SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •
Special Focus on Deep Learning for Computer Vision

February 2020, Vol. 63 120101:1–120101:14
https://doi.org/10.1007/s11432-019-2710-7

# Progressive rectification network for irregular text recognition

Yunze GAO[1,2], Yingying CHEN[1*], Jinqiao WANG[1] & Hanqing LU[1]

[1]*National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing* 100190*, China;*
[2]*University of Chinese Academy of Sciences, Beijing* 100049*, China*

Received 26 August 2019/Accepted 10 October 2019/Published online 14 January 2020

**Abstract** Scene text recognition has received increasing attention in the research community. Text in the wild often possesses irregular arrangements, which typically include perspective, curved, and oriented texts. Most of the existing methods do not work well for irregular text, especially for severely distorted text. In this paper, we propose a novel progressive rectification network (PRN) for irregular scene text recognition. Our PRN progressively rectifies the irregular text to a front-horizontal view and further boosts the recognition performance. The distortions are removed step by step by leveraging the observation that the intermediate rectified result provides good guidance for subsequent higher quality rectification. Additionally, by decomposing the rectification process into multiple procedures, the difficulty of each step is considerably mitigated. First, we specifically perform a rough rectification, and then adopt iterative refinement to gradually achieve optimal rectification. Additionally, to avoid the boundary damage problem in direct iterations, we design an envelope-refinement structure to maintain the integrity of the text during the iterative process. Instead of the rectified images, the text line envelope is tracked and continually refined, which implicitly models the transformation information. Then, the original input image is consistently utilized for transformation based on the refined envelope. In this manner, the original character information is preserved until the final transformation. These designs lead to optimal rectification to boost the performance of succeeding recognition. Extensive experiments on eight challenging datasets demonstrate the superiority of our method, especially on irregular benchmarks.

**Keywords** irregular text recognition, progressive rectification, iterative refinement

**Citation** Gao Y Z, Chen Y Y, Wang J Q, et al. Progressive rectification network for irregular text recognition. Sci China Inf Sci, 2020, 63(2): 120101, https://doi.org/10.1007/s11432-019-2710-7

## 1 Introduction

Scene text recognition has drawn remarkable attention in computer vision because of its importance in various real-world applications such as scene understanding, card information entry, and street sign reading. Benefiting from recent advancements in deep learning, reading text in natural images has experienced a rapid evolution during the past few years. In spite of considerable advances, scene text recognition in unconstrained conditions still remains a challenging problem because of complex situations such as blurring, distortion, orientation, and uneven lighting.

Irregular text frequently appears in natural scenes, because of curved character placement, perspective distortion, etc. Recognizing text with arbitrary shape is an extremely difficult task because of unpredictable changes in text layouts. Most existing approaches [1–4] mainly focus on regular text recognition;

* Corresponding author (email: yingying.chen@nlpr.ia.ac.cn)

 

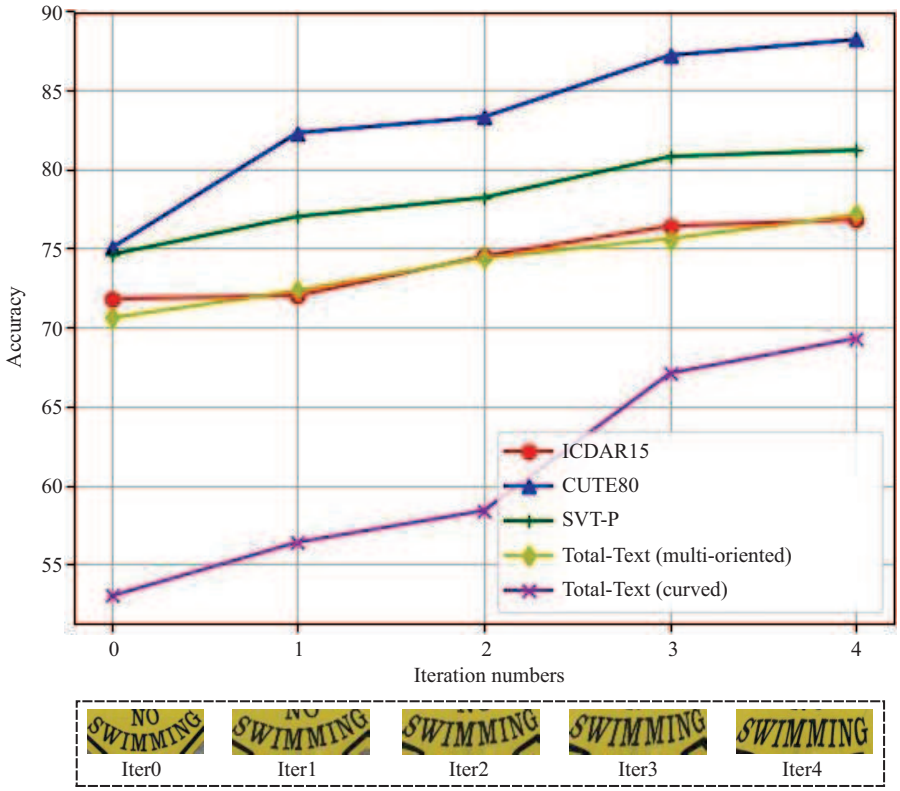


**Figure 1** (Color online) We propose a novel progressive rectification network that progressively rectifies irregular text to a front-horizontal view, leading to optimal rectification and easier recognition. As our experiments will demonstrate, the proposed method enables accurate rectification and considerably improves the performance of challenging irregular text benchmarks. Considering the tradeoff between accuracy and speed, we select three iterations.

however, applying these approaches to identify distorted text is difficult. One idea is to rectify irregular text into an easy-to-recognize front-horizontal view [5–7]. The spatial transformer network (ST-Net) [8] is a learnable module that explicitly allows spatial manipulations. However, addressing complex variations, particularly non-rigid ones, with ST-Net is difficult. In real-world human cognitive processes, difficult tasks are usually broken down into multiple simple instructions to be executed, and phased results are used to guide the next implementation process. Moreover, humans generally adopt multiple refinements to better accomplish complex tasks. For example, when we hang a picture on a wall, we usually first locate its approximate position and then gradually adjust it to the horizontal position.

Motivated by such human cognitive behaviors, we have designed a progressive rectification network (PRN) to progressively rectify irregular text towards the front-horizontal view and achieve optimal recognition, as shown in Figure 1. The challenging rectification task is divided into multiple easier sub-tasks in a progressive manner. We first perform a rough rectification, and then adopt continuous refinements to gradually achieve accurate rectification. In each iteration, the residual between the previous and current geometric transformation fields is estimated based on the previously rectified image to get one step closer to the optimal one. In this way, the difficulty of each step is intrinsically mitigated and severe distortions can be eliminated in a progressive manner. Therefore, a series of rectification processes adapted to increasingly accurate rectification can effectively improve robustness of the model to large variations of text.

However, we observe that direct iterations may lead to irreversible information loss. As illustrated in the first line of Figure 2, although the next rectification adjusts the text in a more ideal direction, the missing character information cannot be recovered, thereby introducing the boundary damage. To address this dilemma, we develop an envelope-refinement structure to maintain the integrity of text during the iterative process. Specifically, we estimate the envelope of the text region that reliably represents the

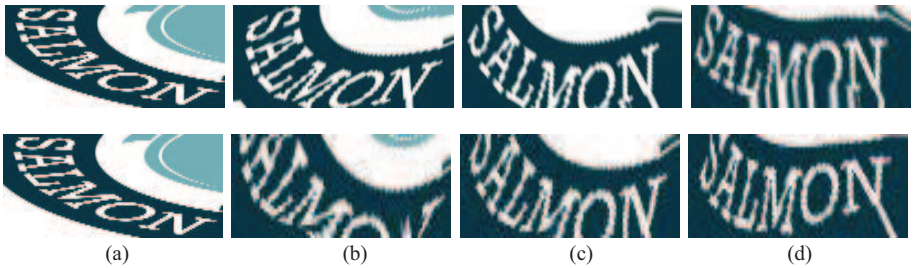


**Figure 2** (Color online) The comparison of two iterative methods. The top and bottom rows are the direct iterations and envelope-refinement structures, respectively. (a) is the original input images; (b)–(d) are the rectified images in the first iteration, in the second iteration, and in the last iteration, respectively. The direct iteration structure discards the information outside the rectified images and leads to boundary damage, while the envelope-refinement structure can recover the missing information and preserve the intact structure of characters.

position and posture of the text line. The transformation information is preserved and transmitted by the envelope during multiple iterations. The intermediate results only provide guidance for the estimation of the envelope, and the original input image is consistently used for spatial transformation based on the refined envelope. In this manner, original textual information can be integrally preserved, and the character structure will not be destroyed, as shown in the second line of Figure 2. Furthermore, the rectification network is jointly optimized with the recognition network under the same objective in an end-to-end scheme. Therefore, our PRN can adaptively learn the optimal transformation for the following recognition task.

The main contributions are summarized below.

(1) We have proposed a novel PRN to rectify irregular text towards the front-horizontal view and achieve optimal recognition in a progressive manner.

(2) We have designed an envelope-refinement structure to maintain the integrity of text, which avoids the boundary damage problem in the scenario of iterative rectification.

(3) The proposed PRN is an end-to-end trainable system that does not require any extra character-level annotations. The system is robust to various text variations and achieves superior performance compared with state-of-the-art methods on challenging datasets, especially on irregular benchmarks.

## 2 Related work

Scene text recognition has been widely researched and numerous methods have recently been proposed. Traditional methods, which recognize scene text in a character-level manner, first perform detection to generate multiple candidates of character locations, and then apply a character classifier for recognition. Wang et al. [9] detected each character by a sliding window and recognized them with a character classifier trained on HOG descriptors. Bissacco et al. [10] designed a fully connected network to extract character feature representations and then used a language model to recognize characters. However, the performance of these methods is limited because of inaccurate character detection. To be free from this problem, some methods directly learn the mapping between entire word images and target strings. For example, Jaderberg et al. [11] assigned a class label to each word in a pre-defined lexicon and performed a 90k-class classification with convolutional neural network (CNN). Rodriguez-Serrano et al. [12] formulated scene text recognition as a retrieval problem that embedded word labels and word images into a common Euclidean space and found the closest word label in this space.

By successfully applying the recurrent neural network (RNN) in sequence recognition, some researchers [1–3,5] developed sequence-based methods and combined CNN and RNN to encode feature representations of word images. Shi et al. [1] and He et al. [2] used the connectionist temporal classification (CTC) [13] loss to calculate the conditional probabilities between the outputs of RNN and target sequences. Later, Shi et al. [5] and Lee et al. [3] introduced an attention mechanism to adaptively weight the features and select the most relevant feature representations in an RNN-based decoder. To eliminate the attention

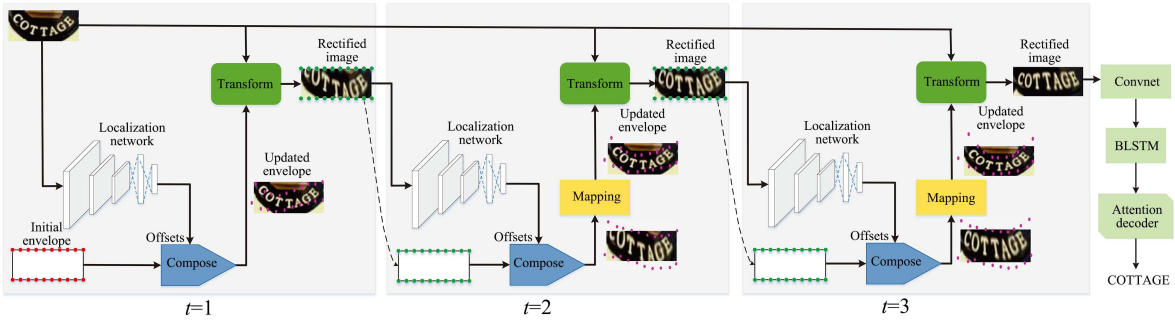


**Figure 3** (Color online) Overview of our PRN for irregular text recognition. The dotted lines represent the coordinates delivery.

drift problem, Cheng et al. [4] employed a focusing attention mechanism to automatically adjust the attention weights. Bai et al. [14] proposed the edit probability to estimate the probability of generating a string while considering possible occurrences of missing or superfluous characters. Fang et al. [15] recognized characters considering visual cues and linguistic rules. Although these approaches have shown promising results, they cannot effectively handle irregular text. The main reason is that word images are encoded into 1D feature sequences, but irregular text is not horizontally arranged.

Research on irregular text recognition has been relatively less. An early attempt [16] adopted scale invariant feature transform (SIFT) in a bag-of-keypoints approach, which was robust to perspective distortions. With the development of deep neural networks, recent studies have explored deep neural architectures and achieved better results. Yang et al. [17] exploited a character detector to learn text-specific features and extended a typical attention-based encoder-decoder by introducing a 2D attention mechanism. Liu et al. [18] and Shi et al. [6] calibrated the distorted text through simple local/global transformation. Cheng et al. [19] proposed to combine textual features in four directions. Zhan and Lu [20] designed a fix-order polynomial to represent irregular text orientation. Yang et al. [7] added symmetrical constraints in the rectification module, which relied on character-level annotations.

Our approach is different from existing methods in the sense that, we explore the rectification of irregular text in a progressive manner that is tolerant of various distortions and dramatically improves recognition performance. The rectification process is decomposed into multiple steps and the rectified results are iteratively refined. Different steps work together to eliminate text distortions for better recognition; thus, the difficulty of each step is considerably mitigated and large distortions can be effectively removed. Moreover, we design an envelope-refinement structure to continually refine the text line envelope, which avoids the boundary damage phenomenon in multiple rectifications. The model is trained end-to-end and does not require character-level annotations. Because our method has the ability to recognize irregular text, it can reduce the pressure of text detection [21].

## 3 Proposed approach

An overview of our PRN for irregular text recognition is shown in Figure 3. Irregular text is progressively rectified towards front-horizontal view, which is fed to the subsequent recognition network. During the rectification process, distortions are eliminated step by step. We use the same rectification network to recurrently update the rectified result based on the previous result. In this manner, the recurrent framework can iteratively refine the spatial transformation without any extra parameters. Moreover, the envelope-refinement structure continuously refines the text line envelope and consistently utilizes the original image for spatial transformation. This design effectively avoids the boundary damage problem and maintains the intact structures of characters. Our PRN can deal with distorted text with various orientations and shapes, including severely distorted text.

### 3.1 Progressive rectification

Generally, single-step rectification often fails to fully remove geometrical distortions and can lead to text content loss, thereby creating undesirable effects on the following recognition. Based on this observation, we decompose the rectification process into multiple progressive steps, which considerably mitigates the difficulty of each step. First, a rough rectification is performed and then multiple refinements are conducted to gradually achieve optimal rectification. In each iteration, the rectified result is further adjusted by re-feeding it into the pipeline, thereby forming a recurrent structure. The rectified image generated by the proposed rectification process provides good guidance for the next higher quality rectification. Denoting the transformation parameters estimation as $E$, and the spatial transformation operation as $S$, we have the following structure:

$$\Theta_t = E(I_{t-1}), \tag{1}$$

$$I_t = S(I_{t-1}, \Theta_t), \tag{2}$$

where $t$ represents the $t$-th iteration and $I_0$ is the original input image. In this manner, large variations can be progressively eliminated to best match the succeeding recognition task.

At the $t$-th transformation, we predict the envelope of the text region and calculate the thin plate spline transformation [22] parameters based on the envelope. Specifically, the envelope is determined by the border points of the text line, which flexibly reflect the position and posture of the text. We define a canonical envelope $B = [b_1, \ldots, b_N]^{\mathrm{T}}$ on a rectified image to describe the desired front-horizontal view:

$$b_i = \begin{cases} \left[\frac{2i}{N-2}, 0\right]^{\mathrm{T}}, & 0 \leqslant i < \frac{N}{2}, \\ \left[\frac{2i-N}{N-2}, 1\right]^{\mathrm{T}}, & \frac{N}{2} \leqslant i < N, \end{cases} \tag{3}$$

where $N$ is the number of border points. In the feed-forward process, the localization network predicts the envelope $B'_t$ on the input image. Then, the transformation parameters can be formulated as follows:

$$\Theta_t = \begin{bmatrix} 0 & 0 & 1_{1\times N} \\ 0 & 0 & B^{\mathrm{T}} \\ 1_{N\times 1} & B & D \end{bmatrix}^{-1} \cdot \begin{bmatrix} 0_{3\times 2} \\ B'_t \end{bmatrix}, \tag{4}$$

in which $D$ is an $N \times N$ square matrix, and $D_{ij} = \|b_i - b_j\|^2 \log(\|b_i - b_j\|)$. Given a random point $R = [R_x \ R_y]$, its corresponding point $R'$ can be obtained by linear projection $M$:

$$R' = [1 \quad R \quad \tilde{D}] \cdot \Theta_t, \tag{5}$$

where $\tilde{D}$ is an $N$-dimensional row vector, and $\tilde{D}_i = \|R - b_i\|^2 \log(\|R - b_i\|)$. In the transform module, a grid $\mathcal{G}$ on the input image is generated by iterating over all the points on the rectified image, and then the pixel values of the rectified image can be obtained by bilinear interpolation based on $\mathcal{G}$.

### 3.2 Envelope refinement

As observed in Figure 2, iterative refinement can gradually rectify the irregular text to be more suitable for recognition; however, the missing information caused by inaccurate transformation cannot be restored in the direct iteration structure. This leads to the boundary damage phenomenon and thus, results in recognition errors. We analyze that the output image samples only from the previously rectified image and pixel information outside the region are discarded. In the top row of Figure 2, this effect is visible while requiring pixel information outside the previously rectified image. In the scenario of iterative rectification, the effects of missing information are accumulated during multiple transformations.

To remedy the boundary damage problem, we designed an envelope-refinement structure to maintain the intact structures of characters in the iterative process. In particular, motivated by [23], we advocate that the transformation information should be transmitted through the envelope rather than be discarded

after transformation. Our approach is different from composing the transformation parameters in [23] in the sense that the envelope of the text region is tracked and continually refined during multiple iterations. Then, we estimate the transformation parameters based on the refined envelope and consistently sample from the original image at each step. In this way, the original character information is preserved until the final transformation. To sample from the original image, we need to map the predicted envelope to the original image. Especially at the first step, the input is the original image; thus, mapping can be omitted. Note that the envelope on the original image is mapped again to the canonical form on the rectified image after each transformation (see the green points in Figure 3). In addition, to facilitate network training, the localization network only predicts the envelope offset. Optimizing the residual values is easier than optimizing the original ones. In the $t$-th recursion, assume that the offset of the envelope is denoted as $O_t$, and the envelopes on the original and rectified images are denoted as $B_t'^{\text{ori}}$ and $B_t'^{\text{rec}}$, respectively. Then the envelope is updated as follows:

$$B_t'^{\text{ori}} = \begin{cases} B'_{\text{initial}} + O_t, & t = 1, \\ M(B_t'^{\text{rec}}, \Theta_{t-1}), & t > 1, \end{cases} \tag{6}$$

$$B_t'^{\text{rec}} = B + O_t, \quad t > 1, \tag{7}$$

$$O_t = L(I_{t-1}), \tag{8}$$

where $I_{t-1}$ $(t > 1)$ is the output rectified image of the previous step, $I_0$ is the original image, $L$ is the localization network, $B'_{\text{initial}}$ is the pre-defined initial envelope, and the mapping operation $M$ is the same as the definition in (5). While the envelope falls outside the rectified image, mapping back to the original image indicated compensating some missing information.

Based on the updated envelope $B_t'^{\text{ori}}$, the transformation parameters $\Theta_t$ can be estimated as in (4), and the next rectified image can be sampled from the original image $I_0$. Thus, Eq. (2) is rewritten as follows:

$$I_t = S(I_0, \Theta_t). \tag{9}$$

In this way, the integrity of the text is effectively maintained because pixel information outside the rectified image is also preserved until the final transformation. Therefore, our network can progressively rectify irregular text without boundary damage. In addition, all the modules of the rectification process are differentiable, thereby allowing for backpropagation within an end-to-end learning framework. Moreover, the rectification process focuses on the text region, which implicitly models the attention mechanism. Localizing the text region accurately not only achieves satisfactory rectification but also effectively removes background noise.

### 3.3 Recognition network

For the recognition network, we adopt the attention-based encoder-decoder pipeline. First, the encoder extracts a sequence of feature vectors $H = (h_1, \ldots, h_k)$ through the CNN-LSTM structure. Then, the attention decoder recurrently generates the character sequence $y = (y_1, \ldots, y_m)$. At step $i$, the decoder dynamically weights the image feature and selects the most relevant content to generate the probability distribution. Given the previous RNN hidden state $s_{i-1}$ and feature sequence $H$, the attention weights can be obtained by separately scoring each element in the feature sequence:

$$e_{i,j} = v^{\mathrm{T}} \tanh(W s_{i-1} + U h_j + b), \tag{10}$$

$$\alpha_{i,j} = \frac{\exp(e_{i,j})}{\sum_{j=1}^{k} \exp(e_{i,j})}. \tag{11}$$

We can then obtain the weighted sum of sequential feature vectors, which focuses on the most relevant features:

$$g_i = \sum_{j=1}^{k} \alpha_{i,j} h_j. \tag{12}$$

Then, the RNN hidden state is updated and the probability distribution is estimated as follows:

$$s_i = \mathrm{RNN}(y_{i-1}, s_{i-1}, g_i), \tag{13}$$

$$p_i = \mathrm{softmax}(V^{\mathrm{T}} s_i). \tag{14}$$

$W$, $U$, $V$, $v$, and $b$, which are mentioned above, are the learnable parameters. In addition, we exploit a bidirectional decoder that comprises two decoders in opposite directions.

### 3.4 Model training

For the given input image $I$ and corresponding ground truth $\hat{y} = (\hat{y_1}, \hat{y_2}, \ldots, \hat{y_m})$, the objective function is formulated as follows:

$$\mathcal{L} = -\frac{1}{2} \sum_{i=1}^{m} \ln(p_f(\hat{y}_i|I) \cdot p_b(\hat{y}_i|I)), \tag{15}$$

where $p_f$ and $p_b$ are the output probability distributions of the forward and backward decoders, respectively.

Note that the rectification network is optimized under the guidance of the recognition network. Therefore, our network is an end-to-end system trained on pairs of images and words, thereby eliminating the tedious labeling procedure of extra character-level annotations. Thus, the rectification network is also encouraged to transform irregular text to best match the succeeding recognition network. Moreover, with the flexible and accurate rectification network, irregular text is progressively rectified to a front-horizontal view, which leads to easier recognition and further benefits the training of the recognition network.

The initial envelope is defined as the same form as the canonical one, and the last fully-connected layer of the localization network is initialized, so that the predicted offset is zero at the beginning of training. In this manner, the scene text remains as its original appearance in the beginning phase, and the network gradually learns to transform the text towards the easy-to-recognize form. This operation avoids randomly distorted text disturbing the training process, which guarantees stable network convergence. We also adopt the MSRA weight initialization method [24] for the convolutional and other fully-connected layers, and the orthogonal weight initialization method [25] is used for the LSTM layers.

## 4 Experiments

In this section, we describe the details of experimental settings and evaluate the effectiveness of our method. We compare the performance of our PRN with other approaches on both regular and irregular datasets. Furthermore, we also conduct experiments to explore the effects of iteration numbers and the proposed envelope-refinement structure.

### 4.1 Datasets

• Synth90k is a synthetic dataset released by Jaderberg et al. [26], which contains approximately eight million images. All the images are generated by a synthetic text engine based on a set of 90000 common English words.

• SynthText is a synthetic dataset released by Gupta et al. [27] that was created for scene text detection. The word patches are cropped according to the ground truth bounding boxes, leading to seven million images.

• Street view text perspective (SVTP) [16] contains 639 word images captured from side-view angles in Google street view. Most images suffer from severe perspective distortion. Each image is specified with a 50 word lexicon and a full lexicon.

• CUTE80 [28] contains 288 word images collected to evaluate the performance of curved text recognition. Note that no lexicon is provided.

**Table 1** Architecture of our progressive rectification network

| | Layer name | Configurations |
|---|---|---|
| Rectification network | Block1 | conv 3×3, 32; pool 2×2 |
| | Block2 | conv 3×3, 64; pool 2×2 |
| | Block3 | conv 3×3, 128; pool 2×2 |
| | Block4 | conv 3×3, 256; pool 2×2 |
| | Block5 | conv 3×3, 256; pool 2×2 |
| | Block6 | conv 3×3, 256 |
| | Block7 | fc1, 512; fc2, 40 |
| Recognition network | Convolution | 3×3, 32 |
| | Residual Unit1_X | $\begin{bmatrix} 1\times 1,\ 32 \\ 3\times 3,\ 32 \end{bmatrix} \times 3$ |
| | Residual Unit2_X | $\begin{bmatrix} 1\times 1,\ 64 \\ 3\times 3,\ 64 \end{bmatrix} \times 4$ |
| | Residual Unit3_X | $\begin{bmatrix} 1\times 1,\ 128 \\ 3\times 3,\ 128 \end{bmatrix} \times 6$ |
| | Residual Unit4_X | $\begin{bmatrix} 1\times 1,\ 256 \\ 3\times 3,\ 256 \end{bmatrix} \times 6$ |
| | Residual Unit5_X | $\begin{bmatrix} 1\times 1,\ 512 \\ 3\times 3,\ 512 \end{bmatrix} \times 3$ |
| | BLSTM 1 | 256 hidden units per LSTM |
| | BLSTM 2 | 256 hidden units per LSTM |
| | LSTM | 256 hidden units |
| | LSTM | 256 hidden units |

• ICDAR 2015 (IC15) [29] contains 2077 word images including considerable irregular text obtained from Google glasses. For fair comparison, we discard the images that contain non-alphanumeric characters. Note that no lexicon is specified.

• Street View Text (SVT) [9] contains 647 images collected from Google street view. Each image is associated with a 50 word lexicon defined by [9]. Many images in this dataset suffer from low resolution, blur, and noise.

• IIIT5k [30] contains 3000 cropped word images collected from the Internet. Each image has a 50 word lexicon and a 1000 word lexicon.

• ICDAR 2003 (IC03) [31] contains 860 cropped word images for testing. Following the evaluation protocol in [9], we recognize images containing only alphanumeric characters with at least three characters. Each image is specified with a 50 word lexicon defined by [9] and a full lexicon comprising all the words that appear in the test set.

• ICDAR 2013 (IC13) [32] is derived from ICDAR 2003. Following [6], we remove images that contain non-alphanumeric characters resulting in 1015 cropped word images without any pre-defined lexicon.

• Total-Text [33] has annotated word images with three different text orientations including horizontal, multi-oriented, and curved text. We select the multi-oriented and curved text collections that contain 480 and 971 images, respectively.

We use the synthetic dataset for training, including Synth90k [26] and SynthText [27]. Our model is evaluated on all other real-world test datasets without any finetuning.

### 4.2 Implementation details

The overall network architecture is described in Table 1. The filters and channels for the convolutional layers are given in detail. The building residual units are shown in brackets with the numbers of units stacked. The pooling layers in the rectification network are max pooling with filter 2×2 and stride 2×2. In the recognition network, features are down-sampled by Residual Unit1_1, Residual Unit2_1 with stride 2×2 and Residual Unit3_1, Residual Unit4_1, and Residual Unit5_1 with stride 2×1. All the

**Table 2** Lexicon-free results on several benchmarks with different number of iterations[a)]

| The number of iterations | SVT (%) | IIIT5k (%) | IC03 (%) | IC13 (%) | SVTP (%) | CUTE80 (%) | IC15 (%) | Total-Text (multi-oriented) (%) | Total-Text (curved) (%) | Time (ms) |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 86.1 | 91.0 | 91.7 | 89.4 | 74.6 | 75.0 | 71.8 | 70.6 | 53.0 | 16.34 |
| 1 | 86.2 | 92.6 | 92.9 | 90.7 | 77.0 | 82.3 | 72.0 | 72.3 | 56.4 | 20.03 |
| 2 | 86.6 | 92.8 | 92.9 | 92.0 | 78.2 | 83.3 | 74.5 | 74.4 | 58.4 | 24.28 |
| 3 | **89.0** | 93.6 | 93.6 | 92.2 | 80.8 | 87.2 | 76.4 | 75.6 | 67.1 | 28.48 |
| 4 | 88.7 | **94.3** | **94.0** | **93.3** | **81.2** | 88.2 | 76.8 | **77.1** | **69.3** | 32.57 |
| 5 | 88.4 | **94.3** | **94.0** | 93.0 | 79.8 | **88.5** | **78.4** | 76.0 | 69.0 | 36.70 |

a) The best results are shown in bold.

**Table 3** The effect of the proposed envelope-refinement structure (ER)[a)]

| Method | SVT (%) | IIIT5k (%) | IC03 (%) | IC13 (%) | SVTP (%) | CUTE80 (%) | IC15 (%) | Total-Text (multi-oriented) (%) | Total-Text (curved) (%) |
|---|---|---|---|---|---|---|---|---|---|
| PRN (w/o ER) | 87.5 | 92.4 | 93.5 | 91.5 | 78.1 | 86.1 | 74.1 | 73.8 | 65.2 |
| PRN | **89.0** | **93.6** | **93.6** | **92.2** | **80.8** | **87.2** | **76.4** | **75.6** | **67.1** |

a) The best results are shown in bold.

convolutional layers are followed by ReLU activation and batch normalization. The input images are resized to 64×256, and the 32×64 downsampled images serve as the input of the localization network. After performing spatial transformation, the rectified images have a size of 64×256 in the middle steps and 32×100 in the last step. Moreover, we do not use any data augmentation. The model is trained for 1.2 million iterations with the ADADELTA [34] optimization method. The initial learning rate is set to 0.1 and is divided by 10 at 50%, 67%, and 83% of the total number of training iterations. The number of border points is set as 20, and the different values cause negligible effects. The label space comprises 94 classes including 10 digits, 52 case sensitive letters, and 32 punctuations. Case insensitive correctly recognized words (C.R.W) is used as the evaluation metric. During evaluation, the non-alphanumeric symbols are filtered and all letters are normalized to lower cases. In the lexicon-free setting, we perform independent beam search for forward and backward decoders to obtain two k-best lists and then reorder the combination of these two lists to find the best candidate. The beam sizes of the two decoders are set as 5. In the lexicon-based setting, we pick the nearest lexicon word with the generated string using the metric of edit distance.

Our network is implemented under the Pytorch [35] framework. Most parts of our model are GPU-accelerated because of the CUDA backend. All experiments are performed on a workstation with one Inter(R) Xeon(R) E5-2630 2.20 GHz CPU, an NVIDIA TITAN X GPU, and a 256 GB RAM.

### 4.3 Ablation studies

The proposed method progressively rectifies irregular text and thus, generates better rectified images, resulting in easier recognition. To investigate the effect of the number of iterations, we conduct experiments on standard benchmarks and report the results in Table 2. As the number of iterations increases, the recognition results gradually perform better. In particular, the performance improvements on curved text benchmarks are remarkable, which suggests the significance of progressive rectification in recognizing severely distorted text. Moreover, the recurrent structure is capable of continually refining the geometric transformation of irregular text under the same parametric capacity. The model reaches the performance plateau at four iterations. Considering the tradeoff between accuracy and speed, the number of iterations is set as three in the following experiments. In addition, by comparing the PRN and PRN (w/o ER) in Table 3, we can see that the envelope-refinement structure leads to significant performance improvement under the same number of iterations. The main reason is that the original character information is preserved and thus, the missing information can be recovered. Furthermore, the succeeding recognition network benefits from the integrity of the text.

Some visualization examples are presented in Figure 4. As observed, the text becomes more regular


**Figure 4** (Color online) Visualizations of the rectified images during progressive refinement.

as the number of iterations increases. In addition, the missing text information in the previous step can be remedied in subsequent processes; therefore, the intact structures of the characters are effectively preserved during iterative rectification leading to better recognition. Note that our network not only transforms the text in the direction that is more beneficial to recognition, but also gradually removes background noise.

### 4.4 Performance on irregular benchmarks

Recognizing irregular text is very challenging, because of various character placements. To validate the effectiveness of our method, we evaluate our PRN on several irregular benchmarks and summarize the results in Table 4 [1, 4–7, 9, 15–20, 30, 36–40]. Considering [41] used extra synthetic and real images for training, we did not compare the results with [41] to ensure fairness. As observed in Table 4, our method outperforms other approaches by a large margin on most benchmarks. Especially compared with the rectification-based methods [5–7, 18, 20, 37], our method outperforms [6] by 8.7% on CUTE80 and outperforms [37] by 5.9% on SVTP-Full, which suggests the effectiveness of our method. Zhan and Lu [20] adopted a fixed-order polynomial to estimate text orientation; however, the text line poses that could be represented were limited. By comparison, we directly predict the envelope of text region that is more flexible. We observe that our method outperforms [20] by 4.9% on CUTE80 and 1.6% on SVTP. Although our network is supervised with only word-level annotations, it still performs better than the methods trained with both word-level and character-level annotations [4, 7, 17, 40] on nearly all the benchmarks. We also find that the performance gains on the curved text benchmarks are better than those on the perspective text benchmarks. The distortions of curved text are complicated and hard to model; therefore, existing methods perform worse on curved text. In contrast, our approach can effectively rectify severely distorted text, and therefore, obtains better performance gains on curved text benchmarks. As shown in Figure 4, our PRN is capable of rectifying the irregular text with various degrees of deformation. Compared with [19], we do not destroy the aspect ratio of text and thus, the characters have no deformation. Note that Luo et al. [37] adopted a more complex curriculum learning strategy to enable stable convergence of their network and required tightly bounding box annotations. However, our method outperforms [37] on all irregular benchmarks. Liu et al. [38] used clean images

**Table 4** Scene text recognition accuracies on irregular datasets[a)]

| Method | SVTP (%) | | | CUTE80 (%) | IC15 (%) | Total-Text (multi-oriented) (%) | Total-Text (curved) (%) |
|---|---|---|---|---|---|---|---|
| | 50 | Full | None | None | None | None | None |
| ABBYY [9] | 40.5 | 26.1 | – | – | – | – | – |
| Mishra et al. [30] | 45.7 | 24.7 | – | – | – | – | – |
| Phan et al. [16] | 75.6 | 67.0 | – | – | – | – | – |
| Shi et al. [1] | 92.6 | 72.6 | 66.8 | 54.9 | – | – | – |
| Shi et al. [5] | 91.2 | 77.4 | 71.8 | 59.2 | – | – | – |
| Liu et al. [18] | – | – | 73.5 | – | – | – | – |
| Cheng et al. [19] | 94.0 | 83.7 | 73.0 | 76.8 | 68.2 | – | – |
| Fang et al. [15] | – | – | – | – | 71.2 | – | – |
| Liu et al. [36] | – | – | 74.4 | – | – | – | – |
| Shi et al. [6] | – | – | 78.5 | 79.5 | 76.1 | – | – |
| Luo et al. [37] | <u>94.3</u> | <u>86.7</u> | 76.1 | 77.4 | 68.8 | – | – |
| Liu et al. [38] | – | – | 73.9 | 62.5 | – | – | – |
| Zhan and Lu [20] | – | – | 79.6 | 83.3 | <u>76.9</u> | – | – |
| Lyu et al. [39] | – | – | **82.3** | 86.8 | 76.3 | – | – |
| Yang et al. [17][b)] | 93.0 | 80.2 | 75.8 | 69.3 | – | – | – |
| Cheng et al. [4][b)] | 92.6 | 81.6 | 71.5 | 63.9 | 66.2 | – | – |
| Yang et al. [7][b)] | – | – | 80.8 | <u>87.5</u> | **78.7** | – | – |
| Liao et al. [40][b)] | – | – | – | 78.1 | – | – | – |
| PRN (ours) | **95.1** | **92.6** | <u>81.2</u> | **88.2** | 76.8 | **77.1** | **69.3** |

a) "50" and "Full" represent the sizes of lexicon used for lexicon-based recognition, and "None" represents lexicon-free recognition. The best results are shown in bold, and the second best results are shown in underline.

b) The approaches are trained with both word-level and character-level annotations.

for additional supervision; however, our method achieves better performance without auxiliary training data. We also report the recognition performance on Total-Text that has not been recorded in previous literatures. Our method achieves promising results on both multi-oriented and curved text collections. Furthermore, the recognition network can be adapted to any other structures. By using the focusing attention mechanism in [4] and the edit probability in [14], the performance can be further improved.

### 4.5 Performance on regular benchmarks

We also conduct experiments on several regular benchmarks. Most samples in these datasets are regular text; however, irregular text also exists. We report our results in Table 5 [1–6, 10–12, 14, 17–20, 36–40, 42–48]. Compared with existing methods, our PRN effectively improves recognition performance. We observe that our PRN performs best on IIIT5k in the lexicon-free setting. IIIT5k contains many curved text, which demonstrates the advantage of our method in dealing with distorted text. In the lexicon-based scenario, our PRN achieves the second best results on SVT and IIIT5k. Note that Bai et al. [14] applied a specially designed edit probability to train their networks, while we only use traditional frame-wise loss. Additionally, we do not use the data augmentation as used in [39]. Jaderberg et al. [11] benefited from a pre-defined 90k lexicon, but their method could only recognize the words in its dictionary. It is worth remarking that Refs. [4, 17, 40, 48] used extra character bounding box annotations. In contrast, our method only requires textual labels to achieve better or comparable performance, which saves a lot of resources.

## 5 Conclusion

In this paper, we proposed a novel PRN for irregular text recognition. We divided the rectification process into multiple progressive steps to mitigate the rectification difficulty of each step. A rough rectification was first performed and then gradually adjusted to optimal rectification through continuous refinements.

**Table 5** Scene text recognition accuracies on regular datasets[a)]

| Method | SVT (%) | | IIIT5k (%) | | | IC03 (%) | | | IC13 (%) |
|---|---|---|---|---|---|---|---|---|---|
| | 50 | None | 50 | 1000 | None | 50 | Full | None | None |
| Wang et al. [42] | 70.0 | – | – | – | – | 90.0 | 84.0 | – | – |
| Bissacco et al. [10] | 90.4 | 78.0 | – | – | – | – | – | – | 87.6 |
| Yao et al. [43] | 75.9 | – | 80.2 | 69.3 | – | 88.5 | 80.3 | – | – |
| Rodriguez-Serrano et al. [12] | 70.0 | – | 76.1 | 57.4 | – | – | – | – | – |
| Jaderberg et al. [44] | 86.1 | – | – | – | – | 96.2 | 91.5 | – | – |
| Jaderberg et al. [11] | 95.4 | 80.7 | 97.1 | 92.7 | – | 98.7 | **98.6** | 93.1 | 90.8 |
| Jaderberg et al. [45] | 93.2 | 71.7 | 95.5 | 89.6 | – | 97.8 | 97.0 | 89.6 | 81.8 |
| Shi et al. [1] | 97.5 | 82.7 | 97.8 | 95.0 | 81.2 | 98.7 | 98.0 | 91.9 | 89.6 |
| Lee et al. [3] | 96.3 | 80.7 | 96.8 | 94.4 | 78.4 | 97.9 | 97.0 | 88.7 | 90.0 |
| Liu et al. [46] | 95.5 | 83.6 | 97.7 | 94.5 | 83.3 | 96.9 | 95.3 | 89.9 | 89.1 |
| He et al. [2] | 92.0 | – | 94.0 | 91.6 | – | 97.0 | 94.4 | – | – |
| Wang and Hu [47] | 96.3 | 81.5 | 98.0 | 95.6 | 80.8 | 98.8 | 97.8 | 91.2 | – |
| Bai et al. [14] | 96.6 | 87.5 | 99.5 | 97.9 | 88.3 | 98.7 | 97.9 | 94.6 | **94.4** |
| Luo et al. [37] | 96.6 | 88.3 | 97.9 | 96.2 | 91.2 | 98.7 | 97.8 | **95.0** | 92.4 |
| Liu et al. [36] | 97.1 | 85.5 | 98.4 | 96.1 | 85.2 | 98.5 | 97.7 | 92.9 | 90.3 |
| Shi et al. [5] | 95.5 | 81.9 | 96.2 | 93.8 | 81.9 | 98.3 | 96.2 | 90.1 | 88.6 |
| Liu et al. [18] | – | 84.4 | – | – | 83.6 | – | – | 91.5 | 90.8 |
| Cheng et al. [19] | 96.0 | 82.8 | 99.6 | 98.1 | 87.0 | 98.5 | 97.1 | 91.5 | – |
| Liu et al. [38] | 96.8 | 87.1 | 97.3 | 96.1 | 89.4 | 98.1 | 97.5 | 94.7 | 94.0 |
| Shi et al. [6] | 97.4 | 89.5 | 99.6 | 98.8 | 93.4 | 98.8 | 98.0 | 94.5 | 91.8 |
| Zhan and Lu [20] | 97.4 | **90.2** | 97.4 | 98.8 | 93.3 | – | – | – | 91.3 |
| Lyu et al. [39] | 97.2 | 90.1 | **99.8** | **99.1** | 94.0 | **99.4** | 98.1 | 94.3 | 92.7 |
| Liu et al. [48][b)] | 96.1 | – | 96.9 | 94.3 | 86.6 | 98.4 | 97.9 | 93.1 | 92.7 |
| Yang et al. [17][b)] | 95.2 | – | 97.8 | 96.1 | – | 97.7 | – | – | – |
| Cheng et al. [4][b)] | 97.1 | 85.9 | 99.3 | 97.5 | 87.4 | 99.2 | 97.3 | 94.2 | 93.3 |
| Liao et al. [40][b)] | **98.5** | 82.1 | **99.8** | 98.9 | 92.0 | – | – | – | 91.4 |
| PRN (ours) | 97.5 | 88.7 | 99.6 | 98.9 | **94.3** | 98.6 | 98.0 | 94.0 | 93.3 |

a) "50", "1000" and "Full" represent the size of lexicon used for lexicon-based recognition, and "None" represents lexicon-free recognition. The best results are shown in bold, and the second best results are shown in underline.

b) The approaches are trained with both word-level and character-level annotations.

Moreover, we designed an envelope-refinement structure to track and refine the text line envelope instead of only transmitting the rectified images. Based on the refined envelope, the original image was consistently used for spatial transformation. Therefore, our network could effectively keep the integrity of the text and avoid the boundary damage problem during iterative rectification. Additionally, the rectification and recognition networks were jointly trained under the same objective for text recognition. Thus, the text was gradually rectified in the direction that was more beneficial to recognition. Moreover, our model can be easily applied using only word-level annotations, thereby avoiding the laborious character-level labeling effort. Extensive experiments revealed that our PRN consistently outperformed other methods and achieved state-of-the-art results on challenging benchmarks, which suggested that it was more suitable for irregular text recognition. A text recognizer usually works in conjunction with a text detector to compose a scene text reading system. Because our PRN can accurately recognize irregular text, our network is capable of effectively relaxing the requirements of a text detector and alleviating the noise caused by the detector. In the future, we plan to investigate how to properly combine the proposed text recognition model with a text detection method.

**Acknowledgements** This work was supported by National Natural Science Foundation of China (Grant Nos. 61772527, 61806200).

**References**

1 Shi B G, Bai X, Yao C. 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, 2017, 39: 2298–2304
2 He P, Huang W L, Qiao Y, et al. Reading scene text in deep convolutional sequences. In: Proceedings of AAAI Conference on Artificial Intelligence, 2016. 3501–3508
3 Lee C Y, Osindero S. Recursive recurrent nets with attention modeling for ocr in the wild. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2016. 2231–2239
4 Cheng Z Z, Bai F, Xu Y L, et al. Focusing attention: towards accurate text recognition in natural images. In: Proceedings of IEEE International Conference on Computer Vision, 2017. 5086–5094
5 Shi B G, Wang X G, Lyu P Y, et al. Robust scene text recognition with automatic rectification. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2016. 4168–4176
6 Shi B G, Yang M K, Wang X G, et al. ASTER: an attentional scene text recognizer with flexible rectification. IEEE Trans Pattern Anal Mach Intell, 2019, 41: 2035–2048
7 Yang M K, Guan Y S, Liao M H, et al. Symmetry-constrained rectification network for scene text recognition. In: Proceedings of IEEE International Conference on Computer Vision, 2019
8 Jaderberg M, Simonyan K, Zisserman A, et al. Spatial transformer networks. In: Proceedings of Advances in Neural Information Processing Systems, 2015. 2017–2025
9 Wang K, Babenko B, Belongie S. End-to-end scene text recognition. In: Proceedings of IEEE International Conference on Computer Vision, 2011. 1457–1464
10 Bissacco A, Cummins M, Netzer Y, et al. Photoocr: reading text in uncontrolled conditions. In: Proceedings of IEEE International Conference on Computer Vision, 2013. 785–792
11 Jaderberg M, Simonyan K, Vedaldi A, et al. Reading text in the wild with convolutional neural networks. Int J Comput Vis, 2016, 116: 1–20
12 Rodriguez-Serrano J A, Gordo A, Perronnin F. Label embedding: a frugal baseline for text recognition. Int J Comput Vis, 2015, 113: 193–207
13 Graves A, Fernández S, Gomez F, et al. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In: Proceedings of International Conference on Machine Learning, 2006. 369–376
14 Bai F, Cheng Z Z, Niu Y, et al. Edit probability for scene text recognition. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2018. 1508–1516
15 Fang S C, Xie H T, Zhang Z J, et al. Attention and language ensemble for scene text recognition with convolutional sequence modeling. In: Proceedings of ACM Conference on Multimedia, 2018. 248–256
16 Phan T Q, Shivakumara P, Tian S, et al. Recognizing text with perspective distortion in natural scenes. In: Proceedings of IEEE International Conference on Computer Vision, 2013. 569–576
17 Yang X, He D F, Zhou Z H, et al. Learning to read irregular text with attention mechanisms. In: Proceedings of International Joint Conference on Artificial Intelligence, 2017. 3280–3286
18 Liu W, Chen C F, Wong K Y K. Char-net: a character-aware neural network for distorted scene text recognition. In: Proceedings of AAAI Conference on Artificial Intelligence, 2018
19 Cheng Z Z, Liu X Y, Bai F, et al. AON: towards arbitrarily-oriented text recognition. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2018. 5571–5579
20 Zhan F N, Lu S J. ESIR: end-to-end scene text recognition via iterative rectification. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2019. 2059–2068
21 Chen J, Lian Z H, Wang Y Z, et al. Irregular scene text detection via attention guided border labeling. Sci China Inf Sci, 2019, 62: 220103
22 Bookstein F L. Principal warps: thin-plate splines and the decomposition of deformations. IEEE Trans Pattern Anal Machine Intell, 1989, 11: 567–585
23 Lin C-H, Lucey S. Inverse compositional spatial transformer networks. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2017. 2568–2576
24 He K, Zhang X Y, Ren S Q, et al. Delving deep into rectifiers: surpassing human-level performance on imagenet classification. In: Proceedings of IEEE International Conference on Computer Vision, 2015. 1026–1034
25 Saxe A M, McClelland J L, Ganguli S. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. 2013. ArXiv: 1312.6120
26 Jaderberg M, Simonyan K, Vedaldi A, et al. Synthetic data and artificial neural networks for natural scene text recognition. 2014. ArXiv: 1406.2227
27 Gupta A, Vedaldi A, Zisserman A. Synthetic data for text localisation in natural images. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2016. 2315–2324
28 Risnumawan A, Shivakumara P, Chan C S, et al. A robust arbitrary text detection system for natural scene images. Expert Syst Appl, 2014, 41: 8027–8048
29 Karatzas D, Gomez-Bigorda L, Nicolaou A, et al. ICDAR 2015 competition on robust reading. In: Proceedings of International Conference on Document Analysis and Recognition (ICDAR), 2015. 1156–1160
30 Mishra A, Alahari K, Jawahar C. Top-down and bottom-up cues for scene text recognition. In: Proceedings of IEEE

Conference on Computer Vision and Pattern Recognition, 2012. 2687–2694
31 Lucas S M, Panaretos A, Sosa L, et al. ICDAR 2003 robust reading competitions: entries, results, and future directions. Int J Document Anal Recogn, 2005, 7: 105–122
32 Karatzas D, Shafait F, Uchida S, et al. ICDAR 2013 robust reading competition. In: Proceedings of International Conference on Document Analysis and Recognition, 2013. 1484–1493
33 Ch'ng C K, Chan C S. Total-text: a comprehensive dataset for scene text detection and recognition. In: Proceedings of International Conference on Document Analysis and Recognition, 2017. 935–942
34 Zeiler M D. ADADELTA: an adaptive learning rate method. 2012. ArXiv: 1212.5701
35 Ketkar N. Introduction to pytorch. In: Deep Learning with Python. Berkeley: Apress, 2017. 195–208
36 Liu W, Chen C F, Wong K K. SAFE: scale aware feature encoder for scene text recognition. In: Proceedings of Asian Conference on Computer Vision, 2018. 196–211
37 Luo C J, Jin L W, Sun Z H. MORAN: a multi-object rectified attention network for scene text recognition. Pattern Recogn, 2019, 90: 109–118
38 Liu Y, Wang Z W, Jin H L, et al. Synthetically supervised feature learning for scene text recognition. In: Proceedings of European Conference on Computer Vision, 2018. 435–451
39 Lyu P Y, Yang Z C, Leng X H, et al. 2D attentional irregular scene text recognizer. 2019. ArXiv: 1906.05708
40 Liao M H, Zhang J, Wan Z Y, et al. Scene text recognition from two-dimensional perspective. In: Proceedings of AAAI Conference on Artificial Intelligence, 2019. 8714–8721
41 Li H, Wang P, Shen C H, et al. Show, attend and read: a simple and strong baseline for irregular text recognition. In: Proceedings of AAAI Conference on Artificial Intelligence, 2019. 8610–8617
42 Wang T, Wu D J, Coates A, et al. End-to-end text recognition with convolutional neural networks. In: Proceedings of International Conference on Pattern Recognition, 2012. 3304–3308
43 Yao C, Bai X, Shi B G, et al. Strokelets: a learned multi-scale representation for scene text recognition. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2014. 4042–4049
44 Jaderberg M, Vedaldi A, Zisserman A. Deep features for text spotting. In: Proceedings of European Conference on Computer Vision, 2014. 512–528
45 Jaderberg M, Simonyan K, Vedaldi A, et al. Deep structured output learning for unconstrained text recognition. 2014. ArXiv: 1412.5903
46 Liu W, Chen C F, Wong K K, et al. Star-net: a spatial attention residue network for scene text recognition. In: Proceedings of British Machine Vision Conference, 2016. 7
47 Wang J F, Hu X L. Gated recurrent convolution neural network for ocr. In: Proceedings of Neural Information Processing Systems, 2017. 334–343
48 Liu Z C, Li Y X, Ren F B, et al. Squeezedtext: a real-time scene text recognition by binary convolutional encoder-decoder network. In: Proceedings of AAAI Conference on Artificial Intelligence, 2018