Robustly Recognizing Irregular Scene Text by Rectifying Principle Irregularities

Changsheng Xu
School of Computer Science
Fudan University
xucs18@fudan.edu.cn

Yang Wang
Dept. of Computer Sci. & Tech.
Tongji University
tongji_wangyang@tongji.edu.cn

Fan Bai
School of Computer Science
Fudan University
fbai19@fudan.edu.cn

Jihong Guan
Dept. of Computer Sci. & Tech.
Tongji University
jhguan@tongji.edu.cn

Shuigeng Zhou[*]
School of Computer Science
Fudan University
sgzhou@fudan.edu.cn

Abstract

Reading irregular scene text is a challenging problem in scene text recognition. Rectification is a popular measure to reduce irregularities of text in images. Existing rectification methods seek to rectify text images into a strictly regular form via free parametric transformation functions. However, they always suffer from information loss or severe deformation due to their poor constraints to the transformation functions. In our investigation, we found that CNN and attention are robust to many slight irregularities. What inspires us to propose a novel and effective rectification method that mainly rectifies the principle regularities, and leaves the slight irregularities to the CNN-LSTM-attention recognizer. Our rectification method first estimates the character densities and directions of the input image in a down-sampled map then finds a best fitting curve from a small predefined Bézier curve set, and finally rectifies the input image with a transformation function corresponding to the selected curve. Transformation functions are carefully designed so that they neither lose important visual information nor cause severe deformation. Extensive experiments on seven benchmark datasets show that our method achieves the state of the art performance in most cases, especially in curved text recognition.

1. Introduction

Scene text recognition (STR) plays an important role in life and industry because it can serve as a bridge between optical text data and natural language processing [33, 36, 30, 4]. Though current STR methods [28, 3] achieve satisfactory accuracy on regular texts by sequence to sequence learning models, reading irregular texts is still an open problem. Regular texts in images are usually arranged horizontally from left to right. In contrast, irregular texts can be multi-oriented, perspective, and curved.

Existing solutions for irregular text recognition include rectification-based approaches and 2D-decoder-based approaches. The former approaches first transform the input images or hidden feature maps with transformation functions, then recognize the transformed images or feature maps with sequence to sequence recognizers. The latter approaches directly decode from 2D feature maps [18, 17]. The most popular text rectifiers are TPS-based spatial transformer networks (STN) [11]. ASTER [31] and RARE [30] train the STN in a weak supervising manner with only text-level annotations. ScRN [34] first trains an estimator for characters' orientations, scales, and centerlines with pixel-level annotations then generate TPS control points with the constraints of the estimated values.

However, as shown in Fig. 1, TPS-based rectifying algorithms may lead to severe distortion or even important visual information loss due to improper transformation.

According to our study, CNN-LSTM-Attention [7, 1] based recognizers are highly robust to small irregularities of texts [30, 3]. And a well-trained recognizer can reliably read the words with about $\frac{1}{4}\times$ to $1\times$ scale variation in width and rotation within $\pm 20°$. Thus, a sophisticated rectification mechanism that transforms texts into strictly regular forms is not necessary for the reading task. On the contrary, a simple rectifier may generate rectified images with some spatial deformation, which can be still well read by a sequence recognizer.

Based on the observation above, we propose to **R**ectify only the **P**rincipal **I**rregularities (RPI in short) in the text images, instead of to find a strictly regular form. As shown in Fig. 1, our method rectifies the input images to a roughly

[*]Correspondence author.

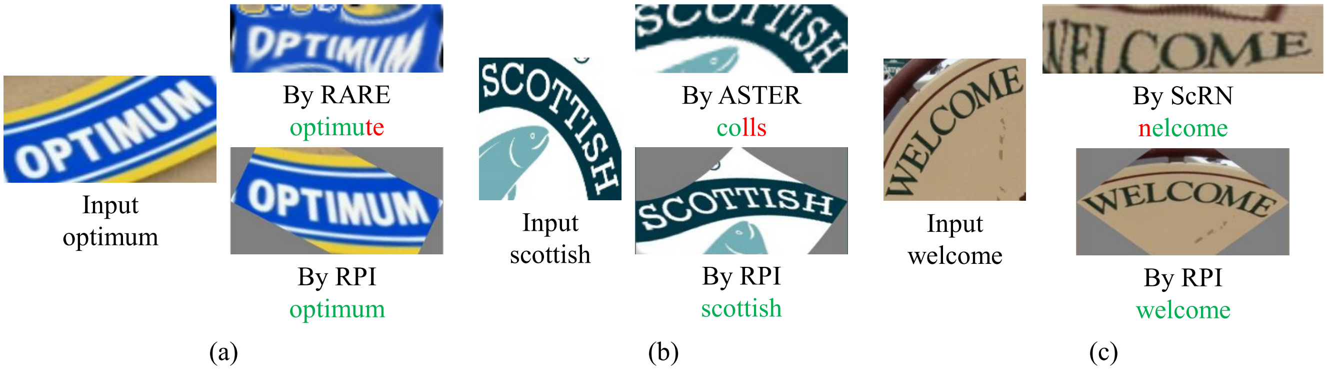


Figure 1: (a) The image rectified by RARE [30] is severely distorted, leading to a wrong recognition. (b) The image rectified by ASTER [29] loses many characters, which causes a disastrous error. (c) The image rectified by ScRN [34] loses a part of the first character, which makes the character not be recognized. In all of the three cases, our method dose not try to rectify the input images to a strictly regular form, but to rectify the major irregularities, while important visual information is preserved for correct recognition.

horizontally left-to-right form, but the boundaries of the rectified images are not guaranteed to be closed to the text's edges. And texts in such a form can still be correctly read by the recognizer. The method is called RPI that will be detailed in Sec. 3.

We conduct extensive experiments to evaluate the effectiveness of the proposed method. Experimental results conform to our expectations and show the performance advantage of our method over the existing ones. Experimental details are presented in Sec. 4.

2. Related Work

Early works mainly use a bottom-up framework that detects individual characters by sliding window [33], connected components [24] or Hough voting [36]. Top-down frameworks become popular later, pioneering works include [9, 10], which use a CNN with structured output for unconstrained recognition and a 90k-word CNN-based classifier.

Sequence to sequence learning is the state of the art of regular scene text recognition. [15] reads scene text by a recursive recurrent net (RN) with attention modeling. [28, 30] propose end-to-end neural networks that combine CNN, RN and an attention or CTC [5] based decoder. [3] and [2] improve the attention decoder with focusing-attention network (FAN) and edit probability (EP), respectively.

Recently, many STR works focus on irregular text recognition. Existing techniques roughly fall into two types: rectification-based approaches and 2D-decoder-based approaches.

Rectification-based approaches handle the irregularities by applying transformations to the input images or the hidden feature maps. [30] trains a TPS-based STN in a weakly supervised manner with only word-level annotations. This work was recently extended by considering more flexible TPS [31]. AON [4] reconstructs text orientation by learning to weigh four directions. [34] advances the STN-based methods by adding character-level annotations and extra constraints. Other rectification mechanisms are also explored. [37] conducts rectification iteratively, and [22] rectifies the images by estimating the offset map.

2D-decoder approaches read irregular texts from a 2D feature map. Li *et al.* [17] employed a tailored 2D attention. Liu *et al.* [19] rectified individual characters after detecting them. Yang *et al.* [35] trained a 2D attention with character-level supervision. Liao *et al.* [18] detected and recognized individual characters by a fully convolutional network, and then recognized the words by character voting. Wan *et al.* [32] decoded texts from character semantic segmentation with their orders in consideration and achieved good performance. However, the semantic segmentation-based methods consume much more computation resources, which limits its application in many situations.

In this paper, we propose a new and effective rectification-based method RPI for irregular text recognition, which pays attention to the significant irregularities in the input images. We innovatively exploit the Bézier curve to model text orientation and achieve state-of-the-art recognition performance in extensive experiments on seven benchmark datasets.

3. Method

The workflow of our proposed method RPI is shown in Fig. 2. RPI consists of a *Densities and Orientations Estimating Module* (DOEM), a *Curve Fitting Module* (CFM), a *Grid Sampling Module* (GSM) and a *Sequence Recognizing Module* (SRM). DOEM first estimates the character densities and orientations on a 4-stride map of the input image. With the estimated results, CFM scores the fitness of each

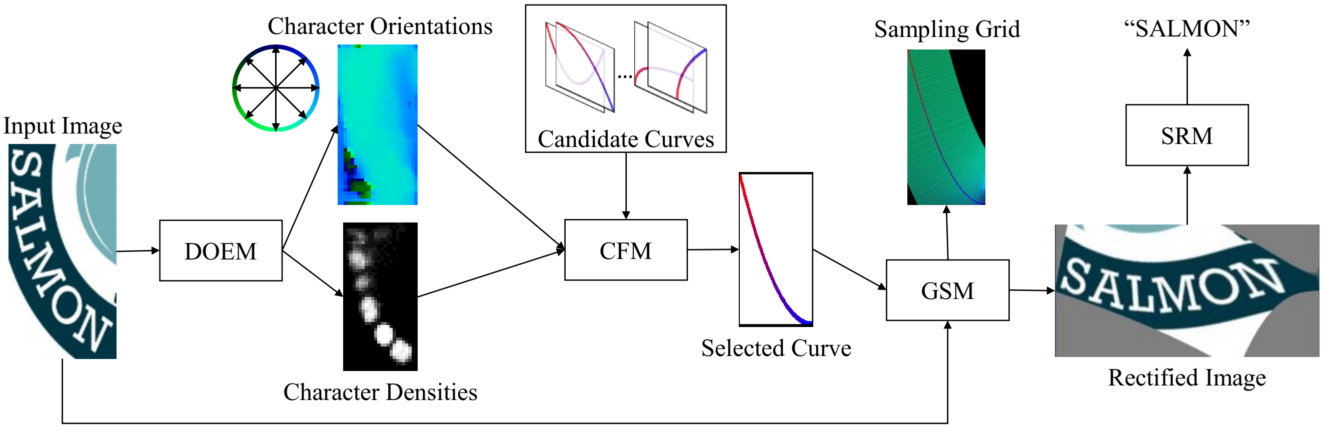


Figure 2: The work flow of the proposed method RPI. DOEM first estimates character densities and orientations, and CFM selects the best fitting curve from the candidates. Then, GSM generates the sampling grid and samples from the input image with bilinear interpolation. Finally, SRM reads "SALMON" from the rectified image.

curve from a predefined curve set and selects the best fitting one. GSM then generates a sampling grid that neither loses important visual information nor causes severe abnormal deformation and rectifies the input image based on the generated grid with bilinear interpolation. SRM finally recognizes the rectified image with a CNN-LSTM-attention encoder-decoder.

In what follows, we introduce these modules in detail.

3.1. Densities and Orientations Estimating Module

The backbone of DOEM is a small down-sample-up-sample pyramid architecture. The output of DOEM is in $\frac{1}{4}$ height and $\frac{1}{4}$ width of the input image with 3 channels. A sigmoid function is applied to the first channel to generate the character densities α, and normalization is applied to the last 2 channels to generate the character orientations $\vec{\theta}$. The annotation $\hat{\alpha}$ for character densities is a map of one-hot vectors, indicating whether the pixels belong to a character or the background. And annotation $\hat{\vec{\theta}}$ is a map of normalized vectors, indicating the orientations of pixels if they belong to a character, otherwise zero vectors. DOEM is trained to minimize the cross entropy between α and $\hat{\alpha}$, and to maximize the inner product of $\vec{\theta}$ and $\hat{\vec{\theta}}$ where $\hat{\alpha} = 1$. The loss function of DOEM is as follows:

$$\mathcal{L}_{DOEM} = -\ln(\alpha)\hat{\alpha} - \ln(1-\alpha)(1-\hat{\alpha}) - \hat{\alpha}(\vec{\theta}\odot\hat{\vec{\theta}} - 1)\,. \quad (1)$$

3.2. Curve Fitting Module

Bézier curve is a continuous parametric curve commonly used in computer graphics. By checking a large number of scene text images, we found that the shapes of most scene texts can be represented by quadratic Bézier curves. Therefore, we try to find quadratic Bézier curves to fit the estimated character densities and orientations. Most STN-based methods predict control points from a continuous space, which may lead to unexpected results without explicitly considering the constraints on the control points [34]. Our CFM selects a Bézier curve from a small predefined candidate curve set by scoring their fitness to the estimated character densities and orientations, which makes it simple, yet robust and effective.

The set of candidate curves is not necessary to be too large, because slight irregularities can be actually handled by the recognizer. Therefore, we select only the quadratic Bézier curves whose control points are confined to $\{-1, 0, 1\} \times \{-1, 0, 1\}$ with the rule that each curve should be neither too short (length less than 2), nor too close to a boundary. Eventually, 37 curves are selected as candidates, which are shown in Fig. 3.

For the convenience of description, we also map the coordinates of the input image, α and $\vec{\theta}$ to $[-1, 1] \times [-1, 1]$. Intuitively, with the estimated α, $\vec{\theta}$ and a given expected direction $\vec{d}$, we define the fitness of a point $P(x, y)$ on the image to $\vec{d}$ as follows:

$$\mathrm{F}_{\alpha,\vec{\theta}}(P, \vec{d}) = \alpha_{x,y}\vec{\theta}_{x,y} \odot \tau(\vec{d}) \quad (2)$$

where "$\odot$" stands for element wise production, and

$$\tau(\vec{v}) = \frac{\vec{v}}{||\vec{v}||} \quad (3)$$

is the normalized vector that represents the orientation of $\vec{v}$.

Then, given a point P and a hyper parameter $\lambda \in (0, 1]$, considering an expansion line segment $\tilde{P}$ of 2λ length that

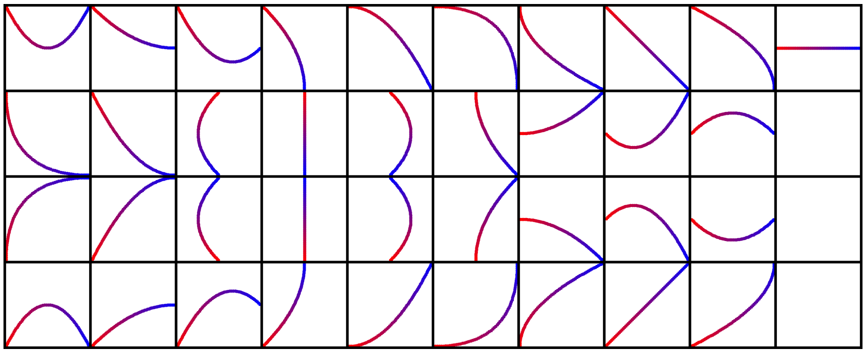


Figure 3: The 37 candidate Bézier curves. The curves begin from red and end to blue.

is perpendicular to $\vec{d}$ and centered at P, we define the score of $\tilde{P}$ as the integral of the fitness of all the points on $\tilde{P}$, i.e.,

$$\mathrm{C}_{\alpha,\vec{\theta}}(\tilde{P},\vec{d}) = \int_{P-\lambda\omega(\vec{d})}^{P+\lambda\omega(\vec{d})} \mathrm{F}_{\alpha,\vec{\theta}}(s,\vec{d})\,\mathrm{d}s \tag{4}$$

where

$$\omega(\vec{v}) = (-\vec{v}^{y}, \vec{v}^{x}) \tag{5}$$

is the counter clockwise rotation of $\vec{v}$.

Finally, the fitting score of a Bézier curve $\mathbb{B}$ is defined as the integral of the scores of the expansion line segments of the points on $\mathbb{B}$, and the result is normalized by the length of $\mathbb{B}$. Formally,

$$\mathrm{S}_{\alpha,\vec{\theta}}(\mathbb{B}) = \frac{1}{||\mathbb{B}||}\int_{\mathbb{B}} \mathrm{C}_{\alpha,\vec{\theta}}\left(\tilde{s},\tau(\vec{\mathrm{d}s})\right)\mathrm{d}s\,. \tag{6}$$

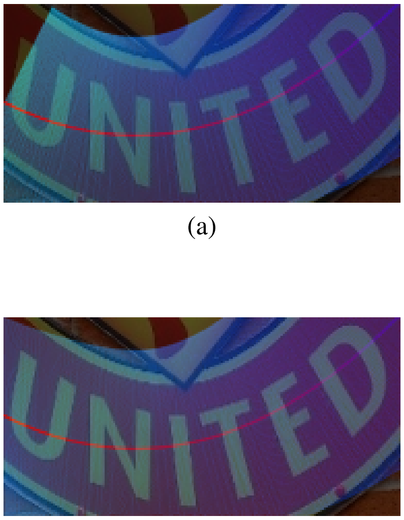
And the curve with the maximum score will be selected.

We implement CFM discretely. Specifically, we evenly pick 64 points on the curve $\mathbb{B}$, and then evenly pick 32λ points on each expansion line segment where 32λ should be an even integer. And the score of $\mathbb{B}$ is computed as the sum of the fitness of those picked points. Note that CFM has no trainable parameter.

3.3. Grid Sampling Module

Given a selected curve $\mathbb{B}$, we generate the sampling grid that is vertically centered at $\mathbb{B}$. Firstly, $\mathbb{B}$ is evenly divided into 256 segments. Generally, for the midpoint P_i of the i-th segment, we evenly pick 64 points over the perpendicular line of length 2 and centered at P_i, and take these 64 points as the i-th column of the output.

However, in some exceptional circumstances, such a naive sampling strategy may lose information. For example, in Fig. 4 (a), some character pixels on the left-top corner are lost. After a lot of observation, we found that such information loss always happens when the starting or ending point of the curve is in the middle of a boundary. Hence, as shown in Fig. 4(b), we design a more reasonable sampling strategy that simultaneously considers the boundary and the perpendicular of the curve.



(a)

(b)

Figure 4: The selected curves and sampling regions of the naive strategy and the proposed GSM. (a) The naive sampling strategy loses some information on the left-top corner. (b) The proposed GSM preserves all the useful information.

Detailedly, for the starting or ending point P' at the middle of a boundary with the tangent direction $\vec{p'}$ on the curve, we define a normalized vector $\vec{e}$ that coincides with the boundary, and $\vec{e}\odot\omega(\vec{p'}) > 0$. And for a midpoint $P_i(i+0.5<\frac{256}{3})$ with a tangent direction $\vec{p_i}$ on the curve, we define the orientation of the sampling column over P_i as follows:

$$\omega'(i,\vec{e},\vec{p_i}) = \tau\left((\frac{256}{3}-(i+0.5))\vec{e}\right) + \tau\left((i+0.5)\omega(\vec{p})\right) \ . \tag{7}$$

And vice versa, for $P_i(i+0.5>\frac{2\times256}{3})$ we have

$$\omega''(i,\vec{e},\vec{p_i}) = \tau\left(((i+0.5)-\frac{2\times256}{3})\vec{e}\right) + \tau\left((256-(i+0.5))\omega(\vec{p})\right) \ . \tag{8}$$

Consider Eq. (7) and Eq. (8) together, the coordinate of the sampling point $P_{i,j}$ at i-th column and j-th row is evaluated as follows:

$$P_{i,j} = P_i + (j-31.5)\begin{cases}\omega'(i,\vec{e},\vec{p_i}) & \text{if } i+0.5<\dfrac{256}{3}\\ \omega''(i,\vec{e},\vec{p_i}) & \text{if } i+0.5>\dfrac{2\times256}{3}\\ \omega(\vec{p_i}) & \text{otherwise}\end{cases} . \tag{9}$$

Finally, GSM samples from the input image with bilinear interpolation. The output is a 64×256 rectified image. Like CFM, GSM also has no trainable parameter.

3.4. Sequence Recognizing Module

Following [30, 3], we exploit a CNN-LSTM-attention based sequence encoder-decoder to generate character predictions. It consists of a CNN backbone, a bidirectional LSTM with 256 hidden channels, and LSTM-based attention with 512 hidden channels. The loss function of the recognizer is as follows:

$$\mathcal{L}_{SRM} = -\sum_{t=1}^{|\hat{y}|}\ln p_t(\hat{y}_t) \tag{10}$$

where p is the probability distribution sequence output by the recognizer, and $\hat{y}$ is the sequence annotation of each character in the word.

4. Performance Evaluation

4.1. Datasets

We evaluate our model on 4 general datasets (IIIT5K, SVT, IC03 and IC13) that contain mostly regular samples, and 3 special datasets (IC15, SVTP and CT80) that contain mainly irregular samples. For fair comparison, we evaluate recognition performance following the case-insensitive protocol [31, 17, 3]. Details of the datasets are as follows:

IIIT5K-Words (IIIT5K) [23] contains 3000 web images for test. Most samples are regular, and a few are curved.

Street View Text (SVT) [33] has 647 test images collected from Google Street View.

ICDAR 2003 (IC03) [21] contains 860 images of cropped words after filtering by [21].

ICDAR 2013 (IC13) [13] is the successor of IC03, it inherits most of IC03's data and contains 1015 for test after removing those containing non-alphanumeric characters.

ICDAR 2015 (IC15) [12] contains 2077 images for test. These images are collected via a pair of Google Glasses without careful positioning and focusing.

SVT-Perspective (SVTP) [26] contains 645 cropped images for test. Images are picked from side-view angle snapshots in Google Street View. Therefore, there may be severe perspective distortion in these images.

CUTE80 (CT80) [27] is collected for evaluating curved text recognition. It contains 288 cropped natural images for test.

Some of the aforementioned datasets associate lexicons to simplify the recognition task in early years. However, the lexicons do not help in most cases of real-world applications. And recent methods achieve nearly saturated performance with such constraint. Hence, we do not use these lexicons in our performance evaluation.

The datasets used to train the model are SynthText [6], Synth90K [8] and a collection of real-world images. SynthText contains about 8M synthetic words with character-level annotations. Synth90K are generated based on 90k generic English words, and contains about 9M synthetic words without character-level annotations. We also collect real-world images from public datasets following [17], resulting in about 50K real-world samples.

4.2. Implementation Details

Preprocessing. In both training and test stages, the input images are resized to 64×256, 96×192, 128×128, 192×96 or 256×64 according to the original aspect ratios. Formally, with the original height H and width W, we set the resized aspect ratio to

$$\frac{H'}{W'} = 2^{\max\{\min\{\lfloor\log_2(\frac{H}{W})+0.5\rfloor,2\},-2\}} \ . \tag{11}$$

Annotations for DOEM. The SynthText dataset has bounding boxes for each character. For a character and its bounding box, we use the vector that starts from the center of the left boundary and ends at the center of the right boundary as the orientation of the character. For the pixels inside a character's bounding box, the ground-truth of the character's densities are labeled as 1, and the ground-truth of the character's orientations are labeled as the normalized orientation of the corresponding character. And for the pixels outside any character's bounding box, the ground-truth of the character's densities are labeled as 0, and the ground-truth of the character's orientations are ignored.

Approach	Method	IIIT5k	SVT	IC03	IC13	IC15	SVTP	CT80
Unrectified sequence to sequence	R^{2}AM [16]	78.4	80.7	88.7	90.0	-	-	-
	CRNN [28]	81.2	82.7	81.9	89.6	-	-	-
	FAN [3]	87.4	85.9	94.2	93.3	70.6	-	-
	EP [2]	88.3	87.5	94.6	**94.4**	73.9	-	-
	Liu *et al.* [20]	89.4	87.1	94.7	94.0	-	73.9	62.5
2D-decoder based	SAR [17]	95.0	91.2	-	94.0	**78.8**	**86.4**	89.6
	CA-FCN [18]	91.9	86.4	-	91.5	-	-	79.9
	Yang *et al.* [35]	-	-	-	-	-	75.8	69.3
Rectification based	RARE [30]	81.9	81.9	90.1	88.6	-	71.8	59.2
	AON [4]	87.0	82.8	91.5	-	68.2	73.0	76.8
	Char-Net [19]	92.0	85.5	92.0	91.1	74.2	78.9	-
	ASTER [31]	93.4	89.5	94.5	91.8	76.1	78.5	79.5
	ScRN [34]	94.4	88.9	95.0	93.9	78.7	80.8	87.5
	ESIR [37]	93.3	90.2	-	91.3	-	79.6	83.3
	RPI$^-$ (ours)	93.0	88.9	93.7	91.5	69.4	77.7	88.5
	RPI (ours)	**95.1**	**91.7**	**96.0**	92.9	78.1	84.8	**91.7**

Table 1: Recognition accuracies on seven benchmarks in percent while $32\lambda = 4$. RPI$^-$ is trained with only SynthText and Synth90K.

Backbones of DOEM and SRM. The backbone of DOEM is a down-sample-up-sample pyramid architecture. The down-sample part consists of two 3×3 convolutions and eight ResBlocks. Each convolution is followed by a 2×2 max-pooling. The stride is 2×2 for the 3-rd, 5-th and 7-th ResBlocks, and 1×1 for the others. The convolutions output 32 and 64 channels respectively and the ResBlocks output 64, 64, 128, 128, 256, 256, 256 and 256 channels respectively. The up-sample part applies stage-by-stage nearest neighbor up-sampling to the outputs of the 8-th, 6-th, 4-th and 2-nd ResBlocks, each of which has a 1×1 skip connection. Finally, a 3×3 convolution with 3 channels is used to generate the output. The backbone of SRM is the same as that used in [3], except for an extra 2×1 max-pooling that is inserted before $conv5\_x$.

Model training. We first train DOEM with SynthText. The training samples are randomly rotated within ±90°. CFM, GSM and the trained DOEM are then used to rectify samples in SynthText. Synth90K and real-world data are used to train SRM. Both DOEM and SRM are trained by Adam optimizer [14] and converge after 5 epochs. The learning rate is set to 10^{-3} in the 1-*st* epoch, then decays smoothly to 10^{-5} from the 2-*nd* epoch to the 4-*th* epoch, and finally keeps at 10^{-5} for the 5-*th* epoch. The hyper-parameter λ is set to $\frac{4}{32}$ in training. Our model recognizes 95 symbols including 10 digits, 26 uppercase letters, 26 lowercase letters, 32 punctuation marks and an EOS symbol (the end-of-sequence).

Environments. This work is implemented with PyTorch-1.4 [25]. The training and evaluation are accelerated by an NVIDIA TITAN RTX GPU.

4.3. Comparison with the State of the Art Methods

We train RPI with SynthText, Synth90K, and the collected real-world data. Besides, we also train RPI$^-$ with only SynthText and Synth90K. The results are presented in Tab. 1.

Comparison with rectification-based methods. The rectification-based methods listed in Tab. 1 are trained with only synthetic data. Therefore, we compare RPI$^-$ with them. From Tab. 1 we can see that RPI$^-$ wins all the other rectification-based methods on CT80 (the dataset specifically built for evaluating curved text recognition), and performs comparably with them on IIIT5K, SVT, IC03, IC13, and SVTP, but is worse than 3 of them on IC15.

Comparison with 2D-decoder based methods. SAR is the most representative 2D-decoder-based method, and note that SAR is trained with real data and extra SynthAdd data. Therefore, we compare RPI with SAR. We can see that RPI wins SAR on CT80, IIIT5K, and SVT, and is comparable to SAR on the other three benchmarks (IC13, IC15, and SVTP).

Performance comparison summary and discussion. According to the results in Tab. 1, our method performs better than all the existing methods on IIIK50, IC03, SVT, and CT80. Note that CT80 contains mainly curved data, and our method achieves a 2.1% advantage in recognition accuracy over the 2-nd best method SAR on CT80. However, we also see that RPI does not perform so well on perspective data, especially when trained without real-world data. This indicates that real-world data are important to our approach. We analyze this phenomenon and attribute it to the difference of irregularity patterns between synthetic data and real-world data. We will try to improve RPI based on these findings in the future.

By analyzing some unrecognized samples, we found that our model also suffers attention drift like most attention-based sequence to sequence models, which means that the focusing mechanism [3] and semantic segmentation can help our method.

4.4. The Effect of λ

λ is a hyper-parameter that influences the performance of curve selection. We conduct experiments to evaluate the impact of λ on performance. The results are given in Tab. 2. We can see that our method performs best when λ takes value from $\frac{2}{32}$ to $\frac{8}{32}$. But the exceptions are IC13 and IC15, on which we get the best accuracy when λ is $\frac{16}{32}$. Generally, when λ is too small, many pixels cannot be covered by the sampled points used for scoring. On the contrary, a too-large λ will lead to the decreasing of scoring resolution because many sampled points are overlapped among different curves.

5. Conclusion

In this work, we try to read irregular scene texts by rectifying their principle irregularities. We propose the RPI method that samples feature sequences from input images based on Bézier curve fitting. RPI can be implemented in any sequence to sequence scene text recognition model. Experiment results show the advantage and robustness of the proposed method. RPI's outstanding performance on irregular samples conforms to our expectations. In the future, we will continue to improve RPI and make it effective in more complex situations, and design an end-to-end mechanism to train the model without character-level annotations. We will also try to extend this idea to end-to-end text spot tasks and applications.

Acknowledgement

This work was supported by Shanghai Artificial Intelligence Innovation and Development Projects funded by Shanghai Municipal Commission of Economy and Informatization. Yang Wang and Jihong Guan were supported by National Natural Science Foundation of China (NSFC) under grant No. U1936205.

References

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *ICLR*, 2015.

[2] Fan Bai, Zhanzhan Cheng, Yi Niu, Shiliang Pu, and Shuigeng Zhou. Edit probability for scene text recognition. In *CVPR*, pages 1508–1516, 2018.

[3] Z. Cheng, F. Bai, Y. Xu, G. Zheng, S. Pu, and S. Zhou. Focusing attention: Towards accurate text recognition in natural images. In *ICCV*, pages 5086–5094, 2017.

[4] Z. Cheng, Y. Xu, F. Bai, Y. Niu, S. Pu, and S. Zhou. AON: towards arbitrarily-oriented text recognition. In *CVPR*, pages 5571–5579, 2018.

[5] Alex Graves, Santiago Fernández, Faustino J. Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *ICML*, pages 369–376, 2006.

[6] Ankush Gupta, Andrea Vedaldi, and Andrew Zisserman. Synthetic data for text localisation in natural images. In *CVPR*, pages 2315–2324, 2016.

[7] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, June 2016.

[8] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Synthetic data and artificial neural networks for natural scene text recognition. *CoRR*, abs/1406.2227, 2014.

[9] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep structured output learning for unconstrained text recognition. In *ICLR*, 2015.

[10] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Reading text in the wild with convolutional neural networks. *IJCV*, 116(1):1–20, 2016.

[11] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. In *NIPS*, pages 2017–2025, 2015.

[12] Dimosthenis Karatzas, Lluis Gomez-Bigorda, Anguelos Nicolaou, Suman K. Ghosh, Andrew D. Bagdanov, Masakazu Iwamura, Jiri Matas, Lukas Neumann, Vijay Ramaseshan Chandrasekhar, Shijian Lu, Faisal Shafait, Seiichi Uchida, and Ernest Valveny. ICDAR 2015 competition on robust reading. In *ICDAR*, pages 1156–1160, 2015.

[13] Dimosthenis Karatzas, Faisal Shafait, Seiichi Uchida, Masakazu Iwamura, Lluis Gomez i Bigorda, Sergi Robles Mestre, Joan Mas, David Fernández Mota, Jon Almazán, and Lluís-Pere de las Heras. ICDAR 2013 robust reading competition. In *ICDAR*, pages 1484–1493, 2013.

[14] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ICLR*, 12 2014.

[15] C. Lee and S. Osindero. Recursive recurrent nets with attention modeling for ocr in the wild. In *CVPR*, pages 2231–2239, June 2016.

[16] Chen-Yu Lee and Simon Osindero. Recursive recurrent nets with attention modeling for ocr in the wild. *CVPR*, pages 2231–2239, 2016.

[17] Hui Li, Peng Wang, Chunhua Shen, and Guyu Zhang. Show, attend and read: A simple and strong baseline for irregular text recognition. In *AAAI*, pages 8610–8617, 2019.

[18] Minghui Liao, Jian Zhang, Zhaoyi Wan, Fengming Xie, Jiajun Liang, Pengyuan Lyu, Cong Yao, and Xiang Bai. Scene text recognition from two-dimensional perspective. In *AAAI*, pages 8714–8721, 2019.

32λ	IIIT5K	SVT	IC03	IC13	IC15	SVTP	CT80
2	95.0	**91.7**	**96.0**	92.9	78.0	**84.8**	90.6
4	95.1	**91.7**	**96.0**	92.9	78.1	**84.8**	**91.7**
6	**95.3**	**91.7**	95.9	92.9	78.1	**84.8**	**91.7**
8	**95.3**	91.5	95.9	92.9	78.2	**84.8**	91.3
10	95.1	91.5	95.9	**93.0**	78.3	84.7	91.0
12	95.0	91.5	95.9	**93.0**	78.4	84.5	91.3
14	95.0	91.5	95.9	**93.2**	78.3	84.5	90.3
16	94.8	91.5	95.8	**93.2**	**78.5**	84.3	89.9

Table 2: Performance on seven benchmarks when applying different λ values.

[19] Wei Liu, Chaofeng Chen, and Kwan-Yee K. Wong. Charnet: A character-aware neural network for distorted scene text recognition. In *AAAI*, pages 7154–7161, 2018.

[20] Yang Liu, Zhaowen Wang, Hailin Jin, and Ian J. Wassell. Synthetically supervised feature learning for scene text recognition. In *ECCV*, pages 449–465, 2018.

[21] Simon M. Lucas, Alex Panaretos, Luis Sosa, Anthony Tang, Shirley Wong, and Robert Young. ICDAR 2003 robust reading competitions. In *ICDAR*, pages 682–687, 2003.

[22] Canjie Luo, Lianwen Jin, and Zenghui Sun. A multi-object rectified attention network for scene text recognition. *Pattern Recognition*, 90:109–118, 2019.

[23] Anand Mishra, Karteek Alahari, and C. V. Jawahar. Scene text recognition using higher order language priors. In *BMVC*, pages 1–11. BMVA Press, 2012.

[24] L. Neumann and J. Matas. Real-time scene text localization and recognition. In *CVPR*, pages 3538–3545, June 2012.

[25] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8026–8037. Curran Associates, Inc., 2019.

[26] Trung Quy Phan, Palaiahnakote Shivakumara, Shangxuan Tian, and Chew Lim Tan. Recognizing text with perspective distortion in natural scenes. In *ICCV*, pages 569–576, 2013.

[27] Anhar Risnumawan, Palaiahnakote Shivakumara, Chee Seng Chan, and Chew Lim Tan. A robust arbitrary text detection system for natural scene images. *Expert Syst. Appl.*, 41(18):8027–8048, 2014.

[28] B. Shi, X. Bai, and C. Yao. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *TPAMI*, 39(11):2298–2304, Nov 2017.

[29] B. Shi, X. Bai, and C. Yao. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *TPAMI*, 39(11):2298–2304, Nov 2017.

[30] Baoguang Shi, Xinggang Wang, Pengyuan Lyu, Cong Yao, and Xiang Bai. Robust scene text recognition with automatic rectification. In *CVPR*, pages 4168–4176, 2016.

[31] B. Shi, M. Yang, X. Wang, P. Lyu, C. Yao, and X. Bai. Aster: An attentional scene text recognizer with flexible rectification. *TPAMI*, 41(9):2035–2048, Sep. 2019.

[32] Zhaoyi Wan, Mingling He, Haoran Chen, Xiang Bai, and Cong Yao. Textscanner: Reading characters in order for robust scene text recognition. *CoRR*, abs/1912.12422, 2019.

[33] Kai Wang, Boris Babenko, and Serge J. Belongie. End-to-end scene text recognition. In *ICCV*, pages 1457–1464, 2011.

[34] Mingkun Yang, Yushuo Guan, Minghui Liao, Xin He, Kaigui Bian, Song Bai, Cong Yao, and Xiang Bai. Symmetry-constrained rectification network for scene text recognition. In *ICCV*, October 2019.

[35] Xiao Yang, Dafang He, Zihan Zhou, Daniel Kifer, and C. Lee Giles. Learning to read irregular text with attention mechanisms. In *IJCAI*, pages 3280–3286, 2017.

[36] Cong Yao, Xiang Bai, Baoguang Shi, and Wenyu Liu. Strokelets: A learned multi-scale representation for scene text recognition. In *CVPR*, pages 4042–4049, 2014.

[37] Fangneng Zhan and Shijian Lu. Esir: End-to-end scene text recognition via iterative image rectification. In *CVPR*, June 2019.