NRTR: A No-Recurrence Sequence-to-Sequence Model For Scene Text Recognition

Fenfen Sheng
Institute of Automation, Chinese Academy of Sciences
University of Chinese Academy of Sciences
Beijing 100190, China
shengfenfen2015@ia.ac.cn

Zhineng Chen
Institute of Automation, Chinese Academy of Sciences
Beijing 100190, China
zhineng.chen@ia.ac.cn

Bo Xu
Institute of Automation, Chinese Academy of Sciences
Beijing 100190, China
xubo@ia.ac.cn

***Abstract*—Scene text recognition has attracted a great many researches due to its importance to various applications. Existing methods mainly adopt recurrence or convolution based networks. Though have obtained good performance, these methods still suffer from two limitations: slow training speed due to the internal recurrence of RNNs, and high complexity due to stacked convolutional layers for long-term feature extraction. This paper, for the first time, proposes a no-recurrence sequence-to-sequence text recognizer, named NRTR, that dispenses with recurrences and convolutions entirely. NRTR follows the encoder-decoder paradigm, where the encoder uses stacked self-attention to extract image features, and the decoder applies stacked self-attention to recognize texts based on encoder output. NRTR relies solely on self-attention mechanism thus could be trained with more parallelization and less complexity. Considering scene image has large variation in text and background, we further design a modality-transform block to effectively transform 2D input images to 1D sequences, combined with the encoder to extract more discriminative features. NRTR achieves state-of-the-art or highly competitive performance on both regular and irregular benchmarks, while requires only a small fraction of training time compared to the best model from the literature (at least 8 times faster).**

***Keywords*-No-Recurrence; Self-attention; Modality-transform block; Faster and better**

I. Introduction

Scene text recognition has drawn increasing interests as it could extract rich semantic information relevant to scene and object. Although extensive studies have been carried out, recognizing scene texts is still challenging due to its high complexity, e.g., low quality texts, arbitrary orientations, cluttered backgrounds and complex deformations, see Fig.1.

Current text recognition methods [1], [2], [3], [4], [5], [6] mainly follow the sequence-to-sequence (seq2seq) paradigm, where input images and output texts are separately represented as patch sequences and character sequences. These methods could be roughly classified into two branches: the recurrent neural network (RNN) based recognizers and the convolutional neural network (CNN) based ones.

RNN based text recognizers [1], [3], [2], [6], [7] have show great success as they are superior to learn contextual information and capture strong correlation among different characters in each text. However, the inherently sequential nature of RNN precludes computation parallelization, which brings heavy time and computational burdens when input image sequence is long, as memory constraints limit batching across examples. Besides, the training procedure of RNN is sometimes tricky due to the problem of gradient vanishing/exploding [8].

Figure 1: Qualitative results of NRTR. (left) Correct results with texts in yellow. (right) Incorrect results with labels in yellow and outputs in red.

Recently, CNN based recognizers [4], [5] are proposed to accelerate sequential computation. By leveraging CNN instead of RNN, they enable to compute hidden representation in parallel. However, the number of operations required to relate two arbitrary signals grows along with their distances. CNN based methods are difficult to learn dependencies among distant positions, unless much more convolutional layers are stacked, which in turn increases the complexity. Therefore, CNN based methods suffer from the dilemma of low complexity and satisfactory performance.

In this paper, we propose, for the first time, a no-recurrence seq2seq scene text recognizer, named NRTR, that dispenses with recurrences and convolutions entirely. Motivated by recent success of Transformer [9] in natural language processing field, NRTR relies solely on

2379-2140/19/$31.00 ©2019 IEEE
DOI 10.1109/ICDAR.2019.00130

self-attention mechanism. Specifically, NRTR follows the encoder-decoder framework, while the encoder uses stacked self-attention to transform input image sequence to hidden feature representation, and the decoder applies stacked self-attention to output sequence of characters based on the encoder output. NRTR draws global dependencies between different input and output positions at once rather than one by one in RNN, and reduces the whole operation to a constant number unlike that in CNN. It therefore allows for more computation parallelization with higher performance.

Besides, unlike [9] that uses 1D sentence as input, scene text recognizer receives 2D images with large variation in scales/aspect ratios and backgrounds. We further proposes a novel modality-transform block as a preprocessing step before the encoder, to effectively convert 2D image to corresponding 1D sequence. Experiments demonstrate that the specially designed modality-transform block could greatly influence the whole recognizer performance.

We conduct extensive experiments on standard benchmarks, including both regular (IIIT5K, SVT, ICDAR2003 and ICDAR2013) and irregular datasets (SVT-P, CUTE80, ICDAR2015). Without the bells and whistles, NRTR achieves state-of-the-art or highly competitive performance in both lexicon-free and lexicon-based cases without any rectification module, while accompanies with a 8 times faster training speed than the existing best recognizer.

Our contributions are summarized as follows:

- We analysis, for the first time, the deficiency of current RNN and CNN based recognizers and propose a no-recurrence seq2seq model based solely on self-attention mechanism. Our model could be trained with more computation parallelization and less complexity.
- We design a modality-transform block to efficiently map input image to corresponding sequence, combined with encoder, to extract more discriminative features.
- Our model achieves state-of-the-art performance on various benchmarks and significantly surpasses competitive recognizers on both accuracy and training speed.

II. Related Work

A. Scene Text Recognition

Traditional text recognizers mainly adopt bottom-up scheme by first detecting individual characters using sliding window [10], then integrating characters into texts by dynamic programming or lexicon search [10]. Others adopt top-down scheme by directly recognizing texts from images.

Recent methods regard text recognition as a sequence recognition problem where images and texts are represented as patch and character sequences separately. Shi et al. [3] combine CNN and RNN to learn spatial dependencies and applies CTC to translate per-slice prediction into a label sequence. They also develop an attention-based spatial transformer network to rectify irregular texts [6]. Besides, Lee et al. [11] and Cheng et al. [2] both construct attention-based recurrent network to decode feature sequence and predict labels recurrently. Instead of RNNs, Gao et al. [4] and Yin et al. [5] leverage stacked CNN for the pursuit of greater computational parallelism.

B. Our Method Versus Some Related Works

The most related work to NRTR is Transformer, a recent development in NLP field. Inspired by Transformer, we uses solely self-attention mechanism as the fundamental module but has distinct differences indeed. First, Transformer aims for machine-based English-to-French translation task but fails to read texts in natural images. As input text images generally convey much more information than sentences in machine translation, text recognizers tend to be more complicated. Our model relies on the specifically designed encoder/decoder to efficiently solve this problem. Second, unlike Transformer that receives 1D sentences as input, we use 2D images with large variation in scales/aspect ratios and backgrounds. The locations and geometric features of scene texts lying in images play a more complex role than the word location in an 1D sentence. To obtain the most useful sequence for the encoder, we design a novel modality-transform block to transform each input image effectively.

III. Methodology

The architecture of NRTR is depicted in Fig.2. NRTR consists of three sub-networks: the encoder, the decoder and the modality-transform block served as the preprocessing. As the encoder and the decoder are both based on the self-attention mechanism, we first review it and then describe the three main sub-networks.

A. Self-Attention Mechanism

Self-attention extracts correlation information between different input and output positions. Here, we use Scaled Dot-Product Attention, an effective self-attention module proposed in [9]. It has three inputs: queries and keys of dimension d_k, and values of dimension d_v. Dot product is performed between the query and all keys to obtain their similarity. A softmax function is applied to obtain the weights on the values. Given a query $\mathbf{q}$, all keys (packed into matrices $\mathbf{K}$) and values (packed into $\mathbf{V}$), the output value is weighted average over input values:

$$v^{out} = \mathrm{softmax}\left(\frac{\mathbf{q}\mathbf{K}^t}{\sqrt{d_k}}\right)\mathbf{V} \tag{1}$$

where t means element numbers of corresponding inputs and scalar $\frac{1}{\sqrt{d_k}}$ is used to prevent softmax into regions where it has extremely small gradients. Therefore, self-attention could connect all positions with a constant number and allow parallelization. More details please refer to [9].

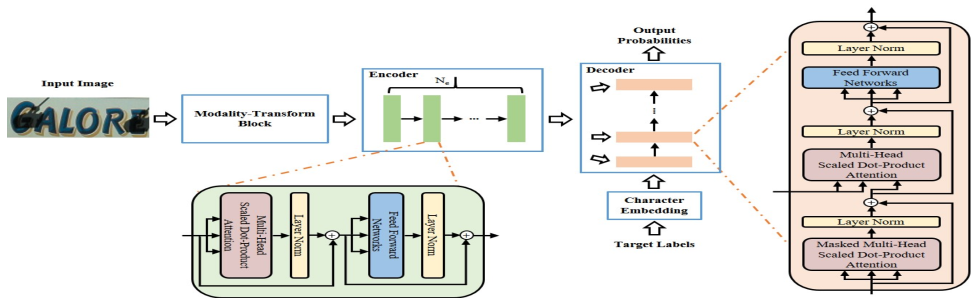

Figure 2: The overall architecture of NRTR.

B. Modality-Transform Block

Modality-transform block consists of several convolutional layers. For each layer, the stride is set to 2 and channel number is increased progressively by $2\times$. The product of the height and channel number of each layer therefore remains constant and equal to d_{model}, the dimension used in our encoder-decoder model. More specifically, for each image with (w_0, h_0), at the n-th layer, we get $(w, h, c) = \left({}^{w_0}/_{2^n}, {}^{h_0}/_{2^n}, \left({}^{d_{\text{model}}}/_{h_0}\right) \times 2^n\right)$, where c means channel number. After the final layer, a concatenate operation is applied to reshape features from different channels into an input sequence $(input\_step, dim) = \left({}^{w_0}/_{2^n}, d_{\text{model}}\right)$, each element of which has d_{model} dimensions. We design various block architectures to observe their influences on the whole model. More information is detailed in the experiment part.

Additionally, as NRTR contains no recurrences, we use positional encoding to indicate each position in sequence:

$$PE_{(pos,i)} = \begin{cases} sin(pos/10000^{2i/d_{\text{model}}}) & 0 \leq i \leq d_{\text{model}}/2 \\ cos(pos/10000^{2i/d_{\text{model}}}) & d_{\text{model}}/2 \leq i \leq d_{\text{model}} \end{cases} \tag{2}$$

where pos indicates the position in input image sequence and i indicates the i-th dimension. We choose this function since for arbitrary fixed offset k, PE_{pos+k} can be represented as a linear function of PE_{pos}. We get the final input sequence by adding the positional encoding to the above input sequence.

C. Encoder

The encoder consists of N_e number of connected identical encoder-blocks (green block in Fig.2), each of which contains two sub-layers: a multi-head scaled dot-product attention and a position-wise fully connected network.

The multi-head scaled dot-product attention allows the encoder to jointly attend to information from different representation subspaces at different positions. Similar to a convolutional layer that applies a set of filters to extract various features, multi-head attention stacks h times scaled dot-product attention, where h is called the head number. The entire process of the multi-head scaled dot-product attention includes three operations. Firstly, each scaled dot-product attention goes through three different linear projections to project the queries, keys, and values from the input sequence to more discriminative representations. Then, the stacked h times scaled dot-product attentions are performed in parallel, and finally, their outputs are concatenated and undergo a linear layer to get the final d_{model}-dimensional outputs:

$$\text{MultiHead}\left(\mathbf{Q}, \mathbf{K}, \mathbf{V}\right) = \text{Concat}\left(\text{head}_1, \ldots, \text{head}_\text{h}\right) \mathbf{W}^O \tag{3}$$

$$\text{where}, \text{head}_\text{i} = \text{Attention}\left(\mathbf{Q}\mathbf{W}_i^Q, \mathbf{K}\mathbf{W}_i^K, \mathbf{V}\mathbf{W}_i^V\right) \tag{4}$$

Since $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ have the same dimension of d_{model}, the predictions are parameter matrices $\mathbf{W}_i^Q \in \Re^{d_{\text{model}} \times d_q}$, $\mathbf{W}_i^K \in \Re^{d_{\text{model}} \times d_k}$, $\mathbf{W}_i^V \in \Re^{d_{\text{model}} \times d_v}$ and $\mathbf{W}_i^O \in \Re^{d_c \times d_{\text{model}}}$, where $d_c = h \times d_v$, and we set $d_q = d_k = d_v = d_{\text{model}}$.

The position-wise fully connected network consists of two linear transformations with a RELU activation in between.

$$\text{FFN}\left(x\right) = \max\left(0, x\mathbf{W}_1 + b_1\right)\mathbf{W}_2 + b_2 \tag{5}$$

where the weights are $\mathbf{W}_1 \in \Re^{d_{\text{model}} \times d_{ff}}$ and $\mathbf{W}_2 \in \Re^{d_{ff} \times d_{\text{model}}}$, and the bias are $b_1 \in \Re^{d_{ff}}$ and $b_2 \in \Re^{d_{\text{model}}}$. The linear transformations are the same across different positions, but use different parameters from layer to layer.

Meanwhile, layer normalization and residual connection are introduced into each sub-layer for effective training. Given each sub-layer x, the corresponding outputs are:

$$LayerNorm\left(x + Sublayer\left(x\right)\right) \tag{6}$$

D. Decoder

The decoder generate text sequence based on encoder outputs and input labels. For each input label, we apply a learnable character-level embedding to convert per character to a d_{model}-dimensional vector. The resulted vectors combines with the positional encoding to form the decoder input.

The decoder consists of N_d number of connected identical decoder-blocks (orange block in Fig.2). Similar to the encoder, the decoder-block is based on the multi-head scaled dot-product attention and the position-wise fully connected network, but has two differences. Firstly, due to the autoregressive property, a masked multi-head attention is added to each decoder-block to ensure that the predictions for position j can only depend on the known outputs prior to

j. We implement this by masking out (setting to $-\infty$) all values in the input of the softmax which correspond to illegal connections. Second, the multi-head attention has keys and values coming from the encoder outputs, and queries coming from the previous decoder block outputs.

The outputs are transformed to the probabilities for character classes by a linear projection and a softmax function.

IV. Experiment

In this section, we describe the standard benchmarks, the detailed experimental settings and results with comparisons.

A. Benchmark Datasets

IIIT5K [12] contains 3000 text images in test set. Each is accompanied with a 50-word and a 1k-word lexicons.
SVT [10] is collected from Google Street View and most images are severely corrupted by noise and blur. It contains 647 text images, each of which has a 50-word lexicon.
ICDAR 2003 (IC03) [13] contains 1065 cropped text images after data filtering as in [10]. Each image is associated with a 50-word and a full lexicon as defined in [10].
ICDAR 2013 (IC13) [14] includes texts on sign boards and objects with large variations. After filtered as done in IC03, it finally contains 1015 cropped text images in test set.
ICDAR 2015 (IC15) [15] is taken from Google Glasses and contains plenty of irregular texts. For fair comparison, we discard the images that contain non-alphanumeric characters and finally obtain 1922 ones. No lexicon is specified.
SVT-P [16] contains 645 cropped text images captured from the side-view angles in Google Street View. Each image is specified with a 50 words lexicon and a full lexicon.
CUTE80 [17] is specially collected for evaluating the performance of curved text recognition. It contains 287 cropped text images in test set. No lexicon is provided.

B. Implementation Details

NRTR is trained purely on Synth90k [18] and evaluated on standard benchmarks without any finetuning. In both training and inference, heights of input images are set to 32 and widths are proportionally scaled. The output consists of 38 classes, including 26 lowercase letters, 10 digital numbers, 1 space and 1 end-of-sequence token.

At training, samples are batched together by approximate image widths. We use Adam with $\beta_1 = 0.9$, $\beta_2 = 0.98$, $\epsilon = 10^{-9}$, and vary learning rate according to the formula:

$$lrate = d_{model}^{-0.5} \cdot \min\left(n^{-0.5}, n \cdot warmup\_n^{-1.5}\right) \quad (7)$$

where n represents current step and $warmup\_n$ (set to 16000) controls over the learning rate first increase then decrease. In order to prevent over-fitting, we set residual dropout to 0.1. We train NRTR for about 6 epochs before convergence and average the last 10 checkpoints for inference. All the experiments are implemented in Tensorflow with one Titan X GPU.

Model	N_e	N_d	d_{ff}	IIIT5K	SVT	IC03	IC13
6enc6dec (base)	6	6	1024	85.4	86.8	93.5	92.8
12enc6dec-4096 (big)	12	6	4096	86.5	88.3	95.4	94.7
8enc4dec	8	4	1024	85.7	86.6	93.7	93.9
4enc8dec	4	8	1024	85.2	86.5	93.0	93.2
10enc5dec	10	5	1024	85.9	87.2	93.9	94.2
12enc6dec	12	6	1024	86.2	87.7	94.5	94.2
6enc6dec-2048	6	6	2048	85.9	87.4	94.2	93.9
6enc6dec-4096	6	6	4096	86.3	87.5	95.1	93.9
base model with 2Conv	6	6	1024	85.4	86.8	93.5	92.8
base model with 3Conv	6	6	1024	84.2	86.6	93.7	92.8
base model with 7Conv	6	6	1024	77.4	80.7	89.1	89.5
base model with 2CNNLSTM	6	6	1024	85.5	86.1	94.4	94.9
big model with 2Conv	6	6	1024	86.5	88.3	95.4	94.7
big model with 2CNNLSTM	6	6	1024	84.7	84.8	94.1	93.4

Table I: *Exploration of the encoder, the decoder and the modality-transform blocks on lexicon-free benchmarks.*

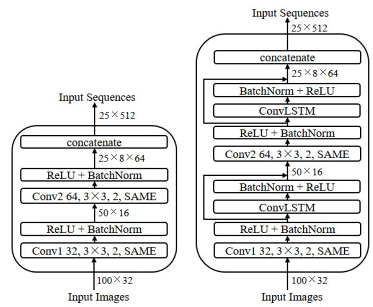

Figure 3: Examples of the proposed modality-transform block. (left) The general CNN block. (right) The CNNLSTM block.

C. Ablation Study

We first investigate the configuration of three sub-networks in NRTR. All experiments are executed with same training strategy and evaluated under the lexicon-free case.

1) Exploration of the encoder and the decoder: We explore the encoder-block number N_e, the decoder-block number N_d and the fully connected inner dimension d_{ff}. We set $d_{model} = 512$, $h = 8$ and the modality-transform block with two convolutional layers during these experiments. As listed in Tab.I, we take the 6enc6dec model as our baseline. We first keep total block number identical and find that more encoder blocks achieve better accuracy (8enc4dec vs 6enc6dec, 4enc8dec). Then, we add more blocks and see that deeper model obtains higher accuracy (12enc6dec vs 8enc4dec, 10enc5dec). We no longer increase N_e and N_d considering time and memory costs. These two comparisons indicate that deeper encoder/decoder could extract more representation, while image information is tend to be more complex than target ones (8enc4dec vs 4enc8dec). We further test different d_{ff} and observe that wider inner dimension is more beneficial to NRTR (6enc6dec-4096 vs 6enc6dec-2048,6enc6dec). Based on the above analysis, we take the 12enc6dec-4096 as our big model.

2) Exploration of the modality-transform block: We investigate various architectures and depict two examples in Fig.3. As listed in Tab.I, more convolutional layers (2Conv vs 3Conv, 7Conv) lead to a decline in performance, even a seven-layer convnet used in CRNN [3] and RARE [6].

Methods	Regular Text									Training Time
	IIIT5K			SVT		IC03			IC13	h/GPU
	50	1k	None	50	None	50	Full	None	None	
ABBYY[10]	24.3	-	-	35.0	-	56.0	55.0	-	-	-
Wang et al.[10]	-	-	-	57.0	-	76.0	62.0	-	-	-
Mishra et al.[12]	64.1	57.5	-	73.2	-	81.8	67.8	-	-	-
Goel et al.[21]	-	-	-	77.3	-	89.7	-	-	-	-
Bissacco et al.[22]	-	-	-	90.4	78.0	-	-	-	87.6	-
Alsharif et al.[23]	-	-	-	74.3	-	93.1	88.6	-	-	-
Almázan et al.[24]	91.2	82.1	-	89.2	-	-	-	-	-	-
Yao et al.[25]	80.2	69.3	-	75.9	-	88.5	80.3	-	-	-
Jaderberg et al.[26]	-	-	-	86.1	-	96.2	91.5	-	-	-
Su and Lu et al.[27]	-	-	-	83.0	-	92.0	82.0	-	-	-
Jaderberg et al.[28]	97.1	92.7	-	95.4	80.7	98.7	**98.6**	93.1	90.8	-
Lee et al.[11]	96.8	94.4	-	96.3	80.7	97.9	97.0	88.7	90.0	-
Shi et al.[6]	96.2	93.8	81.9	95.5	81.9	98.3	96.2	90.1	88.6	16/Titan X
Shi et al.[3]	97.6	94.4	78.2	96.4	80.8	98.7	97.6	89.4	86.7	-
Ghosh et al.[29]	-	-	-	95.2	80.4	95.7	94.1	92.6	-	-
Yin et al.[5]	98.9	96.7	81.6	95.1	76.5	97.7	96.4	84.5	85.2	-
Gao et al.[4]	99.1	97.9	-	97.4	82.7	98.7	96.7	89.2	88.0	-
Cheng et al.*[2]	99.3	97.5	87.4	97.1	85.9	**99.2**	97.3	94.2	93.3	40/M40
Bai et al.*[1]	**99.5**	97.9	88.3	96.6	87.5	98.7	97.9	94.6	94.4	41/P40
Our proposed NRTR	99.2	98.4	86.5	98.0	88.3	98.9	97.9	**95.4**	94.7	2.8/Titan X
Our proposed NRTR*	99.2	**98.8**	**90.1**	**98.1**	**91.5**	98.9	98.0	94.7	**95.8**	5.0/Titan X

Table II: *Accuracies (%) on regular benchmarks. "50", "1k" and "Full" are lexicon sizes. "None" means the lexicon-free case. 'h/GPU' indicates training time cost per epoch on their GPUs. Note that the FLOPS is* $P40 > M40 \approx TitanX$. * *means training with both Synth90k and SynthText.*

Methods	Irregular Text			
	IC15	SVT-P		CUTE80
	None	50	None	None
AON*[30]	68.2	94.0	73.0	76.8
Aster*[31]	76.1	-	78.5	79.5
Liao et al.*[32]	-	-	-	79.9
SAR[33]	78.8	**95.8**	86.4	**89.6**
Our proposed NRTR*	**79.4**	94.9	**86.6**	80.9

Table III: *Accuracies (%) on irregular benchmarks. * means training with both Synth90k and SynthText.*

We conjecture that the loss of detailed information due to resolution subsampling outweighs the gain of high-level semantics. Since the encoder itself has strong feature extraction ability, we prefer to apply a two-layer convolution in the block and combine it with the encoder to draw more discriminative features. Besides, as the newly CNNLSTM [19] could captures more temporal information by recurrent connections, we replace CNN with it. Results show an accuracy boost in our base model, but a little reduction in our big model. The reason we guess is the redundant extraction of image information when associates CNNLSTM with excessive encoder components.

D. Comparisons with the State-of-the-arts

Based on the above analysis, we construct the final NRTR by setting $N_e = 12$, $N_d = 6$, $d_{ff} = 4096$ and the modality-transform block with two convolutional layers. Since recent methods are trained with both Synth90k [18] and SynthText [20], for fair comparison, we further train NRTR on two synthetic datasets. Quantitative results are listed in Tab.II.

1) Accuracy: For regular benchmarks, NRTR shows an accuracy boost and averagely beats previous best recognizers [2], [1] in both lexicon-free and lexicon-based cases. Specifically, NRTR obtains 0.9% on IIIT5K with 1k lexicon, 1.5% on SVT and 0.2% on IC03 with Full lexicon, only litter decline on IIIT5K (0.3%) with50 lexicons. In the lexicon-free case, NRTR surpasses [1] on all benchmarks.

Moreover, we test NRTR on irregular benchmarks and show results in Tab.III. Note that comparative models are all designed specially for irregular texts. NRTR does not perform any special operation but still shows great tolerance on handling irregular texts, which further illustrates its strong ability in text feature extraction.

2) Speed: We give training speed of these approaches in Tab.II. Only a few methods report their training time. For each epoch, Shi et al.[6] costs 16 hours/Titan X and Cheng et al.[2] needs 40 hours/Tesla M40. Both need 3 epochs before converging. The best previous method Bai et al.[1] takes more time (41 hours per epoch on P40). NRTR takes merely 5.0 hours per epoch and therefore is at least 8 times faster than the existing best recognizer. The inference speed of NRTR is approximately 0.03s per image, compared to 0.11s in [1] and 0.2s in [6].

3) Visualization: We show both correct and incorrect examples of NRTR in Fig.1. As can be seen, NRTR could recognize extremely challenging scene images, e.g., low resolution, complex geometric deformations and cluttered background. Some are even hard to human. We carefully analyze incorrect results and split them into three types according to caused reasons. First, texts are severely occluded by other objects, e.g., tree or barrier in example of 'redwood'. Second, characters that look similar are mixed, like 'i' in image of 'valerie' and its fault result 'l'. Third, text orientation are seriously curved, e.g, nearly ninety degrees to the horizontal plane. These failed examples also highlight future research directions of the proposed NRTR.

V. Conclusions

This paper points out two problems lying in current RNN/CNN-based scene text recognizers and proposes a no-recurrence model aiming at increasing computation parallelization and performance. Experiments demonstrate its superiority on accuracy and training speed. We intend to extend the idea to end-to-end text spotting system.

Acknowledgment

This work is supported by the Strategic Priority Research Program of the Chinese Academy of Sciences (XDB-S01070101).

References

[1] F. Bai, Z. Cheng, Y. Niu, S. Pu, and S. Zhou, "Edit probability for scene text recognition," *arXiv preprint:1805.03384*, 2018.

[2] Z. Cheng, F. Bai, Y. Xu, G. Zheng, S. Pu, and S. Zhou, "Focusing attention: Towards accurate text recognition in natural images," in *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2017, pp. 5086–5094.

[3] B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *IEEE transactions on pattern analysis and machine intelligence*.

[4] Y. Gao, Y. Chen, J. Wang, and H. Lu, "Reading scene text with attention convolutional sequence modeling," *arXiv preprint arXiv:1709.04303*, 2017.

[5] F. Yin, Y.-C. Wu, X.-Y. Zhang, and C.-L. Liu, "Scene text recognition with sliding convolutional character models," *arXiv preprint arXiv:1709.01727*, 2017.

[6] B. Shi, X. Wang, P. Lyu, C. Yao, and X. Bai, "Robust scene text recognition with automatic rectification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4168–4176.

[7] P. He, W. Huang, Y. Qiao, C. C. Loy, and X. Tang, "Reading scene text in deep convolutional sequences." in *AAAI*.

[8] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE transactions on neural networks*.

[9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, pp. 6000–6010.

[10] K. Wang, B. Babenko, and S. Belongie, "End-to-end scene text recognition," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1457–1464.

[11] C.-Y. Lee and S. Osindero, "Recursive recurrent nets with attention modeling for ocr in the wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2231–2239.

[12] A. Mishra, K. Alahari, and C. Jawahar, "Scene text recognition using higher order language priors," in *BMVC 2012-23rd British Machine Vision Conference*. BMVA, 2012.

[13] S. M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, R. Young, K. Ashida, H. Nagai, M. Okamoto, H. Yamamoto *et al.*, "Icdar 2003 robust reading competitions: entries, results, and future directions," *International Journal of Document Analysis and Recognition (IJDAR)*.

[14] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L. G. i Bigorda, S. R. Mestre, J. Mas, D. F. Mota, J. A. Almazan, and L. P. De Las Heras, "Icdar 2013 robust reading competition," in *ICDAR, 2013 12th International Conference*. IEEE, 2013, pp. 1484–1493.

[15] D. Karatzas, L. Gomez-Bigorda *et al.*, "Icdar 2015 competition on robust reading," in *ICDAR, 2015 13th International Conference*.

[16] T. Quy Phan, P. Shivakumara, S. Tian, and C. Lim Tan, "Recognizing text with perspective distortion in natural scenes," in *Proceedings of the IEEE International Conference on Computer Vision*.

[17] A. Risnumawan, P. Shivakumara, C. S. Chan, and C. L. Tan, "A robust arbitrary text detection system for natural scene images," *Expert Systems with Applications*.

[18] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, "Synthetic data and artificial neural networks for natural scene text recognition," *arXiv preprint arXiv:1406.2227*, 2014.

[19] Y. Zhang, W. Chan, and N. Jaitly, "Very deep convolutional networks for end-to-end speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 4845–4849.

[20] A. Gupta, A. Vedaldi, and A. Zisserman, "Synthetic data for text localisation in natural images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2315–2324.

[21] V. Goel, A. Mishra, K. Alahari, and C. Jawahar, "Whole is greater than sum of parts: Recognizing scene text words," in *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*. IEEE, 2013, pp. 398–402.

[22] A. Bissacco, M. Cummins, Y. Netzer, and H. Neven, "Photoocr: Reading text in uncontrolled conditions," in *Computer Vision (ICCV), 2013 IEEE International Conference on*. IEEE, 2013, pp. 785–792.

[23] O. Alsharif and J. Pineau, "End-to-end text recognition with hybrid hmm maxout models," *arXiv preprint arXiv:1310.1811*, 2013.

[24] J. Almazán, A. Gordo, A. Fornés, and E. Valveny, "Word spotting and recognition with embedded attributes," *IEEE transactions on pattern analysis and machine intelligence*.

[25] C. Yao, X. Bai, B. Shi, and W. Liu, "Strokelets: A learned multi-scale representation for scene text recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 4042–4049.

[26] M. Jaderberg, A. Vedaldi, and A. Zisserman, "Deep features for text spotting," in *European conference on computer vision*.

[27] B. Su and S. Lu, "Accurate scene text recognition based on recurrent neural network," in *Asian Conference on Computer Vision*. Springer, 2014, pp. 35–48.

[28] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, "Reading text in the wild with convolutional neural networks," *International Journal of Computer Vision*.

[29] S. K. Ghosh, E. Valveny, and A. D. Bagdanov, "Visual attention models for scene text recognition," *arXiv preprint arXiv:1706.01487*, 2017.

[30] Z. Cheng, Y. Xu, F. Bai, Y. Niu, S. Pu, and S. Zhou, "Aon: Towards arbitrarily-oriented text recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5571–5579.

[31] B. Shi, M. Yang, X. Wang, P. Lyu, C. Yao, and X. Bai, "Aster: An attentional scene text recognizer with flexible rectification," *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[32] M. Liao, J. Zhang, Z. Wan, F. Xie, J. Liang, P. Lyu, C. Yao, and X. Bai, "Scene text recognition from two-dimensional perspective," *arXiv preprint arXiv:1809.06508*, 2018.

[33] H. Li, P. Wang, C. Shen, and G. Zhang, "Show, attend and read: A simple and strong baseline for irregular text recognition," *arXiv preprint arXiv:1811.00751*, 2018.