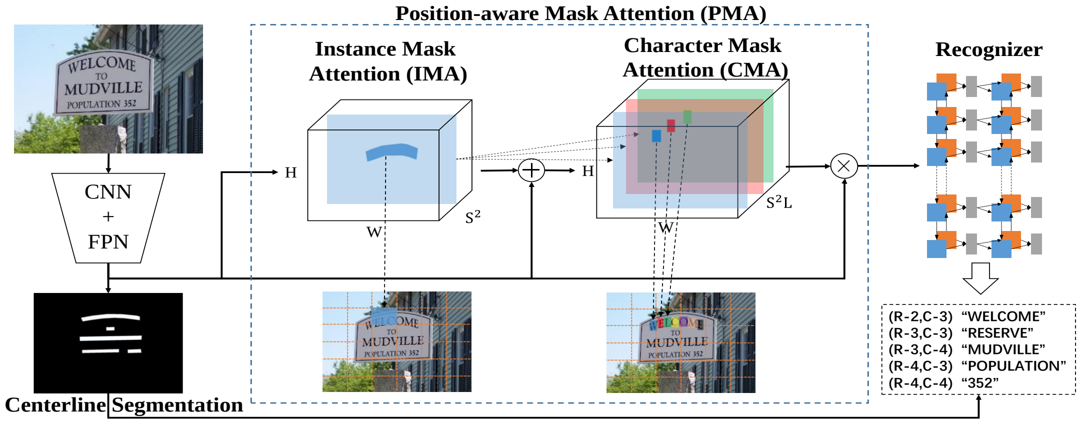

Figure 2: The workflow of MANGO. We take $S{=}6$ as an example. The input features are fed into a Position-aware Mask Attention module to map different features of instances/characters into different channels. The recognizor finally outputs character sequences in a batch all at once. A Centerline Segmentation branch is used to generate the rough positions of all text instances. Prefix 'R-' and 'C-' separately denote the grid row and column.

are usually expensive and not always suited to the recognition part, as mentioned previously.

2.2 One-stage End-to-end Scene Text Spotting

In general object localization area, many recent advances have demonstrated the efficiency and effectiveness of one-stage frameworks studied in Object Detection (Redmon et al. 2016; Liu et al. 2016; Lin et al. 2017b; Tian et al. 2019; Duan et al. 2019) or Instance Segmentation (Wang et al. 2019b; Tian, Shen, and Chen 2020; Wang et al. 2020c; Xie et al. 2020; Chen et al. 2020). However, scene text spotting is a much more challenging task, since it involves the sequence recognition problem instead of single object classification. This is because scene text has many particular traits: arbitrary shaped (e.g., curve, slant or perspective, etc.), millions of character combinations, and even unconstrained reading orders (e.g., from right to left). Recently, (Xing et al. 2019) proposed a one-stage scene text spotting approach by directly segmenting single characters. However, it loses the sequence context information among individual characters and is hard to be transferred to more character classes. To the best of our knowledge, there is no previous work to cope with the sequence-level scene text spotting task in a one-stage framework.

3 Methodology

3.1 Overview

We propose a one-stage scene text spotter named MANGO, as shown in Figure 2. The deep features are extracted through the backbone of ResNet-50 (He et al. 2016) and a feature pyramid network (FPN) (Lin et al. 2017a). The generated feature maps are then fed into three learnable modules: (1) The position-aware mask attention (PMA) module for learning spatial attention of individual text instances, which consists of the instance-level mask attention (IMA) sub-module and the character-level mask attention (CMA) sub-module. (2) The sequence decoding task for decoding the attending instance features as character sequences. (3) The global text centerline segmentation task for providing the rough text position information in the inference stage.

3.2 Position-aware Mask Attention Module

A one-stage text spotting problem can be treated as a pure text recognition task in the original image. The critical step is to build the direct one-to-one mapping between the text instances to the final character sequences in a fixed order. Here, we develop the position-aware attention (PMA) module to capture all represented text features once for the following sequence decoding module. Inspired by the grid mapping strategy used in (Wang et al. 2019b), we find that different instances can be mapped into different specific channels and achieve the instance-to-feature mapping. That is, we first divide the input image into $S{\times}S$ grids. Then the information around a grid will be mapped into the specific channel of feature maps by the proposed PMA module.

Specifically, we denote the obtained feature map after feature extraction as $x{\in}\mathbb{R}^{C\times H\times W}$, where C, H and W are channel size, width and height of the feature map, respectively. We then feed x into PMA (including IMA and CMA modules) to generate the feature representations of text instances (described bellow).

Instance-level Mask Attention IMA is responsible for generating the instance-level attention mask and assigning different instances' features into different feature map channels. It is achieved by operating a group of dynamic convolutional kernels (Wang et al. 2020c) on the sliced grids, denoted as $G^{S\times S\times C}$. The kernel size is set as $1{\times}1$.

Therefore, the instance-level attention mask can be generated by applying these kernels to the original feature map:

$$x_{ins} = G(x), \tag{1}$$

where $x_{ins} \in \mathbb{R}^{S^2 \times H \times W}$. Note that the generated feature channels are corresponding to the grid numbers.

To learn the dynamic convolutional kernels G, we need to make grid matching between the text instances and grids. Unlike general object detection or instance segmentation task, text instances usually appear in a large aspect ratio or even seriously curved. It is not reasonable to directly use the center of the text bounding box to perform grid matching. Hence, we define the term *occupation ratio* $o_{i,j}$ to represent how closely a text instance t_i matches a grid g_j:

$$o_{i,j} = \max\left(\frac{Inter(A(g_j), A(t_i))}{A(g_j)}, \frac{Inter(A(g_j), A(t_i))}{A(t_i)}\right), \tag{2}$$

where $A(.)$ is the region area and $Inter(.,.)$ is the intersection area of two regions. We say that text instance t_i *occupies* a grid g_j if $o_{i,j}$ is larger than a preset threshold μ. Then the feature channel j of x_{ins} is in charge of learning the attention mask of text t_i. In our experiments, μ is set to 0.3. Note that, in the training stage, *occupation ratio* is calculated based on the shrunk detected ground truth, such as text centerline regions.

For example in Figure 2, we set S=6. The word 'WELCOME' occupies the (row-2, col-3) and (row-2,col-4) grids. Thus, the 9-th ($(2-1)\times 6+3$) and the 10-th ($(2-1)\times 6+4$) grids will predict the same attention mask. If there are two instances *occupying* the same grid, we simply choose the one with a larger *occupation ratio*.

Character-level Mask Attention As many works (Cheng et al. 2017; Xing et al. 2019) demonstrated, the character-level position information can help to improve the recognition performance. This inspires us to design the global character-level attention submodule to provide the fine-grained feature for the subsequent recognition task.

As shown in Figure 2, CMA first concatenates the original feature map x and the instance-level attention mask x_{ins}, and then two convolutional layers (kernel size=3×3) are followed to predict the character-level attention mask:

$$x_{char} = f(x_{ins} \oplus x), \tag{3}$$

where $x_{char} \in \mathbb{R}^{(S^2 \times L) \times H \times W}$ and $\oplus$ means the channel-wise concatenation. Here, L is the predefined maximum length of character strings.

With the same grid matching strategy to IMA, if a text instance t_i occupies grid g_j at (row-h,col-w), the $((h-1)\times S \times L + (w-1)\times L + k)$ channel of x_{char} is in charge of predicting the text's k-th character mask. We again take the word 'WELCOME' as an example (See Figure 2). If $L = 25$, then the 151-st ($(2-1)\times 6 \times 25 + (3-1)\times 25 + 1$) channel predicts the attention mask of the character 'W', and the 152-nd channel predicts 'E' and so on.

3.3 Sequence Decoding Module

Since attention masks of different text instances are allocated to different feature channels, we can packet the text instance features into a batch. A simple idea is to conduct the attention fusion operation as used in (Wang et al. 2020b) to generate the batched sequential features x_{seq}, i.e.,

$$x_{seq} = x'_{char} \otimes x'^{\top}, \tag{4}$$

where $x_{seq} \in \mathbb{R}^{S^2 \times L \times C}$, $\otimes$ is the matrix multiplication operation. $x'_{char} \in \mathbb{R}^{(S^2 \times L) \times (H \times W)}$ and $x' \in \mathbb{R}^{C \times (H \times W)}$ are reshaped matrices of x_{char} and x, respectively.

Then we can transfer the text spotting problem as a pure sequence classification problem. The following sequence decoding network is responsible for generating a batch (S^2) of character sequences. Concretely, we add two layers of Bidirectional long short-term memory (BiLSTM) (Hochreiter and Schmidhuber 1997) on x_{seq} to capture the sequential relations, and finally output the character sequences by a fully connected (FC) layer.

$$x_{recog} = FC(BiLSTM(x_{seq})) \tag{5}$$

where $x_{recog} \in \mathbb{R}^{S^2 \times L \times M}$ and M is the size of character dictionary (including 26 letters, 10 digits, 32 ASCII punctuation marks and 1 EOS symbol). In specific, if the length of the predicted character string is less than L, the rest of the predictions are supplemented with the EOS symbols.

Since x_{ins} are sparse at most time, we only focus on the positive ($o_{i,j} > \mu$) samples in x_{ins} for reducing computational cost. In both training and inference stages, after the computation of Equation (1), we dynamically choose positive channels of the feature map as follows:

$$x'_{ins} = \oplus_{\substack{j \in S^2, \\ t_i \in \mathcal{T}, \\ o_{i,j} > \mu}} x_{ins}[j], \tag{6}$$

where $x'_{ins} \in \mathbb{R}^{N \times H \times W}$, $x_{ins}[j]$ denotes the j-th channel of x_{ins} and $\mathcal{T}$ is the set of text instances. N is the dynamic selected number, which equals to the number of grids that are *occupied* by texts. Then, x_{mul} in Equation (4) and x_{recog} in Equation (5) can be separately rewritten as $x_{seq} \in \mathbb{R}^{N \times L \times C}$ and $x_{recog} \in \mathbb{R}^{N \times L \times M}$.

3.4 Text Centerline Segmentation

The model is now able to output all predicted sequences for S^2 grids separately. However, if there are more than two text instances in an image, we still need to point out which grids correspond to those recognition results. Therefore, a text detection branch is required.

Since our method does not rely on the accurate boundary information, we can apply any text detection strategy (*e.g.*, RPN (Ren et al. 2015b) and YOLO (Redmon et al. 2016)) to obtain the rough geometry information of text instances. Considering that scene texts might be arbitrary-shaped, we follow most segmentation-based text detection methods (Long et al. 2018; Wang et al. 2019a) to learn the global text centerline region segmentation (or shrunk ground truth) for individual text instances.

3.5 Optimization

Both IMA and CMA modules serve to make the network focus on the specific instance and character positions, which can be learned theoretically by only the final recognition part. However, in complicated scene text scenarios, it might be difficult for the network to converge without the assistance of position information. Nevertheless, we find that the model can be easily transferred if it has been pre-trained on

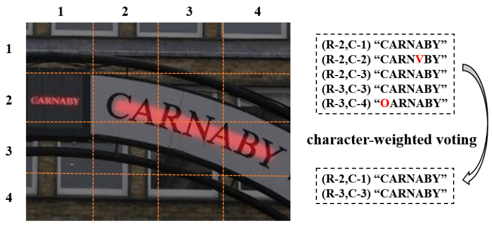

Figure 3: Illustration of the inference process. The final predictions are generated by merging all *occupied* grids' results through the character-weighted voting strategy.

the synthetic datasets with character-level supervision in advance. Therefore, the model can be optimized in two steps.

First, we can treat the learning of IMA and CMA as pure segmentation tasks. Together with centerline region segmentation, all segmentation tasks are trained using binary Dice coefficient loss (Milletari, Navab, and Ahmadi 2016), and the recognition tasks simply use cross-entropy loss. The global optimization can be written as

$$\mathcal{L} = \lambda_1 \mathcal{L}_{cls} + \lambda_2 \mathcal{L}_I + \lambda_3 \mathcal{L}_C + \mathcal{L}_{recog}, \tag{7}$$

where $\mathcal{L}_{cls}$, $\mathcal{L}_I$, $\mathcal{L}_C$ and $\mathcal{L}_{recog}$ denote the losses generated by the centerline segmentation, IMA, CMA and recognition, respectively. λ_1, λ_2 and λ_3 are weighted parameters.

Given the pre-trained weights on synthetic datasets, the model can be simply optimized at any scene by

$$\mathcal{L} = \lambda \mathcal{L}_{cls} + \mathcal{L}_{recog}. \tag{8}$$

Note that the pre-training step is actually a one-off task, and the CMA and IMA will then be mainly learned to fit the recognition task. In contrast to previous methods that need to balance the weights of detection and recognition, the end-to-end results of MANGO are mostly supervised by the final recognition tasks.

3.6 Inference

In the inference stage, the network outputs a batch ($S \times S$) of probability matrices ($L \times M$). According to the predictions of the centerline segmentation task, we can determine which grids should be treated as valid. We first conduct a Breadth-First-Search (BFS) to find the individual connected regions. Many text-like textures can be filtered during this process. Since each connected region may intersect with several grids, we adopt a character-weighted voting strategy to generate the final character strings, as shown in Figure 3.

Specifically, we calculate the *occupation ratio* $o_{i,j}$ of the connected region i with the grid j as the weight of each character. For the k-th character of the instance i, its character-weighted voting result is achieved by

$$instance_i^{(k)} = \arg\max \left(\sum_{j \in (S \times S)} (o_{i,j} \cdot x_{recog}[j][k]) \right), \tag{9}$$

where $x_{recog}(j,k) \in \mathbb{R}^M$ is the predicted probability vector of the k-th character of the j-th grid. Here, the *occupation ratio* provides the confidence of each grid, and multiple outputs fusion could generate more reliable results. The grid with the maximum occupation ratio will be treated as the rough output position, which can be replaced by any form according to the specific task.

4 Experiments

4.1 Datasets

We list the datasets used in this paper as follows.

Training Data. We use *SynthText 800k* (Gupta, Vedaldi, and Zisserman 2016) as the pretraining dataset. Both instance-level and character-level annotations are exploited to pre-train the PMA module.

In the finetuning stage, we aim to obtain a general text spotter supporting both regular and irregular scene text reading. Here, we construct a general dataset used for finetuning, which includes 150k images from the Curved SynthText (Liu et al. 2020), 13k images filtered from COCO-Text (Veit et al. 2016), 7k images filtered from ICDAR-MLT (Nayef et al. 2019) as well as all training images in ICDAR2013 (Karatzas et al. 2013), ICDAR2015 (Karatzas et al. 2015) and Total-Text (Ch'ng and Chan 2017). Note that, here we use only the instance-level annotations to train the network.

Testing Dataset. We evaluate our method on two standard text spotting benchmarks *ICDAR2013* (Karatzas et al. 2013) (IC13) and *ICDAR2015* (Karatzas et al. 2015) (IC15), which mainly contain horizontal and perspective text, and two irregular benchmarks *Total-Text* (Ch'ng and Chan 2017) and *SCUT-CTW1500* (Liu et al. 2019) (CTW1500), which contains many curved text.

We also demonstrate the generalization ability of our method in a license plate recognition dataset, CCPD (Xu et al. 2018).

4.2 Implementation Details

All experiments are implemented in Pytorch with 8×32 GB-Tesla-V100 GPUs.

Network Details. The feature extractor uses ResNet-50 (He et al. 2016) and FPN (Lin et al. 2017a) to obtain fused features from different feature map levels. Here, the (4×) feature map with C=256 is used to perform the subsequent training and testing tasks. L is set to 25 to cover most scene text words. The BiLSTM module has 256 hidden units.

Training Details. All models are trained by the SGD optimizer with batch-size=2, momentum=0.9 and weight-decay=1×10^{-4}. In the pretraining stage, the network is trained with an initial learning ratio of 1×10^{-2} for 10 epochs. The learning rate is divided by 10 every 3 epochs. In the finetuning stage, the initial learning rate is set to 1×10^{-3}. To balance the numbers of synthetic images and real images in each batch, we maintain the sampling ratio of 1:1 for the Curved SynthText dataset versus the other realistic datasets. The finetuning process lasts for 250k iterations in which the learning rate is divided by 10 at the 120k-th iteration and the 200k-th iteration.

We also conduct the data augmentation for all training processes, including 1) randomly scaling the longer side of the input images to lengths in the range [720, 1800],

Dataset	Method	Input Size	End-to-End			Word Spotting			FPS
			S	W	G	S	W	G	
IC13	He et al. † (He et al. 2018)	-	91.0	89.0	86.0	93.0	92.0	87.0	-
	FOTS † (Liu et al. 2018)	L-920	88.8	87.1	80.8	92.7	90.7	83.5	**22.0**
	TextNet (Sun et al. 2018)	L-920	89.8	88.9	83.0	94.6	**94.5**	87.0	2.7
	Mask TextSpotter* (Liao et al. 2019)	S-1000	93.3	91.3	88.2	92.7	91.7	87.7	3.1
	Boundary (Wang et al. 2020a)	L-1280	88.2	87.7	84.1	-	-	-	-
	Text Perceptron (Qiao et al. 2020)	L-1440	91.4	90.7	85.8	**94.9**	94.0	88.5	-
	MANGO	L-1080	89.7	89.3	85.3	94.0	93.4	88.4	9.8
	MANGO	L-1440	90.5	90.0	86.9	94.8	94.1	**90.1**	6.3
	MANGO*	L-1440	**93.4**	**92.3**	**88.7**	92.9	92.7	88.3	6.3
IC15	He et al. † (He et al. 2018)	-	82.0	77.0	63.0	85.0	80.0	65.0	-
	FOTS † (Liu et al. 2018)	L-2240	81.1	75.9	60.8	84.7	79.3	63.3	**7.5**
	TextNet (Sun et al. 2018)	-	78.7	74.9	60.5	82.4	78.4	62.4	-
	Mask TextSpotter* (Liao et al. 2019)	S-1600	83.0	77.7	73.5	82.4	78.1	73.6	2.0
	CharNet R-50 (Xing et al. 2019)	-	83.1	79.2	69.1	-	-	-	-
	TextDragon (Feng et al. 2019)	-	82.5	78.3	65.2	86.2	81.6	68.0	-
	Unconstrained (Qin et al. 2019)	S-900	83.4	79.9	68.0	-	-	-	-
	Boundary (Wang et al. 2020a)	1080×1920	79.7	75.2	64.1	-	-	-	-
	Text Perceptron (Qiao et al. 2020)	L-2000	80.5	76.6	65.1	84.1	79.4	67.9	-
	MANGO	L-1440	80.3	77.8	66.1	84.7	81.8	69.0	6.2
	MANGO	L-1800	81.8	78.9	67.3	**86.4**	**83.1**	70.3	4.3
	MANGO*	L-1800	**85.4**	**80.1**	**73.9**	85.2	81.1	**74.6**	4.3

Table 1: Results on IC13 and IC15. ‘S’, ‘W’ and ‘G’ mean recognition with strong, weak and generic lexicon, respectively. Superscript ‘*’ means that the method uses the specific lexicons from (Liao et al. 2019). Methods marked with † are not support for irregular text. Prefix ‘L-’ and ‘S-’ separately represent that resizing input images by the longer and shorter side.

Method	End-to-End		FPS
	None	Full	
Mask TextSpotter (Liao et al. 2019)	65.3	77.4	2.0
CharNet R-50 (Xing et al. 2019)	66.2	-	1.2
TextDragon (Feng et al. 2019)	48.8	74.8	-
Unconstrained (Qin et al. 2019)	67.8	-	-
Boundary (Wang et al. 2020a)	65.0	76.1	-
Text Perceptron (Qiao et al. 2020)	69.7	78.3	-
ABCNet (Liu et al. 2020)	64.2	75.7	**17.9**
MANGO (1280)	71.7	82.6	8.9
MANGO (1600)	**72.9**	**83.6**	4.3

Table 2: Results on Total-Text. ‘Full’ indicates lexicons of all images are combined. ‘None’ means lexicon-free. The number in brackets is the resized longer side of input image.

2) randomly rotating the images by angles in the range $[-15°, 15°]$, and 3) applying random brightness, jitters, and contrast to input images.

According to the density of text instances in different datasets, we set S=60 for evaluation of IC15 and S=40 for evaluations of IC13, Total-Text and CTW1500. We simply set all weight parameters as λ_1=λ_2=λ_3=λ=1.

Testing Details. Since the input image's size is an important essential impacting performance, we will report the performance in different input scales, i.e., keep the original ratio and resize the longer side of the image into a fixed value. All images are tested at a single scale.

Since current implementation only provides rough positions, we modify the end-to-end evaluation metric of (Wang, Babenko, and Belongie 2011) by considering all detection results with an IoU>0.1. In such case, the performance of previous methods will even be decreased due to the decline of precision by some low-grade proposal matching.

Method	End-to-End		FPS
	None	Full	
Text Perceptron (Qiao et al. 2020)	57.0	-	-
ABCNet (Liu et al. 2020)	45.2	74.1	-
MANGO (1080)	**58.9**	**78.7**	**8.4**

Table 3: Results on CTW1500. “Full” indicates lexicons of all images are combined. “None” means lexicon-free. The number in brackets is the resized longer side of input image.

4.3 Results on Text Spotting Benchmarks

Evaluation of regular text We first evaluate our method on IC13 and IC15, following the conventional evaluation metrics (Karatzas et al. 2015), two evaluation items (‘End-to-End’ and ‘Word Spotting’) based on three different lexicons (Strong, Weak, and Generic).

Table 1 shows the evaluation results. Compared to previous methods evaluated with conventional lexicons, our method achieves the best results on the ‘Generic’ item (except for the end-to-end generic result on IC15), and obtains the competitive results on the rest evaluated items (‘Strong’ and ‘Weak’). Compared to the recent state-of-the-art, Mask TextSpotter (Liao et al. 2019) using the specific lexicon, our method obviously outperforms it on all evaluation items.

For the inference speed, though FOTS obtains the highest FPS (Frames Per Second), it fails to handle the irregular cases. Compared with those irregular-based methods, our method achieves the highest FPS.

Evaluation of irregular text We test our method on Total-Text, as shown in Table 2. We see that our method surpasses the state-of-the-art by 3.2% and 5.3% in “None” and “Full” metrics. Notice that even without an explicit rectification mechanism, our model can handle irregular text well only

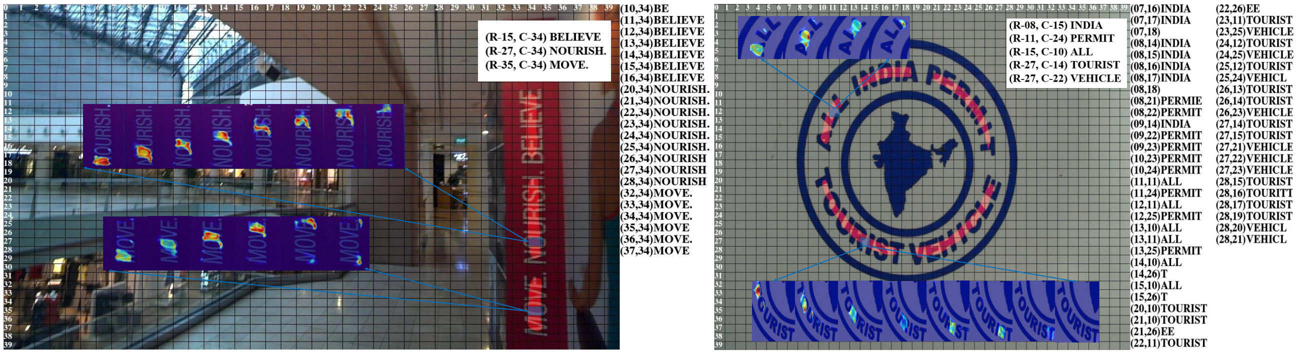

Figure 4: Visualization of End-to-End recognition on IC15 and Total-Text with some cropped instances' CMA, where $S = 40$. The right parts of the images show all the positive predictions before character voting. Two numbers in the brackets (.,.) separately mean the row and column number.

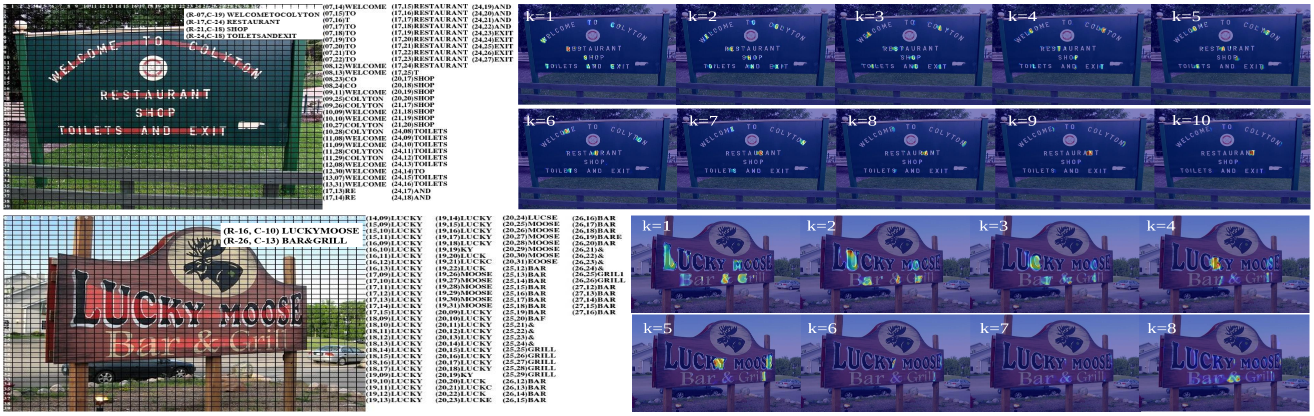

Figure 5: Visualization of the end-to-end results on SCUT-CTW1500 with the CMA maps in different character positions.

S	IC13				IC15				Total-Text		
	S	W	G	FPS	S	W	G	FPS	None	Full	FPS
20	83.2	82.5	78.7	**6.58**	33.8	33.0	29.1	**5.12**	46.9	58.5	**4.49**
30	88.8	88.3	85.9	6.32	69.4	67.1	57.8	4.57	69.8	80.6	4.37
40	**90.5**	**90.0**	**86.9**	6.25	80.4	77.3	66.8	4.43	72.9	**83.6**	4.28
50	90.3	89.8	86.7	6.12	81.6	78.8	**67.8**	4.38	**73.1**	83.0	4.23
60	89.9	89.3	85.7	6.07	**81.8**	**78.9**	67.3	4.27	72.2	82.9	4.21

Table 4: Evaluation results under different grid numbers.

Supervision Type	IC15			Total-Text	
	S	W	G	None	Full
Strong	81.8	78.9	67.3	72.9	83.6
Weak	81.8	78.3	64.0	69.7	80.6

Table 5: Results under different detection supervision types. 'Strong' means the original annotations, and 'Weak' means rectangular bounding box annotations.

driven by the recognition supervision. Though the inference speed is about 1/2 of ABCNet with the test scale of 1280, our method achieves the remarkable performance gains.

We also evaluate our method on CTW1500. There are few works that reported the end-to-end results because it mainly contains the line-level text annotations. To adapt to this situation, we retrain the detection branch on the training set of CTW1500 to learn the line-level centerline segmentation, and fix the weights of the backbone and other branches. Note that the recognition will not be affected and still output the word-level sequences. The final results will be simply concatenated from left to right according to the inferred connected regions. Chinese characters are set as NOT CARE.

Results are shown in Table 3. We find that our method obviously surpasses previous advances by 1.9% and 4.6% on 'None' and 'Full' metrics, respectively. Therefore, we believe that if there are enough data with only line-level annotations, our model can adapt to such scenarios well.

4.4 Visualization Analysis

Figure 4 visualizes the end-to-end text spotting results on IC15 and Total-Text. We detailedly show the prediction results of each positive grids ($o_{i,j}$>0.3) before character voting. We see that our model can correctly focus on the cor-

Method	Base(100k)	DB	FN	Rotate	Tilt	Weather	Challenge	AP
SSD300 + HC	98.3	96.6	**95.9**	88.4	91.5	87.3	83.8	95.2
RPnet(Xu et al. 2018)	98.5	96.9	94.3	90.8	92.5	87.9	**85.1**	95.5
MANGO	**99.0**	**97.1**	95.5	**95.0**	**96.5**	**95.9**	83.1	**96.9**

Table 6: End-to-End recognition precision results on CCPD.

responding position and learn the complex reading order of character sequences for arbitrary-shaped (*e.g.* curved or vertical) text instances. After the character voting strategy, the word with the highest confidence will be generated.

We also demonstrate some of the results of CTW1500 with their visualized CMA, as shown in Figure 5. Note that we only fine-tune the line-level segmentations part based on the dataset's position labels while fixing the remaining parts. Here, We visualize the feature maps of CMAs by overlaying all grids' attention map in the same character position (k) as:

$$x^*_{char}[k] = \sum_{i \in S^2} x_{char}[i][k] \tag{10}$$

where $x^*_{char}[k] \in \mathbb{R}^{L \times H \times W}$, and $k = 1, 2, ..., L$. As shown in Figure 5, we see that model indeed pays attention to all correct character positions of all text instances in the image at the same time. At the end of each text instance, there is a highlight region that means the 'EOS' position's attention.

4.5 Ablation Studies

Ablation of grid numbers The grid number S^2 is a crucial parameter affecting the final results. If S is too small, there will be too many texts occupying the same grid. Otherwise, too big of S will result in more computation cost. Here, we conduct experiments to find the feasible value of S for different datasets.

From Table 4, we find that the best S for both IC13 and Total-Text is 40. The value for IC15 is 60. This is because IC15 contains more dense and small instances. In sum, the overall performance increases along with increasing of S and becomes stable when $S >= 40$. Of course, FPS will decrease slightly along with increasing of S.

Evaluation of Coarse Position Supervision As mentioned above, our method can be learned well with only rough position information. To demonstrate this, we also conduct the experiments to transfer all localization annotations as the form of rectangular bounding boxes. We simply adopt the RPN head as the detection branch.

Table 5 shows the results on IC15 and Total-Text. Even with the rough position supervision, MANGO only decreases the performance ranging from 0% to 3%, and is comparable with the state-of-the-arts. Note that, the coarse position only serves the grid selection so that it can be simplified as much as possible according to specific tasks' requirement.

4.6 Challenging License Plate Recognition without Position Annotations

To demonstrate the model's generalization ability, we conduct experiments to evaluate the end-to-end license plate recognition results on a public dataset, CCPD (Xu et al. 2018). For fairness, we follow the same experimental settings and use the initially released version of the dataset with 250k images. The CCPD-Base dataset is separated into two equal parts: 100k samples for training and 100k samples for testing. There are six complex testing sets (including DB, FN, Rotate, Tilt, Weather, and Challenge) for evaluating the algorithm's robustness, which have 50k images in total.

Since each image in CCPD contains only one plate, our model can be further simplified by removing the detection branch to predict the final character sequence directly. Therefore, the grid number is reduced to $S = 1$, and the maximum sequence length is set to $L = 8$. We directly fine-tune the model (having been pre-trained by SynthText) on CCPD training set with only the sequence-level annotations, and then evaluate the final recognition accuracy on the above seven testing datasets. The testing phase is performed on the original image with a size of 720 × 1160.

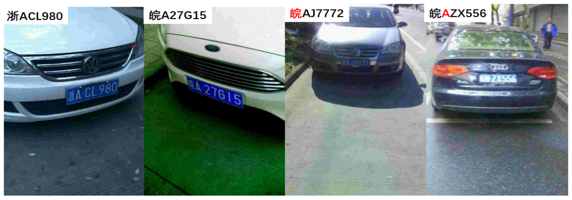

Figure 6: Visualization result on CCPD. Since $S = 1$, no position information is involved.

Table 6 shows the end-to-end recognition results. Although the proposed method is not designed for the license plate recognition task, it still can be easily transferred to such scenarios. We see that the proposed model outperforms previous methods in 5 out of 7 test sets and achieves the highest average precision. Figure 6 shows some visualization results on the CCPD test sets. The failure samples are mainly from the situation that images are too blurred to be recognized.

This experiment demonstrates that in many situations with only one text instance (*e.g.,* industrial printing recognition or meter dial recognition), a good End-to-End model can be obtained without detection annotations.

5 Conclusion

In this paper, we propose a novel one-staged scene text spotter named MANGO. This model removes the RoI operations and designs the position-aware attention module to coarsely localize the text sequences. After that, a lightweight sequence decoder is applied to obtain all of the final character sequences into a batch. Experiments show that our method achieves competitive and even state-of-the-art results on popular benchmarks.

References

Baek, Y.; Lee, B.; Han, D.; Yun, S.; and Lee, H. 2019. Character region awareness for text detection. In *CVPR*, 9365–9374.

Bartz, C.; Yang, H.; and Meinel, C. 2018. SEE: towards semi-supervised end-to-end scene text recognition. In *AAAI*, 6674–6681.

Bookstein, F. L. 1989. Principal Warps: Thin-Plate Splines and the Decomposition of Deformations. *IEEE TPAMI* 11(6): 567–585.

Busta, M.; Neumann, L.; and Matas, J. 2017. Deep textspotter: An end-to-end trainable scene text localization and recognition framework. In *ICCV*, 2204–2212.

Chen, H.; Sun, K.; Tian, Z.; Shen, C.; Huang, Y.; and Yan, Y. 2020. BlendMask: Top-down meets bottom-up for instance segmentation. In *CVPR*, 8573–8581.

Cheng, Z.; Bai, F.; Xu, Y.; Zheng, G.; Pu, S.; and Zhou, S. 2017. Focusing Attention: Towards Accurate Text Recognition in Natural Images. In *ICCV*, 5076–5084.

Ch'ng, C. K.; and Chan, C. S. 2017. Total-text: A Comprehensive Dataset for Scene Text Detection and Recognition. In *ICDAR*, volume 1, 935–942.

Duan, K.; Bai, S.; Xie, L.; Qi, H.; Huang, Q.; and Tian, Q. 2019. Centernet: Keypoint triplets for object detection. In *ICCV*, 6569–6578.

Feng, W.; He, W.; Yin, F.; Zhang, X.; and Liu, C. 2019. TextDragon: An End-to-End Framework for Arbitrary Shaped Text Spotting. In *ICCV*, 9075–9084.

Gupta, A.; Vedaldi, A.; and Zisserman, A. 2016. Synthetic Data for Text Localisation in Natural Images. In *CVPR*, 2315–2324.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *CVPR*, 770–778.

He, P.; Huang, W.; He, T.; Zhu, Q.; Qiao, Y.; and Li, X. 2017. Single Shot Text Detector with Regional Attention. In *ICCV*, 3066–3074.

He, T.; Tian, Z.; Huang, W.; Shen, C.; Qiao, Y.; and Sun, C. 2018. An End-to-End TextSpotter with Explicit Alignment and Attention. In *CVPR*, 5020–5029.

Hochreiter, S.; and Schmidhuber, J. 1997. Long Short-Term Memory. *Neural Computation* 9(8): 1735–1780.

i Bigorda, L. G.; and Karatzas, D. 2017. TextProposals: A text-specific selective search algorithm for word spotting in the wild. *Pattern Recognition* 70: 60–74.

Jaderberg, M.; Vedaldi, A.; and Zisserman, A. 2014. Deep Features for Text Spotting. In *ECCV*, 512–528.

Karatzas, D.; Gomez-Bigorda, L.; Nicolaou, A.; Ghosh, S.; Bagdanov, A.; Iwamura, M.; Matas, J.; Neumann, L.; Chandrasekhar, V. R.; Lu, S.; et al. 2015. ICDAR 2015 Competition on Robust Reading. In *ICDAR*, 1156–1160.

Karatzas, D.; Shafait, F.; Uchida, S.; Iwamura, M.; i Bigorda, L. G.; Mestre, S. R.; Mas, J.; Mota, D. F.; Almazan, J. A.; and De Las Heras, L. P. 2013. ICDAR 2013 Robust Reading Competition. In *ICDAR*, 1484–1493.

Li, H.; Wang, P.; and Shen, C. 2017. Towards End-to-end Text Spotting with Convolutional Recurrent Neural Networks. In *ICCV*, 5248–5256.

Liao, M.; Lyu, P.; He, M.; Yao, C.; Wu, W.; and Bai, X. 2019. Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes. *TPAMI* 1(1).

Liao, M.; Shi, B.; and Bai, X. 2018. TextBoxes++: A Single-Shot Oriented Scene Text Detector. *IEEE TIP* 27(8): 3676–3690.

Liao, M.; Shi, B.; Bai, X.; Wang, X.; and Liu, W. 2017. TextBoxes: A Fast Text Detector with a Single Deep Neural Network. In *AAAI*, 4161–4167.

Lin, T.-Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; and Belongie, S. 2017a. Feature Pyramid Networks for Object Detection. In *CVPR*, 2117–2125.

Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; and Dollár, P. 2017b. Focal loss for dense object detection. In *ICCV*, 2980–2988.

Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S. E.; Fu, C.; and Berg, A. C. 2016. SSD: Single Shot MultiBox Detector. In Leibe, B.; Matas, J.; Sebe, N.; and Welling, M., eds., *ECCV*, 21–37.

Liu, X.; Liang, D.; Yan, S.; Chen, D.; Qiao, Y.; and Yan, J. 2018. FOTS: Fast Oriented Text Spotting with a Unified Network. In *CVPR*, 5676–5685.

Liu, Y.; Chen, H.; Shen, C.; He, T.; Jin, L.; and Wang, L. 2020. ABCNet: Real-time Scene Text Spotting with Adaptive Bezier-Curve Network. In *CVPR*, 9809–9818.

Liu, Y.; Jin, L.; Zhang, S.; Luo, C.; and Zhang, S. 2019. Curved Scene Text Detection via Transverse and Longitudinal Sequence Connection. *Pattern Recognition* 90: 337–345.

Long, S.; Ruan, J.; Zhang, W.; He, X.; Wu, W.; and Yao, C. 2018. Textsnake: A Flexible Representation for Detecting Text of Arbitrary Shapes. In *ECCV*, 19–35.

Luo, C.; Jin, L.; and Sun, Z. 2019. Moran: A multi-object rectified attention network for scene text recognition. *Pattern Recognition* 90: 109–118.

Ma, J.; Shao, W.; Ye, H.; Wang, L.; Wang, H.; Zheng, Y.; and Xue, X. 2018. Arbitrary-Oriented Scene Text Detection via Rotation Proposals. *IEEE TMM* 20(11): 3111–3122.

Milletari, F.; Navab, N.; and Ahmadi, S.-A. 2016. V-net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation. In *3DV*, 565–571.

Nayef, N.; Liu, C.; Ogier, J.; Patel, Y.; Busta, M.; Chowdhury, P. N.; Karatzas, D.; Khlif, W.; Matas, J.; Pal, U.; and Burie, J. 2019. ICDAR2019 Robust Reading Challenge on Multi-lingual Scene Text Detection and Recognition - RRC-MLT-2019. In *ICDAR*, 1582–1587.

Neumann, L.; and Matas, J. 2015. Real-time lexicon-free scene text localization and recognition. *TPAMI* 38(9): 1872–1885.

Qiao, L.; Tang, S.; Cheng, Z.; Xu, Y.; Niu, Y.; Pu, S.; and Wu, F. 2020. Text Perceptron: Towards End-to-End Arbitrary-Shaped Text Spotting. In *AAAI*, 11899–11907.

Qin, S.; Bissacco, A.; Raptis, M.; Fujii, Y.; and Xiao, Y. 2019. Towards unconstrained end-to-end text spotting. In *ICCV*, 4704–4714.

Redmon, J.; Divvala, S. K.; Girshick, R. B.; and Farhadi, A. 2016. You Only Look Once: Unified, Real-Time Object Detection. In *CVPR*, 779–788.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015a. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *NeurIPS*, 91–99.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015b. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *TPAMI* 39(6).

Shi, B.; Bai, X.; and Yao, C. 2017. An End-to-End Trainable Neural Network for Image-based Sequence Recognition and Its Application to Scene Text Recognition. *IEEE TPAMI* 39(11): 2298–2304.

Shi, B.; Wang, X.; Lyu, P.; Yao, C.; and Bai, X. 2016. Robust Scene Text Recognition with Automatic Rectification. In *CVPR*, 4168–4176.

Sun, Y.; Zhang, C.; Huang, Z.; Liu, J.; Han, J.; and Ding, E. 2018. TextNet: Irregular Text Reading from Images with an End-to-End Trainable Network. In Jawahar, C. V.; Li, H.; Mori, G.; and Schindler, K., eds., *ACCV*, 83–99.

Tian, Z.; Shen, C.; and Chen, H. 2020. Conditional Convolutions for Instance Segmentation. *arXiv preprint arXiv:2003.05664* .

Tian, Z.; Shen, C.; Chen, H.; and He, T. 2019. FCOS: Fully Convolutional One-Stage Object Detection. In *ICCV*, 9626–9635.

Veit, A.; Matera, T.; Neumann, L.; Matas, J.; and Belongie, S. 2016. COCO-Text: Dataset and Benchmark for Text Detection and Recognition in Natural Images. *arXiv preprint arXiv:1601.07140* .

Wang, H.; Lu, P.; Zhang, H.; Yang, M.; Bai, X.; Xu, Y.; He, M.; Wang, Y.; and Liu, W. 2020a. All You Need Is Boundary: Toward Arbitrary-Shaped Text Spotting. In *AAAI*, 12160–12167.

Wang, K.; Babenko, B.; and Belongie, S. J. 2011. End-to-end scene text recognition. In Metaxas, D. N.; Quan, L.; Sanfeliu, A.; and Gool, L. V., eds., *ICCV*, 1457–1464.

Wang, P.; Li, H.; and Shen, C. 2019. Towards End-to-End Text Spotting in Natural Scenes. *arXiv preprint arXiv:1906.06013* .

Wang, T.; Wu, D. J.; Coates, A.; and Ng, A. Y. 2012. End-to-End Text Recognition with Convolutional Neural Networks. In *ICPR*, 3304–3308.

Wang, T.; Zhu, Y.; Jin, L.; Luo, C.; Chen, X.; Wu, Y.; Wang, Q.; and Cai, M. 2020b. Decoupled Attention Network for Text Recognition. In *AAAI*, 12216–12224.

Wang, W.; Xie, E.; Li, X.; Hou, W.; Lu, T.; Yu, G.; and Shao, S. 2019a. Shape Robust Text Detection With Progressive Scale Expansion Network. In *CVPR*, 9336–9345.

Wang, X.; Kong, T.; Shen, C.; Jiang, Y.; and Li, L. 2019b. SOLO: Segmenting objects by locations. *arXiv preprint arXiv:1912.04488* .

Wang, X.; Zhang, R.; Kong, T.; Li, L.; and Shen, C. 2020c. SOLOv2: Dynamic, Faster and Stronger. *arXiv preprint arXiv:2003.10152* .

Xie, E.; Sun, P.; Song, X.; Wang, W.; Liu, X.; Liang, D.; Shen, C.; and Luo, P. 2020. Polarmask: Single shot instance segmentation with polar representation. In *CVPR*, 12193–12202.

Xing, L.; Tian, Z.; Huang, W.; and Scott, M. R. 2019. Convolutional character networks. In *ICCV*, 9126–9136.

Xu, Y.; Wang, Y.; Zhou, W.; Wang, Y.; Yang, Z.; and Bai, X. 2019. TextField: Learning a Deep Direction Field for Irregular Scene Text Detection. *IEEE TIP* 28(11): 5566–5579.

Xu, Z.; Yang, W.; Meng, A.; Lu, N.; and Huang, H. 2018. Towards End-to-End License Plate Detection and Recognition: A Large Dataset and Baseline. In *ECCV*, 255–271.

Zhan, F.; and Lu, S. 2019. Esir: End-to-end scene text recognition via iterative image rectification. In *CVPR*, 2059–2068.

Zhou, X.; Yao, C.; Wen, H.; Wang, Y.; Zhou, S.; He, W.; and Liang, J. 2017. EAST: An Efficient and Accurate Scene Text Detector. In *CVPR*, 2642–2651.