Pattern Recognition Letters 149 (2021) 1–8

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Unconstrained end-to-end text reading with feature rectification☆

Chen Du[a,b], Yanna Wang[a], Chunheng Wang[a,*], Baihua Xiao[a], Cunzhao Shi[a]

[a] The State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
[b] University of Chinese Academy of Sciences, Beijing, China

ARTICLE INFO

Article history:
Received 27 October 2020
Revised 25 April 2021
Accepted 31 May 2021
Available online 15 June 2021

Keywords:
Text recognition
Text detection
Position-sensitive network
Features incompatibility
End-to-end

ABSTRACT

We propose an end-to-end trainable network that can simultaneously localize and recognize irregular text from images. Specifically, we find the feature incompatibility problem, which arises from the contradiction between detection and recognition tasks for feature extraction of the convolutional neural network, and propose to introduce the larger-scale features for the recognition part to improve the accuracy of recognition instead of using the same feature with the detection. To extract effective text features for perspective and curved text recognition, we propose a position-sensitive network to rectify the text proposal features in the recognition branch. The position-sensitive network, which is trained in a weak supervision way, takes the proposal detection feature as input and outputs the feature rectification information. Experiments demonstrate that the proposed method can achieve state-of-the-art or highly competitive performance compared with baselines on a number of benchmarks.

1. Introduction

Traditional optical character recognition (OCR) pipeline methods generally consist of two components, text detection and text recognition. The goal of text detection is to spot text instances in the input image and get their bounding boxes, while text recognition aims to recognize the detected text region by decoding its textual content. Although existing text detectors and recognizers work well on horizontal texts, it still remains as a challenge when it comes to spotting curved text instances in scene images.

Classical two-step pipelines deal with curved samples from the two aspects of text detection and recognition. To recognize curved texts in an image, many detectors [1–4] try to predict complex geometry or apply complicated post-processing techniques to get the geometric attributes of curved texts, and the recognizers [5,6]apply multi-directional encoding or take rectification modules to enhance the accuracy of the recognizer on curved texts. In the two-step pipeline, the performance of text recognition heavily relies on text detection results. Each model in the pipeline depends on the outputs of the previous step, which makes it hard to jointly maximize the end-to-end performance, and fine-tune the engine with new data or adapt it to a new domain.

Scene text spotting/end-to-end recognition is a task that combines the detection and recognition tasks which overcome those disadvantages and thus have recently started gaining traction in the research community[7–9]. The detector and recognizer share the same CNN feature extractor in the end-to-end recognition framework. The detector and recognizer are jointly optimized during training and then predict locations and transcriptions in a single forward pass at inference time. However, different feature descriptions are required in the detection task and the recognition task. The detector tends to extract the common features of the text or the overall characteristics of the text area to implement the segmentation task of text area and detection of text, while the recognizer needs more detailed information to achieve better classification effect. Due to the different tasks between text detection and recognition, there is a contradiction between the two tasks for feature extraction of convolutional neural networks. Meanwhile, the scale of features for text recognition is also a key bottleneck for improving the performance of the recognition branch. We denote above problem features incompatibility between text detection and recognition. The feature incompatibility problem makes the network be trained more difficult and struggle to generalize and produce convincing results on more challenging datasets with curved text.

In this paper, we propose a simple and flexible end-to-end OCR model to read irregular text from images, especially for perspective and curved cases. Fig. 1 shows the concept of the proposed method. By sharing the convolutional layers, we can compute the shared feature maps from the input image only once and imple-

☆ Handle by Associate Editor Lianwen Jin.

* Corresponding author.
E-mail addresses: duchen2016@ia.ac.cn (C. Du), wangyanna2013@ia.ac.cn (Y. Wang), chunheng.wang@ia.ac.cn (C. Wang), baihua.xiao@ia.ac.cn (B. Xiao), cunzhao.shi@ia.ac.cn (C. Shi).

https://doi.org/10.1016/j.patrec.2021.05.017
0167-8655/

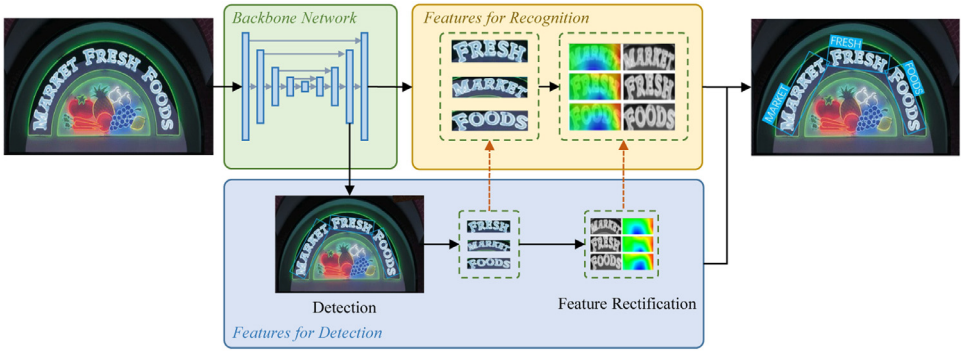

Fig. 1. Concept of the proposed method. We use different scale features for detection and recognition respectively in order to deal with the feature incompatibility problem and propose a feature rectification module to rectify the text proposal features in the recognition branch.

ment text detection and recognition simultaneously. Different from the existing methods in which the text detection branch and recognition branch using the same size convolutional features in the backbone architecture, we use different scale features for detection and recognition respectively in order to deal with the feature incompatibility problem. Specifically, we combine larger scale features in the recognition part to ensure the detailed information needed for recognition, which further improves the accuracy of recognition.

The text detection branch is built on the feature map for detection to predict the results of detection bounding boxes. The RoIRotate operator extracts text proposal features corresponding to the detection results from the feature map for detection and the feature map for recognition respectively. The text proposal features from detection are used as the input of the position-sensitive network (PSN) to learn the feature rectification information. The text proposal features from recognition are rectified with the rectify information and then are fed into Recurrent Neural Network (RNN) encoder and attention-based sequence recognition network for text recognition. The whole system is designed as an end-to-end trainable network as all the modules in the network are differentiable.

The contributions of this paper are listed as follows:

- We propose a flexible and powerful end-to-end trainable framework that can simultaneously localize and recognize both regular and irregular text in one model.
- We find the feature incompatibility problem, which arises from the contradiction between detection and recognition tasks for feature extraction of the convolutional neural network, and propose to introduce the larger-scale features for the recognition part to improve the accuracy of recognition instead of using the same feature with the detector.
- To extract effective text features for perspective and curved text recognition, we propose PSN to rectify the text proposal features in the recognition branch. The PSN, which is trained in a weak supervision way, takes the proposal detection feature as input and outputs the feature rectification information.
- Experiments on datasets demonstrate that the proposed method can achieve comparable or state-of-the-art performance on a number of text detection and text spotting benchmarks.

2. Related work

In this section, we briefly review the related works including text detection, text recognition and end-to-end text reading and spotting.

2.1. Scene text detection

The goal of text detection is to spot the text instances in the input image and get their bounding boxes. With the development of Convolutional Neural Networks (CNNs), the state-of-the-art object detection frameworks such as Faster-RCNN [10] and SSD [11] have been widely applied to text detection field. Generally, these methods can be divided into two categories: Regression-based methods and segmentation-based methods. Regression-based methods [12–15] take words or text lines as a special case of object and aim to directly regress the bounding boxes of the text instances. Segmentation-based text detectors [1,3,4] are built on Fully Convolutional Networks [16], by generating text score maps or producing pixel-wise prediction of text or non-text. Nevertheless, among these methods, most of the regression-based methods often require complex anchor design and cumbersome multiple stages, while segmentation-based methods need complicated post-processing steps to get final detection results during inference. Sheng et al. [17] points out that existing approaches could not keep a good balance between accuracy and speed and then proposes Pyrboxes to detect multi-scale scene texts with proper speed and accuracy model.

2.2. Scene text recognition

Text recognition aims to recognize the detected text regions by decoding its textual content. Recent scene text recognition methods can be grouped into two main categories, regular text recognition and irregular text recognition. Shi et al. [18] propose CRNN which takes LSTM models to encode the CNN features and adopts CTC to decode the encoded sequence to recognize scene text images. After CRNN, multiple variants have been proposed to improve performance. Furthermore, attention based methods [19,20] focus on informative regions to achieve better performance. Suman et al. [20] introduce a LSTM-based visual attention model for unconstrained scene text recognition. Cheng et al. [21] propose the Focusing Attention Network that employs a focusing attention mechanism to automatically draw back the drifted attention. Zhao et al. [22] propose an adversarial learning based attentional scene text recognizer to solve the distortion problem of scene text image. Zhang et al. [23] propose a scale-aware hierarchical attention network (SaHAN) to solve the character scale-variation problem in scene text recognition. To handle irregular input images, transformation modules [5,6] have been proposed to normalize text images.

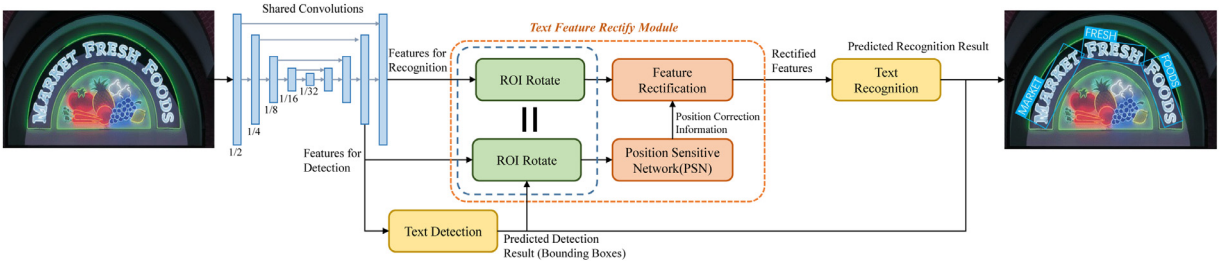

Fig. 2. Schematic overview of our end-to-end OCR model architecture.

2.3. End-to-end scene text reading

The basic idea behind end-to-end text reading is to have the detector and recognizer share the same CNN feature extractor. FOTS [7] concatenates popular detection and recognition methods by introducing RoIRotate to share convolutional features between detection and recognition. In Textnet [9], the detector outputs quadrilaterals and an attention-based model is used to decode the textual content. Mask TextSpotter [24] takes advantage of the designed unified model to treat the recognition task as a semantic segmentation problem. CharNet [8] directly outputs bounding boxes of words and characters, with corresponding character labels. Qin et al. [25] propose an end-to-end OCR model which is based on Mask R-CNN [26] and attention decoder. All these approaches can be trained in an end-to-end fashion. The detector and recognizer are jointly optimized during training and then predict locations and transcriptions in a single forward pass at inference time. However, the detector tends to extract the common features of the text or the overall characteristics of the text area to implement the segmentation task of text area and detection of text, while the recognizer needs more detailed information to achieve better classification effect. Due to the different tasks between text detection and recognition, there is a contradiction between the two tasks for feature extraction of convolutional neural networks.

3. Methodology

In this section, we describe the proposed method in detail. The overall architecture of our end-to-end OCR model is schematically illustrated in Fig. 2. Our main objective is to precisely localize and recognize text in natural images. To this end, the overall network architecture consists of four building blocks: the backbone network, the text detection branch, the position sensitive network, and the text recognition branch.

We employ a fully convolutional network architecture based on ResNet [27] and Feature Pyramid Network [28] as our backbone for our framework. By sharing the convolutional layers, we can compute the shared feature maps from the input image only once and implement text detection and recognition simultaneously. Different from the existing methods in which the text detection branch and recognition branch use the same size convolutional features in the backbone architecture, we use different scale features for detection and recognition respectively in order to deal with the feature incompatibility problem. Specifically, we aggregate the multi-scale features into $\frac{1}{4}$ and $\frac{1}{2}$ resolution of the input image for the detection and recognition separately. The detector part of the model is based on EAST which has been widely used in scene text detection and other related tasks. The text detection branch can directly predict the locations of text in quadrangles. In order to obtain feature information from quadrangle proposals for recognition, we develop a RoIRotate layer to convert the features of quadrangle proposals into fixed-height features while keeping the original region aspect ratio and a Position Sensitive Network (PSN) to rectify the text proposal features using the features for detection as input. Finally, the text recognition branch recognizes words in region proposals. The overall architecture containing both detection and recognition branches can be jointly trained in an end-to-end manner.

3.1. Text detection module

A fully convolutional network is adopted as the text detector. We upscale and merge the feature maps from 1/32 to 1/4 size of the original input image. After extracting the detection features, one convolution is applied to output dense per-pixel predictions of words.

The output of the text detection branch consists of two parts: the rotated box (RBox) for predicting the text bounding box and the score map for computing the probability of each pixel being a positive text. For each positive sample in the score map, the rotated box is represented by four channels for predicting the distance of a positive pixel to the top, bottom, left, right sides of the bounding box that contains the positive pixel, and one channel for predicting the orientation of the related bounding box. Finally, the text detection results in quadrangles are produced by NMS (Non-Maximum Suppression) in the predicting stage.

During the training stage, the detection branch loss function is composed of two terms: text classification term and bounding box regression term. The whole detection loss can be written as:

$$\mathcal{L}_{detect} = \mathcal{L}_{reg} + \lambda_c \mathcal{L}_{cls} \tag{1}$$

where $\mathcal{L}_{reg}$ and $\mathcal{L}_{cls}$ represent loss for regression loss and text classification loss, respectively. λ_c is the hyper-parameter to control the balance among losses. We set $\lambda_c = 0.01$ in our experiments.

Text classification loss: The text classification term can be seen as pixel-wise classification loss between text and non-text. To automatically balance the loss between positive and negative classes, we use class-balanced cross-entropy loss function. The loss function for classification can be formulated as:

$$\mathcal{L}_{cls} = \frac{1}{|R|} \sum_{x \in R} (-\beta p_x^* \log p_x - (1-\beta)(1-p_x^*) \log(1-p_x)) \tag{2}$$

where $\beta = \frac{|R^-|}{|R^+|}$, $|R^+|$ and $|R^-|$ denote the number of elements in text and non-text ground truth label sets, $R = |R^+| \bigcup |R^-|$. p_x is the prediction of score map and p_x^* is the binary label that indicates text or non-text. $|\cdot|$ represents the number of elements in a set.

Bounding box regression loss: As for the regression loss, we adopt the IoU loss and the loss of rotation angle for the bounding box regression loss. The overall regression loss is the weighted sum of IoU loss and angle loss, computed as

$$\mathcal{L}_{reg} = \frac{1}{|R|} \sum_{x \in R} -log\text{IoU}(R_x, R_x^*) + \lambda_\theta (1 - cos(\theta_x - \theta_x^*)) \tag{3}$$

where $-log\ \text{IoU}(R_i, R_i^*)$ is the IoU loss between the predicted bounding box R_x and the ground truth R_x^*. The second term is the loss of rotation angle. θ_x is the prediction of the rotation angle and θ_x^* represents the ground truth. λ_θ is set to 10 in our experiments.

Table 1
Architecture of the PSN.

Type	Configurations [kernel, stride, padding]	Out Channels
conv_bn_relu	[3,1,1]	64
conv_bn_relu	[3,1,1]	32
max-pool	[2,2,0]	32
conv_bn_relu	[3,1,1]	16
conv_bn	[1,1]	2
Tanh	–	2
unpool	×4	2

3.2. Position sensitive network

In this stage, a RoIRotate layer is adopted to crop out text instance features with fixed height and unchanged aspect ratio using predicted rotated rectangles or quadrilaterals. With the RoIRotate layer, the extracted features are reshaped to be a sequence of features $\mathcal{V}^d$ from the detection branch and a sequence of features $\mathcal{V}^r$ from the recognition branch. The height of the text instance feature is set to 8 in $\mathcal{V}^d$ and 16 in $\mathcal{V}^r$. We use different scale features extracted from the backbone network for detection and recognition respectively. Specifically, for the input of the recognition branch, convolution features are 1/2 resolution of the input image but 1/4 for the detection branch. The height 8 in the detection branch or 16 in the recognition branch could be considered to correspond to the original cropped text image with a height of 32. This scale can adapt to the size change of different text images and is also the most commonly used scale in the scene text recognition methods [5,6,18].

The detailed architecture of the PSN is given in Table 1. Each convolutional layer is followed by a batch normalization layer and a ReLU layer except for the last one. The PSN takes the features in $\mathcal{V}^d$ as input and predicts the offsets of each part of the text instance features. Specifically, the PSN branch is a stack of convolutional layers, which is composed of two 3×3 convolutional layers, followed by a max-pool layer and another 3×3 convolutional layer and a 1×1 convolutional layer. The offsets are predicted by the PSN branch in parallel with text detection branch. The offset values are activated by $Tanh(\cdot)$, resulting in values within the range of $(-1, 1)$. There are two channels in the offset maps for each v^d in $\mathcal{V}^d$, which denote the x-coordinate and y-coordinate respectively. As the predicted offsets are used to rectify the recognition features, we apply bilinear interpolation to smoothly resize the offset maps to the size of the corresponding v^r in $\mathcal{V}^r$. The output offset maps are the same size as the recognition corresponding features.

In our method, different scale features in the shared convolutional neural network are used in the detection branch and the recognition branch to solve the feature incompatibility problem between the two different tasks. The position-sensitive network is designed to connect the detection and recognition branches and rectify the text proposal features in the recognition branch. The position-sensitive network increases the information transmission path between the detection branch and the recognition branch so that the model can more effectively use the complementary information between text detection features and text recognition features.

3.3. Text recognition module

The text recognition branch aims to predict text contents from the text region features. As mentioned above, the text regions are converted into fixed-height features $\mathcal{V}^r$ from the recognition branch via the RoIRotate layer. Then PSN is employed to rectify the shared feature maps $\mathcal{V}^r$ to regular onesV^r.

Table 2
The detailed structure of the text recognition branch.

Type	Configurations [kernel, stride, padding]	Out Channels
conv_bn_relu	[3,1,1]	64
conv_bn_relu	[3,(2,1),1]	64
conv_bn_relu	[3,1,1]	128
conv_bn_relu	[3,(2,1),1]	128
conv_bn_relu	[3,1,1]	256
max-pool	[(2,1),(2,1)]	256
conv_bn_relu	[3,(2,1),0]	256
bi-directional lstm	256 hidden units	–
bi-directional lstm	256 hidden units	–
GRU	256 hidden units	–

In our framework, we take advantage of the shared convolutional layers to process feature extraction. Thus, these text instance feature maps are directly fed into sequence modeling and transcription layers. Our recognition network employs a sequence-to-sequence model with an attention mechanism. It consists of an encoder and a decoder. The detailed network structure is given in Table 2. The major structure of the recognizer is a CNN-BLSTM framework. The encoder takes high-level semantic features as input. Each convolutional layer is followed by a batch normalization layer and a ReLU layer. Then, two layers of bidirectional LSTM with 256 hidden units are applied for further feature fusion. The decoder is an attention-based sequence prediction model which automatically captures the information flow within the input sequence to predict the output sequence. It is based on an RNN and directly generates the target sequence $(y_1, y_2, ...)$. The details are as follows: at t-step, the decoder predicts an output y_t as

$$y_t = softmax(W_o s_t + b_o) \tag{4}$$

where s_t is the hidden state at time step t in the GRU, W_0 and b_0 are trainable parameters. State s_t is computed as

$$s_t = GRU(y_{t-1}, c_t, s_{t-1}) \tag{5}$$

where c_t represents the context vector,calculated as

$$c_t = \sum_{i=1}^{I} \alpha_{t,i} h_i \tag{6}$$

where $H = h_1, ..., h_I$ denotes the sequential feature vectors from the former encoder stage and I is the length of the feature maps. $\alpha_{t,i}$ is an attention weight and computed by

$$\alpha_{t,i} = \frac{exp(e_{t,i})}{\sum_{k=1}^{I} exp(e_{t,k})} \tag{7}$$

$$e_{t,k} = Tanh(W s_{t-1} + V h_i + b) \tag{8}$$

where W, V and b are trainable parameters.

The training is conducted by minimizing the objective function that negative log-likelihood of the conditional probability of word label. Let $D = \{X_i, Y_i\}$, $i = 1...N$ denotes the training set, the objective function $\mathcal{L}_{recog}$ in the recognition stage is calculated as

$$\mathcal{L}_{recog} = -\sum_{i=1}^{N} \sum_{t=1}^{|y_i|} \log p(Y_i|X_i) \tag{9}$$

3.4. End-to-end training and loss function

The network is able to jointly optimize the detector and recognizer with an end-to-end training strategy. Thus the total multi-task loss is defined as the combination of detection and recognition:

$$\mathcal{L} = \mathcal{L}_{detect} + \lambda_r \mathcal{L}_{recog} \tag{10}$$

where $\lambda_r > 0$ is the trade-off parameter. λ_r is set to 1 in our experiments.

4. Experiments

To validate the performance of the proposed model, we conduct experiments on several standard benchmarks and compare results with other state-of-the-art methods. We evaluate the performance of our model on a horizontal text set ICDAR2013, an oriented text set ICDAR2015, and a curved text set Total-Text.

4.1. Datasets

ICDAR-13: ICDAR 2013 [29] consists of 229 images for training and 233 images for testing. Text instances in ICDAR-13 dataset are mostly horizontal and annotated in words with word-level bounding box annotations and text labels.

ICDAR-15: The ICDAR 2015 Incidental Text dataset [30] includes 1000 training images and 500 testing images. Text in the dataset can be in arbitrary orientations, or suffer from motion blur and low resolution.

Total-Text: The Total-Text dataset [31] is a collection of irregular text with 1255 images in the training set, and 300 images in the test set and consists of a lot of curved text.

4.2. Implementation details

In the experiments, we adopt the ResNet-50 as the backbone of the network. The backbone is followed by the FPN to further enhance features. The detection branch outputs the predicted bounding boxes of possible texts for later sequential recognition. For each text region, its features of shape 16×50 are extracted from the shared convolutional features by RoIRotate and rectified by PSN. Then the rectified features are encoded by a CNN-LSTM based network and decoded by RNN-based attention, where the number of hidden units is set to 256. The total number of symbols is 94, which covers digits, upper and lower cases of English characters, numbers, and special characters. The network can be trained end-to-end using the standard error back-propagation and ADAM optimizer. The whole training process contains two stages: pre-trained on SynthText and fine-tuned on the real-world data. At the first step, the model is trained on the SynthText dataset [45] for 10 epochs. Then we further train the network on target datasets. We make use of similar data augmentation and online hard example mining (OHEM) as [7]. Data augmentation is a classical and necessary way to improve the robustness of deep neural networks, especially when the number of real data is limited. We observe that on the ICDAR2015 datasets without data augmentation the result denotes a 2.76% relative reduction in F-1 score from 88.19% to 85.43%.

As text recognition is very sensitive to the detection result, a small error in the predicted result of text detection could cut off several characters, which is harmful to the recognition branch training. We first train the detection and recognition branches using ground truth text regions instead of predicted text regions until it almost converges to a steady point, and then jointly train the whole network simultaneously in which the recognition branch is trained using the predicted text regions. This training procedure is effective to achieve the final convergence and both text detection and recognition can benefit from each other.

4.3. Experimental results on straight text

In this section, we perform experiments on ICDAR-15 and ICDAR-13 databases and compare our models with the existing methods. Note that we only use a single scale input. During inference, we resize the longer side of the input to 1280 for ICDAR-15 and 920 for ICDAR-13. The IoU (intersection-of-union) threshold is 0.5 as the default value to decide whether it is a true positive sample or not. The results are summarized in Tables 3 and 4. There are many methods evaluated on these two datasets, but only some of the best results are shown. Our proposed method achieves significant increase in performance when compared with the previous state-of-the-art works, which illustrates that our method can effectively deal with horizontal or oriented text.

(a) (b)

Fig. 3. Results of the proposed method. (a) ICDAR-15. (b) ICDAR-13.

In the detection only task, we achieve 90.53% F-measure on ICDAR-13 and 88.19% F-measure on ICDAR-15, respectively. Our method surpasses the best single scale model with text-instance-level annotations. Note that the CharNet [8] needs character-level annotations. For end-to-end performance, our method outperforms the highest single scale model.

Compared with the detection model without recognition branch, the end-to-end trained model of the proposed method can achieve significant F-measure improvement. It demonstrates that the joint trainable model with text recognition supervision branch can help improve the representation power of the features for text detection. As the text recognition supervision can force the model to consider fine details of characters and learn the difference among characters and background that have similar patterns, which makes our model be able to avoid some mistakes of detecting background regions as text or missing some text regions. Some qualitative results on ICDAR-13 and ICDAR-15 datasets are shown in Fig. 3.

4.4. Experimental results on curved text

We conducted an experiment on the curved text dataset called Total-Text and compare our models with the existing methods. The evaluation protocol for detection is based on [46], the one for end-to-end recognition is based on the end-to-end evaluation protocol of ICDAR15. At inference time, the shorter side of each image is resized to 600 pixels. We compare the results of our model with previous work in Table 5. Our method outperforms the previous works by a large margin in end-to-end evaluations. In the end-to-end recognition task, our model surpasses the previous highest by 5.8%. Our method also achieves better performance in the detection task compared with other end-to-end methods. However, there is still a big gap in the performance of curved text detection between the proposed end-to-end method and methods specially designed for curved text detection.

4.5. Ablation study

In this part, we conduct experiments to verify the effect of our architecture. We do comparative experiments to show the text detection and end-to-end recognition performance on the perspective dataset (ICDAR-15) and irregular dataset (Total-Text). As the detection branch in the proposed method is not specially designed for

Table 3
Results on ICDAR-15 test set. "P", "R", "F" represent "Precision", "Recall", "F-measure" respectively. "S", "W", "G" represent recognition with "Strong", "Weak", "Generic" lexicon respectively. R-50 represents the backbone network using ResNet-50.

Method	Detection			Method	End-to-End Recognition		
	R	P	F		S	W	G
SegLink [32]	76.8	73.1	75.0	TextSpotter [33]	54.0	51.0	47.0
EAST [34]	78.33	83.27	80.72	TextProposals+DictNe [35]	53.30	49.61	47.18
RRPN [36]	77.13	83.52	80.20	HUST_ MCLAB[18,32]	67.9	–	–
PixelLink [1]	82.0	85.5	83.7	He et al. [37]	82.0	77.0	63.0
Mask TextSpotter [24]	81.0	91.60	86.0	Mask TextSpotter [24]	79.30	73.0	62.40
TextNet [9]	85.41	89.42	87.37	TextNet [9]	78.66	74.90	60.45
FOTS(R-50) [7]	85.17	91.0	87.99	FOTS(R-50) [7]	81.09	75.90	60.80
CharNet(R-50) [8]	**88.30**	**91.15**	**89.70**	CharNet(R-50) [8]	80.14	74.45	62.18
Ours(R-50)	85.53	91.02	88.19	Ours(R-50)	**83.04**	**76.41**	**63.85**

Table 4
Results on ICDAR-13 test set under ICDAR-13 evaluation.

Method	Detection			Method	End-to-End Recognition		
	R	P	F		S	W	G
TextFlow [38]	75.89	85.15	80.25	Deep2Text II+ [39]	81.81	79.47	76.99
SSTD [40]	86.0	88.0	87.0	StradVision-1 [29]	81.28	78.51	67.15
TextEdge [15]	84.13	91.85	87.82	Li et al. [39]	91.08	89.81	84.59
TextNet [9]	**89.39**	**93.26**	**91.28**	TextNet [9]	89.77	88.80	82.96
FOTS(R-50)[7]	–	–	88.23	FOTS(R-50) [7]	88.81	87.11	80.81
Ours(R-50)	88.51	92.64	90.53	Ours (R-50)	**92.14**	**90.85**	**85.03**

Table 5
Results on Total-Text. No lexicon is used in end-to-end evaluation.

Type	Detection			E2E
	R	P	F	
DeconvNet [31]	33.0	40.0	36.0	–
Textboxes [14]	45.5	62.1	52.5	36.3
TextSnake [41]	74.5	82.72	78.4	–
MSR [42]	73.0	**85.2**	78.6	–
TextField [43]	**79.9**	81.2	80.6	–
FTSN [44]	78.0	84.7	**81.3**	–
Mask TextSpotter[24]	55.0	69.0	61.3	52.9
TextNet [9]	59.45	68.21	63.53	54.0
Ours(R-50)	61.95	68.44	65.03	**59.80**

Table 6
The end-to-end performance comparisons on ICDAR-15 and Total-Text. "DFS" is the abbreviation of "Different Features Strategy". The E2E results on ICDAR-15 is tested with generic lexicon.

Dataset	Type	Detection			E2E
		R	P	F	
ICDAR 2015	Baseline	84.52	88.65	86.54	60.43
	With PSN	85.26	89.10	87.14	61.27
	With DFS	85.46	89.64	87.50	62.90
	ALL	85.53	91.02	88.19	63.85
TotalText	Baseline	56.82	65.72	60.95	53.18
	With PSN	57.13	66.92	61.64	56.12
	With DFS	58.40	67.15	62.47	57.74
	ALL	61.95	68.44	65.03	59.80

irregular or curved text, the performance is inferior to those methods which need specially designed modular or character level annotations. Our focus is on the behaviors of the detection and recognition branch in our framework and experiments are done to verify the effectiveness of the structure we designed.

Impact of the Position Sensitive Network In this section, we study how Position Sensitive Network(PSN) affects the performance of the recognizer by training a separate network without PSN. Table 6 shows the effect of using PSN on benchmark datasets. Without PSN, the performance drops on all the datasets, especially on the curved dataset (TotalText). This implies that the PSN improves the performance of the recognizer when dealing with irregular texts. This is principally because the PSN is able to rectify features of text to the regular ones, particularly for the irregular text, which decreases the difficulty of recognition and enables the sequence recognition network to read irregular text more easily.

Strategy to deal with the Features Incompatibility In our framework, different scale features are used for detection and recognition respectively in order to deal with the feature incompatibility problem. To demonstrate the effectiveness of our strategy to deal with the features incompatibility, we also train a baseline network with the same shared features. As shown in Table 6, our strategy can significantly improve the end-to-end results. The improvement proves that using different layer features of the convolutional neural network can reduce the contradiction between detection and recognition tasks in feature extraction and a larger-scale feature is more effective for the recognition part to improve the accuracy instead of using the same feature with the detection.

4.6. The speed and model size

To evaluate the speed of our model, we calculate the average time cost during the testing stage. For images from the ICDAR-15 dataset (with resolution 1280×720), the end-to-end inference time is 387ms on a single GeForce GTX 1080ti GPU with the ResNet-50 backbone. The corresponding inference time is 274ms if only run the detection branch and 68ms for the recognition branch. The rest of the time is spent in the RoIRotate and PSN, which is about 45ms.

The total number of parameters of the proposed end-to-end method is about 34M. The backbone network ResNet-50 includes 23M parameters taking the most of parameters in the proposed model. By sharing the backbone network, the proposed method can not only reduces the time cost during predicting stage but also save almost half of parameters compared with two-stage system, in which text detection and recognition models are trained separately. Thus, for scene text images, the computational cost of the recognition branch is reduced. Sharing the same CNN feature

extractor makes the end-to-end model more computationally efficient than two-step methods.

5. Conclusion

In this paper, we propose a novel framework for scene text localization and recognition scene text. The model is trained for both text detection and recognition in a single training framework. Different from the existing methods in which the text detection branch and recognition branch using the same size convolutional features in the backbone architecture, we combine larger scale features in the recognition part to ensure the detailed information needed for recognition, which further improves the accuracy of recognition. To extract effective text features for perspective and curved text recognition, the Position Sensitive Network is introduced to rectify the text proposal features in the recognition branch. Experiments on standard benchmarks have demonstrated the effectiveness of our method.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is supported by the Key Programs of the Chinese Academy of Sciences under Grant No. ZDBS-SSW-JSC004, and No. ZDBS-SSWJSC005, and the National Natural Science Foundation of China (NSFC) under Grant No. 62073326 and No. 62071469.

References

[1] D. Deng, H. Liu, D. Cai, X. Li, Pixellink: detecting scene text via instance segmentation, in: AAAI-18 AAAI Conference on Artificial Intelligence, 2018, pp. 6773–6780.
[2] Y. Baek, B. Lee, D. Han, S. Yun, H. Lee, Character region awareness for text detection, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 9365–9374.
[3] W. Wang, E. Xie, X. Li, W. Hou, T. Lu, G. Yu, S. Shao, Shape robust text detection with progressive scale expansion network, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 9336–9345.
[4] M. Liao, Z. Wan, C. Yao, K. Chen, X. Bai, Real-time scene text detection with differentiable binarization, in: AAAI 2020 : The Thirty-Fourth AAAI Conference on Artificial Intelligence, 34, 2020, pp. 11474–11481.
[5] F. Zhan, S. Lu, Esir: end-to-end scene text recognition via iterative image rectification, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 2059–2068.
[6] B. Shi, M. Yang, X. Wang, P. Lyu, C. Yao, X. Bai, Aster: an attentional scene text recognizer with flexible rectification, IEEE Trans. Pattern Anal. Mach.Intell. 41 (9) (2019) 2035–2048.
[7] X. Liu, D. Liang, S. Yan, D. Chen, Y. Qiao, J. Yan, Fots: fast oriented text spotting with a unified network, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 5676–5685.
[8] L. Xing, Z. Tian, W. Huang, M. Scott, Convolutional character networks, in: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 9126–9136.
[9] Y. Sun, C. Zhang, Z. Huang, J. Liu, J. Han, E. Ding, Textnet: irregular text reading from images with an end-to-end trainable network, in: Asian Conference on Computer Vision, 2018, pp. 83–99.
[10] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: towards real-time object detection with region proposal networks, IEEE Trans. Pattern Anal. Mach.Intell. 39 (6) (2017) 1137–1149.
[11] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S.E. Reed, C.-Y. Fu, A.C. Berg, SSD: single shot multibox detector, Eur. Conf. Comput. Vis. (2016) 21–37.
[12] C. Du, Y. Wang, C. Wang, C. Shi, B. Xiao, Selective feature connection mechanism: concatenating multi-layer CNN features with a feature selector, Pattern Recognit. Lett. 129 (2020) 108–114.
[13] Z. Tian, W. Huang, T. He, P. He, Y. Qiao, Detecting text in natural image with connectionist text proposal network, in: European Conference on Computer Vision, 2016, pp. 56–72.
[14] M. Liao, B. Shi, X. Bai, X. Wang, W. Liu, Textboxes: a fast text detector with a single deep neural network, in: AAAI'17 Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, 2017, pp. 4161–4167.
[15] C. Du, C. Wang, Y. Wang, Z. Feng, J. Zhang, Textedge: multi-oriented scene text detection via region segmentation and edge classification, in: 2019 International Conference on Document Analysis and Recognition (ICDAR), 2019, pp. 375–380.
[16] J. Dai, Y. Li, K. He, J. Sun, R-FCN: object detection via region-based fully convolutional networks, in: Proceedings of the 30th International Conference on Neural Information Processing Systems, 2016, pp. 379–387.
[17] F. Sheng, Z. Chen, W. Zhang, B. Xu, Pyrboxes: an efficient multi-scale scene text detector with feature pyramids, Pattern Recognit. Lett. 125 (2019) 228–234.
[18] B. Shi, X. Bai, 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 (11) (2017) 2298–2304.
[19] C.-Y. Lee, S. Osindero, Recursive recurrent nets with attention modeling for OCR in the wild, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 2231–2239.
[20] S.K. Ghosh, E. Valveny, A.D. Bagdanov, Visual attention models for scene text recognition, in: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), 2017, pp. 943–948.
[21] Z. Cheng, F. Bai, Y. Xu, G. Zheng, S. Pu, S. Zhou, Focusing attention: towards accurate text recognition in natural images, in: 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 5086–5094.
[22] J. Zhao, Y. Wang, B. Xiao, C. Shi, J. Jiang, C. Wang, Adversarial learning based attentional scene text recognizer, Pattern Recognit. Lett. 138 (2020) 217–222.
[23] J. Zhang, C. Luo, L. Jin, T. Wang, Z. Li, W. Zhou, Sahan: scale-aware hierarchical attention network for scene text recognition, Pattern Recognit. Lett. 136 (2020) 205–211.
[24] P. Lyu, M. Liao, C. Yao, W. Wu, X. Bai, Mask textspotter: an end-to-end trainable neural network for spotting text with arbitrary shapes, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 71–88.
[25] S. Qin, A. Bissaco, M. Raptis, Y. Fujii, Y. Xiao, Towards unconstrained end-to-end text spotting, in: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 4704–4714.
[26] K. He, G. Gkioxari, P. Dollar, R. Girshick, Mask R-CNN, IEEE Trans. Pattern Anal. Mach.Intell. 42 (2) (2020) 386–397.
[27] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.
[28] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 936–944.
[29] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L.G. i Bigorda, S.R. Mestre, J. Mas, D.F. Mota, J.A. Almazan, L.-P. de las Heras, Icdar 2013 robust reading competition, in: 2013 12th International Conference on Document Analysis and Recognition, 2013, pp. 1484–1493.
[30] 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, E. Valveny, Icdar 2015 competition on robust reading, in: 2015 13th International Conference on Document Analysis and Recognition (ICDAR), 2015, pp. 1156–1160.
[31] C.K. Chng, C.S. Chan, Total-text: a comprehensive dataset for scene text detection and recognition, in: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), 01, 2017, pp. 935–942.
[32] B. Shi, X. Bai, S. Belongie, Detecting oriented text in natural images by linking segments, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 3482–3490.
[33] M. Busta, L. Neumann, J. Matas, Deep textspotter: an end-to-end trainable scene text localization and recognition framework, in: 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2223–2231.
[34] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, J. Liang, East: an efficient and accurate scene text detector, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 2642–2651.
[35] L. Gomez, D. Karatzas, Textproposals: a text-specific selective search algorithm for word spotting in the wild, Pattern Recognit. 70 (2017) 60–74.
[36] J. Ma, W. Shao, H. Ye, L. Wang, H. Wang, Y. Zheng, X. Xue, Arbitrary-oriented scene text detection via rotation proposals, IEEE Trans. Multimed. 20 (11) (2018) 3111–3122.
[37] T. He, Z. Tian, W. Huang, C. Shen, Y. Qiao, C. Sun, An end-to-end textspotter with explicit alignment and attention, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 5020–5029.
[38] S. Tian, Y. Pan, C. Huang, S. Lu, K. Yu, C. Lim Tan, Text flow: A unified text detection system in natural scene images, in: Proceedings of the IEEE international conference on computer vision, 2015, pp. 4651–4659.
[39] H. Li, P. Wang, C. Shen, Towards end-to-end text spotting with convolutional recurrent neural networks, in: 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 5248–5256.
[40] P. He, W. Huang, T. He, Q. Zhu, Y. Qiao, X. Li, Single shot text detector with regional attention, in: 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 3066–3074.
[41] S. Long, J. Ruan, W. Zhang, X. He, W. Wu, C. Yao, Textsnake: A flexible representation for detecting text of arbitrary shapes, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 20–36.
[42] C. Xue, S. Lu, W. Zhang, Msr: multi-scale shape regression for scene text detection, in: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, 2019, pp. 989–995.
[43] Y. Xu, Y. Wang, W. Zhou, Y. Wang, Z. Yang, X. Bai, Textfield: learning a deep

direction field for irregular scene text detection, IEEE Trans. Image Process. 28 (11) (2019) 5566–5579.
[44] Y. Dai, Z. Huang, Y. Gao, Y. Xu, K. Chen, J. Guo, W. Qiu, Fused text segmentation networks for multi-oriented scene text detection, in: 2018 24th International Conference on Pattern Recognition (ICPR), 2018, pp. 3604–3609.
[45] A. Gupta, A. Vedaldi, A. Zisserman, Synthetic data for text localisation in natural images, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 2315–2324.
[46] C. Wolf, J.-M. Jolion, Object count/area graphs for the evaluation of object detection and segmentation algorithms, Int. J. Docum. Anal. Recognit. 8 (4) (2006) 280–296.