A Simple and Strong Convolutional-Attention Network for Scene Text Recognition

Lu Yang[a], Peng Wang[a,*], Hui Li[b], Ye Gao[a], Linjiang Zhang[a], Chunhua Shen[b], Yanning Zhang[a]

[a]*School of Computer Science, Northwestern Polytechnical University, Xi'an, China*
[b]*School of Computer Science, The University of Adelaide, Australia*

Abstract

Reading irregular scene text of arbitrary shape in natural images is still a challenging problem, despite the progress made recently. Many existing approaches incorporate sophisticated network structures to handle various shapes, use extra annotations for stronger supervision, or employ hard-to-train recurrent neural networks for sequence modeling. In this work, we propose a simple yet robust approach for scene text recognition. With no need to convert input images to sequence representations, we directly connect two-dimensional CNN features to an attention-based sequence decoder. As no recurrent module is adopted, our model can be trained in parallel. It achieves $1.7\times$ to $10\times$ acceleration to backward pass and $1.4\times$ to $9\times$ acceleration to forward pass, compared with the RNN counterparts. The proposed model is trained with only word-level annotations. With this simple design, our method achieves state-of-the-art or competitive recognition performance on the evaluated regular and irregular scene text benchmark datasets.

Keywords: Convolutional-Attention, Transformer, Scene Text Recognition

*Corresponding author
Email addresses: lu.yang@mail.nwpu.edu.cn (Lu Yang), peng.wang@nwpu.edu.cn (Peng Wang), hui.li02@adelaide.edu.au (Hui Li), gaoye@mail.nwpu.edu.cn (Ye Gao), zhanglinjiang@mail.nwpu.edu.cn (Linjiang Zhang), chunhua.shen@adelaide.edu.au (Chunhua Shen), ynzhang@nwpu.edu.cn (Yanning Zhang)

1. Introduction

Text in natural scene images contains rich semantic information that is crucial for visual understanding and reasoning in many cases. Text reading has been integrated in a variety of vision tasks, such as fine-grained image classification [1, 2, 3], image retrieval [1, 4] and visual question answering [5, 6].

Recognizing regular text in almost straight lines can be considered as a sequence-to-sequence problem and solved by an attentional Recurrent Neural Network (RNN) framework as shown in Figure 1(a). In comparison to regular text recognition, it is much more challenging to recognize irregular text of arbitrary shape for a machine. Existing approaches for irregular text recognition can be roughly categorized into four types, namely, shape rectification, multi-direction encoding, character detection and 2D attention based approaches, as shown in Figure 1(b), (c), (d), (e) respectively. The shape rectification based methods [7] first approximately rectify irregular text into regular one, and then apply regular text recognizers. Nevertheless, severely distorted or curved shapes are difficult to be rectified. Cheng *et al*. [8] propose a sophisticated four-directional encoding method to recognize arbitrarily-oriented text, which, however, introduces redundant representations. Character detection based methods [9] firstly detect and recognize individual characters and then connect them using a separate post-processing method, which inherently requires character-level annotations and cannot be trained end-to-end. 2D attention based approaches learn to focus on individual character features in 2D spaces during decoding, which can be trained either with word-level [10] or character-level annotations [11].

Note that a large number of irregular text recognizers (*e.g*., [7, 12, 8, 10, 13]) still need to convert input images into intermediate sequence representations, and use RNNs to encode and decode them. There are two limitations for this type of approaches. First, given that irregular text actually being distributed in two dimensional spaces, to some extent, it is inappropriate and difficult to convert them into one dimensional sequence representations. As shown in [9], solving the irregular text recognition problem from two dimensional perspective may yield more robust performance. Second, RNNs are inherently difficult to be parallelized and typically hard to train due to the problem of

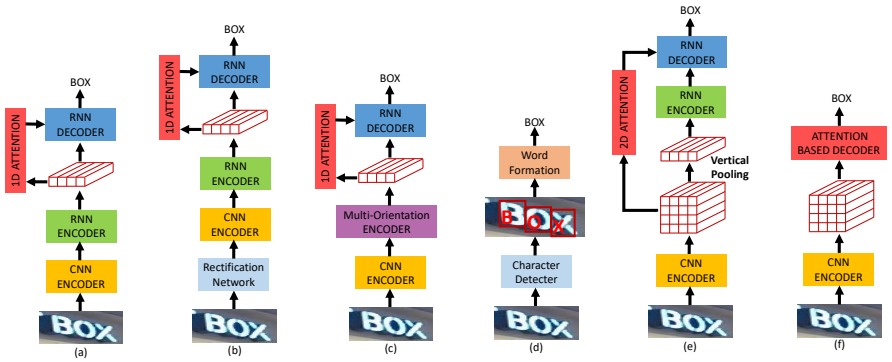

Figure 1: Typical architectures and our model for scene text recognition. (a) is the basic 1D attention based encoder-decoder framework for regular text recognizer [16]. (b)-(f) are all for irregular text recognition. (b) Shape rectification based [7]; (c) Multi-direction encoding based [8]; (d) Character detection based [9]; (e) 2D attention based [10]; (f) our model (end-to-end trainable convolutional attention based, without RNNs being used.)

gradient vanishing/exploding. In the field of regular text recognition, some attempts have been made to replace RNNs with non-recurrent architectures, including convolution based [14] and attention based sequence modeling [15] methods. However, both methods are still based on sequence-to-sequence structures, which is not well capable of handling irregular text of arbitrary shape.

To this end, we propose a simple yet robust architecture for irregular text recognition, as shown in Figure 1(f). Our approach directly connects a CNN-based 2D image encoder to an attention-based 1D sequence decoder, preventing from using intermediate sequence representations. Inspired by the Transformer [17] in NLP, we adopt an attention-based decoder that does not rely on recurrent connections and so can be trained in parallel and converges quickly.

Note that the Transformer is proposed for machine translation, taking 1D sequences as inputs. But the inputs of our proposed irregular text recognizer are 2D images, which makes these two models different from each other. The self-attention mechanism, which plays a key role in the Transformer to model long-range dependencies in both input and output sequences, is relatively less important in our model for text recognition.

Firstly, instead of using self-attention, we use a CNN to encode input scene text images. Accordingly, we need to use 2D attention in the decoder. Secondly, the employment of self-attention in the decoder offers no significant performance gain. This is not surprising: the dependency between characters of a single word is typically weaker than that between words of a sentence or paragraph.

Our main contributions are three-fold:

1) The proposed model is simple by design. It only consists of a CNN model for image encoding and a tailored attention-based sequence decoder. Unlike sequence-to-sequence text recognizers, we do not convert input images to sequence representations, which itself is challenging for text of complex shape. Instead, we convert the input image to a 2D feature map and a 1D global representation by a CNN model, and then connect them directly to the sequence decoder. Furthermore, the training of the proposed model only requires word-level annotations, which enables it to be trained with real data that usually does not come with character-level annotations.

2) Our proposed method is an end-to-end trainable non-recurrent network for both regular and irregular text recognition. Without using any RNN module, this model can be trained in parallel. Compared with state-of-the-art RNN-based irregular text recognizers [7, 10], our model is $1.7\times$ to $10\times$ faster in backward pass and $1.4\times$ to $9\times$ faster in forward pass. This acceleration leads to a rapid experimental turnaround and makes our model scalable to larger datasets.

3) We conduct comprehensive experiments on a variety of public benchmarks, and the results show that our method achieves state-of-the-art or competing performance on both regular and irregular datasets.

Notation. Matrices and column vectors are denoted by bold upper and lower case letters respectively. $\mathbb{R}^m$ and $\mathbb{R}^{m\times n}$ indicate real-valued m dimensional vectors and $m \times n$ matrices respectively. $\langle \mathbf{a}, \mathbf{b} \rangle \in \mathbb{R}$ means the inner-product of $\mathbf{a} \in \mathbb{R}^m$ and $\mathbf{b} \in \mathbb{R}^m$. $[\mathbf{a}, \mathbf{b}] \in \mathbb{R}^{m\times 2}$ and $[\mathbf{a}; \mathbf{b}] \in \mathbb{R}^{2m}$ represent the horizontal and vertical stacks of $\mathbf{a}$ and $\mathbf{b}$ respectively.

2. Related Work

Irregular Scene Text Recognition. Early work for scene text recognition adopts a bottom-up fashion [18, 19], which detects individual characters firstly and integrates them into a word by means of dynamic programming, or a top-down manner [20], which treats the word patch as a whole and recognizes it as multi-class image classification. Considering that scene text generally appears in the form of a character sequence, recent work models it as a sequence recognition problem. RNNs are generally used for sequential feature learning. Connectionist Temporal Classification (CTC) and sequence-to-sequence learning models are two prevalent methods that are widely used for scene text recognition [21, 16, 12, 22, 23].

Methods for irregular text recognition are mostly driven by the above frameworks but involve some improvements to deal with the distortions or curvatures of irregular text. For instance, Shi *et al*. [13, 7] proposed to rectify irregular text images into regular ones by Spatial Transformer Network (STW) [24], and then recognized them using a 1D attentional sequence-to-sequence model. Zhan and Lu [25] proposed to iteratively remove perspective distortion and text line curvature by an innovative rectification network so as to result in a fronto-parallel view of text for recognition. Rather than rectifying the entire word image, Liu *et al*. [26] proposed to detect and rectify individual characters in the word by STW. Cheng [8] captured the deep features of irregular text image along four directions by RNNs, which are then combined by 1D attention based decoder to generate character sequence. A filter gate was designed to fuse those redundant features and remove irrelevant ones. Liao *et al*. [9] argued that it is inappropriate to represent irregular text image with a 1D sequence, and proposed a Character Attention Fully Convolutional Network to detect each character accurately in two-dimensional perspective. Word formation is then realized with a separate segmentation based method. This model cannot be trained end-to-end. Some methods attempt to extend 1D attention mechanism into 2D spaces. Character-level annotations are often needed to supervise the training of 2D attention network. For example, the Focusing Attention Network (FAN) proposed by Cheng *et al*. [12] introduced a focus network to tackle the attention drift between the local character feature and target. Yang *et al*. [11] introduced an auxiliary Fully Convolutional Network for dense character detection. An alignment loss was used to supervise the training of attention model

during word decoding. Li *et al*. [10] modified the attention model and proposed a tailored 2D attention based framework for exact local feature extraction. Nevertheless, 2-layer RNNs are adopted respectively in both encoder and decoder which precludes computation parallelization and suffers from heavy computational burden.

Non-recurrent Sequence Modeling. Some work has been proposed in recent years to remove the recurrent structure in the sequence-to-sequence learning framework, so as to enable fully parallel computation and accelerate the processing speed. Gehring *et al*. [27] proposed an architecture for machine translation with entirely convolutional layers. Compositional structures in the sequence can be discovered based on the hierarchical representations. However, this model still has difficulty to learn dependencies between distant positions. Vaswani *et al*. [17] proposed a "Transformer" for machine translation, which is based solely on attention mechanisms. The fundamental self-attention module can draw dependencies between different positions in a sequence through position-pair computation rather than position-chain computed by RNNs, which leads to more computation parallelization and less model complexity. Inspired by this model, Dong *et al*. [28] introduced Transformer to speech recognition and Yu *et al*. [29] combined local convolution with global self-attention for reading comprehension task. Most recently, Dehghani *et al*. [30] generalized the Transformer and proposed the "Universal Transformer" to deal with string copying or logical inference with string's length exceeding those observed at training time. There are also some efforts for scene text reading without using recurrent networks. Gao *et al*. [14] presented an end-to-end attention convolutional network for scene text recognition, with a CTC layer followed to generate the final label. Wu [31] presented a sliding convolutional attention network for scene text recognition, based on the convolutional sequence-to-sequence learning framework [27]. Sheng *et al*. [15] proposed a non-recurrent sequence-to-sequence model for scene text recognition based on Transformer [17], with self-attention module working as the basic block in both encoder and decoder to learn character dependencies. All these sequence-to-sequence frameworks are mainly for regular text recognition and are not easy to be extended to handle irregular text because of their inherent model design. In contrast, in this work, we propose an simple yet effective 2D image to

1D sequence model based on convolution and attention modules. It maps text images into character sequences directly and can address both regular and irregular scene text recognition.

3. Model Architecture

As shown in Figure 2, the proposed model is based on an encoder-decoder structure, which is popular for many cross-modality transformation tasks. Previous sequence-to-sequence based text recognizers represent input images with 1D sequences, which, however, encounter difficulties when dealing with irregular text scattering in 2D spaces. Alternatively, we employ a CNN encoder to extract both 2D feature map (two dimensional representations) and global representation (one dimensional representation) of text images. The resulting image representations are then fed into an attention-based decoder with a stack of masked self-attention, 2D attention and point-wise feed-forward layers.

During testing, the decoder takes as input at each step the concatenation of the global representation and the embedding of the previously generated character which is added with the encoding of the current position, adaptively focuses on the related image regions via 2D attention, and predict the character at the current position. During training, given ground-truth labels, the computation of the decoder can be easily parallelized. In the following, we introduce each component of our proposed model in detail.

3.1. Encoder

Without bells and whistles, we adopt as our CNN encoder the ResNet34 [32] based architecture, which consists of a ResNet34 and a global representation extractor as shown in Figure 2. The final average pooling layer of the original ResNet34 is removed and then followed two branches. One branch is a 1×1 convolution layer to transform the dimension of the 2D feature map from 512 to 1024 and feed into 2D attention, the other branch is a global representation extractor which consists of B bottlenecks, average pooling and a fully connected layer. The resulting global representation constitute the input of the decoder. The ablation study shows that $B = 6$ is enough in

our case (see Section 4.3 for details). All the input images are uniformly resized into $128 \times 400 \times 3$, resulting in feature maps of size $4 \times 13 \times 512$. Empirically, we find that the larger the input image sizes are, the better is the recognition performance (see Section 4.3 for details). We also evaluate other CNN backbones such as ResNet50 and ResNet152 for image encoding, which do not offer significant performance improvements, as referred to the ablation experiments. Note that it may be more reasonable to rescale images without destroying their original aspect ratios [10], which we leave for future work.

3.2. Decoder

Inspired by [17], the designed attention-based sequence decoder is composed of three layers: 1) a masked self-attention mechanism for modeling dependencies between different characters within output words; 2) a 2D attention module linking encoder and decoder; and 3) a point-wise feed-forward layer applied to each decoding position separately. A residual connection with an addition operation is employed for each of the above three layers, followed by layer normalization. The above three components form a block and can be stacked N times without sharing parameters. There are $N = 6$ blocks in the Transformer [17], but we found that using only one block already achieves saturated performance in our case (see Section 4.3). In the following, we describe the decoder components in detail.

Output Embedding and Positional Encoding. During testing, the previously generated character will be embedded to a $d/2$-dimensional vector at each decoding step, which is further added with the encoding of the current position as follows:

$$\mathrm{PE}(p, i) = \begin{cases} \sin(p/10000^{i/(d/2)}) & \text{if } i \text{ is even} \\ \cos(p/10000^{(i-1)/(d/2)}) & \text{if } i \text{ is odd} \end{cases} \tag{1}$$

where p is the position and $i \in \{1, \ldots, (d/2)\}$ is the dimension. Then they are concatenated to global representation. While at training time, the ground-truth characters are shifted right and embedded simultaneously, which enables parallel training.

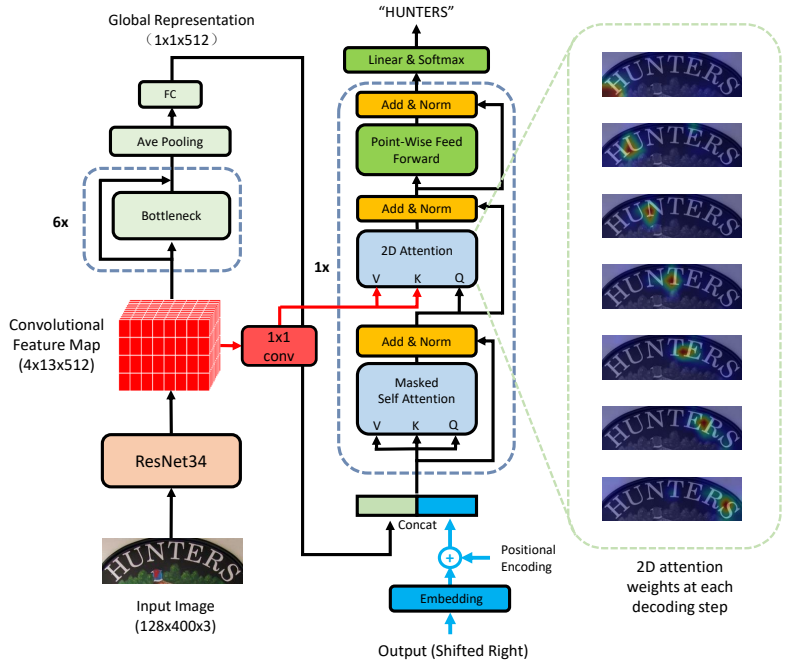

Figure 2: The overall structure of our proposed model. It consists of two parts: a ResNet34-based image encoder (left) and an attention-based sequence decoder (right). The 2D feature maps generated by ResNet34 are connected to the 2D attention module in the decoder by a 1×1 convolution layer, and we stack 6 bottleneck modules to extract the 1D global representation which can help the 2D attention more accurate. The bottleneck module is the same as in ResNet [32]. In contrast to other irregular text recognizers [33, 7], there is no recurrent networks to model the representation. As a non-recurrent network, our model can be trained in parallel. Furthermore, training our model only needs word-level annotations.

Multi-Head Dot-Product Attention. Both masked self-attention and two-dimensional attention in our decoder are based on the multi-head dot-product attention formulation [17]. Here, we briefly review this formulation. The scaled dot-product attention takes as inputs a query $\mathbf{q} \in \mathbb{R}^d$ and a set of key-value pairs of d-dimensional vectors $\{(\mathbf{k}_i, \mathbf{v}_i)\}_{i=1,2,\ldots,M}$ (M is the number of key-value pairs), and computes as output a weighted sum of the values, where the weight for each value is computed by a scaled dot-product of the query and the corresponding key. The formulation of scaled dot-product attention can be expressed as follows:

$$\begin{aligned} &\mathrm{Atten}(\mathbf{q}, \mathbf{K}, \mathbf{V}) = \sum_{i=1}^{M} \alpha_i \mathbf{v}_i \in \mathbb{R}^d \\ &\text{where } \boldsymbol{\alpha} = \mathrm{softmax}\big(\frac{\langle \mathbf{q}, \mathbf{k}_1 \rangle}{\sqrt{d}}, \frac{\langle \mathbf{q}, \mathbf{k}_2 \rangle}{\sqrt{d}}, \ldots, \frac{\langle \mathbf{q}, \mathbf{k}_M \rangle}{\sqrt{d}}\big) \end{aligned} \tag{2}$$

is the attention weights, $\mathbf{K} = [\mathbf{k}_1, \mathbf{k}_2, \ldots, \mathbf{k}_M]$ and $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_M]$. If there is a set of queries $\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_{M'}]$ (M' is the number of queries), then we have:

$$\begin{aligned} &\mathrm{Atten}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_{M'}] \in \mathbb{R}^{d \times M'} \\ &\text{where } \mathbf{a}_i = \mathrm{Atten}(\mathbf{q}_\mathrm{i}, \mathbf{K}, \mathbf{V}). \end{aligned} \tag{3}$$

The above scaled dot-product attention can be applied multiple times (multi-head) with different linear projections to $\mathbf{Q}$, $\mathbf{K}$ and $\mathbf{V}$, followed by a concatenation and projection operation:

$$\begin{aligned} &\mathrm{MHAtten}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathbf{W}^o[\mathbf{A}_1; \ldots; \mathbf{A}_H] \in \mathbb{R}^{d \times M'} \\ &\text{where } \mathbf{A}_i = \mathrm{Atten}(\mathbf{W}_\mathrm{i}^\mathrm{q}\mathbf{Q}, \mathbf{W}_\mathrm{i}^\mathrm{k}\mathbf{K}, \mathbf{W}_\mathrm{i}^\mathrm{v}\mathbf{V}). \end{aligned} \tag{4}$$

The parameters are $\mathbf{W}_i^q \in \mathbb{R}^{\frac{d}{H} \times d}$, $\mathbf{W}_i^k \in \mathbb{R}^{\frac{d}{H} \times d}$, $\mathbf{W}_i^v \in \mathbb{R}^{\frac{d}{H} \times d}$ and $\mathbf{W}^o \in \mathbb{R}^{d \times d}$. We set the number of attention heads H to 16 for our proposed model (see Section 4.3 for the ablation study on the selection of H).

Masked Self-Attention. This attention layer is used to model the dependencies between different decoding positions, where the queries, keys and values are the same, *i.e.*, the right-shifted outputs. In this case, $M = M' =$ the length of decoded sequence.

A mask is applied to prevent each position from attending to positions after that position.

Two-Dimensional Attention. In this layer, the queries come from the masked self-attention layer, and the keys and values are the 2D output features of the CNN encoder. In this case, $M = 4 \times 13$ and M' is the length of decoded sequence. It is the only connection between the encoder and decoder, and allows each decoding position attend to the 2D positions of the encoder outputs.

Point-wise Feed-Forward Layer. A simple feed-forward network is applied at each position of the outputs of two-dimensional attention layer, which contains two linear transformations of dimension d' and a ReLU non-linearity in between. The parameters of this layer are shared across all positions.

Prediction and Loss Function. A linear transformation followed by a softmax function is used to transform the decoder output into prediction probabilities over character classes. Here we use 94 character classes, including digits, case-sensitive letters and 32 punctuation characters. The parameters are also shared over all decoding positions. The standard cross-entropy function is adopted to compute the loss of the predicted probabilities w.r.t. the ground-truth, at each decoding position.

4. Experiments

We evaluate the performance of our method on a number of scene text recognition benchmarks including both regular and irregular text. Ablation study is also conducted to investigate the impact of different model hyper-parameters.

4.1. Datasets

Our model is solely trained on synthetic datasets without using any real-world images. The same trained model, without further fine-tuning, is then evaluated on the following standard datasets: IIIT 5K-Words (IIIT5K) [34], Street View Text (SVT) [18], ICDAR2013 (IC13) [35], ICDAR2015 (IC15) [36], Street View Text Perspective (SVTP) [19] and CUTE80 (CT80) [37].

Synthetic Datasets Three public synthetic datasets are employed to train our model: Synth90K the 9-million-word synthetic data released by [20], SynthText the 8-million-word data proposed by [38] and SynAdd the 1.6-million-word synthetic data released by [10].

IIIT5K [34] is collected from Internet. It has 3000 cropped word images for test, with nearly horizontal text instances.

SVT [18] contains 647 cropped text images for test. It is collected from Google Street View. Although the text instances are mostly horizontal, many images are severely corrupted by noise and blur, or have very low resolutions.

IC13 [35] has 1095 regular word patches for test. For fair comparison, we remove images that contain non-alphanumeric characters, which results in 1015 images.

IC15 [36] consists of images captured incidentally by Google Glasses, and so has many irregular word patches (perspective or oriented). It includes 2077 images for test. To fairly compare with previous methods [8, 33, 25, 12, 23, 7], we also used two simplified versions of the IC15 dataset called IC15-Char&Digit and IC15-1811. IC15-Char&Digit also includes 2077 images, but discards non-alphanumeric characters in the annotations. IC15-1811 discards the images which have non-alphanumeric characters, and contains 1811 images.

SVTP [19] contains 645 cropped images for test. Images are selected from side-view angle snapshots in Google Street View, which are mostly perspective distorted.

CT80 [37] consists of 288 cropped high resolution images for test. It is specically collected for evaluating the performance of curved text recognition.

4.2. Implementation Details

The proposed model is implemented using PyTorch. All experiments are conducted on an NVIDIA GTX 1080Ti GPU with 11GB memory. We use the ADADELTA optimizer [39] to train the model, with a batch size of 128. The model is trained 3 epochs on synthetic datasets. The ResNet34 in CNN encoder is initialized by the ImageNet pre-trained model. The global representation dimension is equal to word embedding dimension, which is 512. So the dimensions d and d' are set to 1024 and 2048 respectively in our experiments.

CNN Backbone	Input Image Size	Accuracy	
		III5K	IC15
ResNet34	32×100	86.9	65.6
ResNet34	64×200	92.9	72.3
ResNet34	128×400	**94.2**	74.8
ResNet50	128×400	93.7	74.0
ResNet152	128×400	93.9	**75.9**

Table 1: Performance with different CNN backbones and input image sizes. Increasing the size of input images significantly improves the performance. ResNet34 achieves a good balance between performance and model size.

A few data augmentation is adopted during test phase. The test image is rotated ± 5 degrees respectively, and fed into our model together with the original image. For images with height twice larger than width, we rotate the image ± 90 degrees. The highest-scored recognition result will be chosen as the final output. Beam search is also applied for the decoder. It keeps the top-k candidates with the highest accumulative scores, where k is empirically set to 5 in our experiments.

4.3. Ablation Study

CNN Backbone Selection. We first experiment with different CNN models for image encoding, including ResNet34, ResNet50 and ResNet152. Experimental results in Table 1 show that ResNet34 achieves a good balence between model size and accuracy. So we choose ResNet34 as our backbone in the following experiments. We also evaluate with different input image sizes, including 32×100, 64×200 and 128×400, and find that larger input size results in higher accuracy.

Number of Decoder Blocks. As shown in Row 4, 6, 7 of Table 2, we set the number of decoder blocks to 1, 2, 4 while keeping the number of attention heads as 16. The results show that best performance of our model is achieved when $N = 1$. This phenomenon is in contrast to the experimental results of the Transformer [17], which shows that using more blocks yield better machine translation performance.

Block Number (N)	Head Number (H)	Accuracy	
		IIIT5K	IC15
1	1	93.5	73.1
1	4	93.8	74.2
1	8	93.8	**75.1**
1	16	**94.2**	74.8
1	32	94.2	74.4
2	16	92.9	73.8
4	16	90.6	70.0

Table 2: The performance with different block numbers and attention head numbers in the decoder. It shows that using more heads can slightly improve the performance but using more blocks (with $H = 16$) degrades the performance.

Number of Attention Heads. Another factor that affects the recognition performance is the number of attention heads H. We evaluate the recognition performance of our models with 1, 4, 8, 16, 32 attention heads, respectively. The experimental results in Table 2 show that the more attention heads we used, the better performance it achieved. In the following, we set the number of attention heads H to 16.

Impact of Global Representation. The global representation vector in our model encodes the rich context information of the entire input image. It is fed into the decoder at each time step, together with the last decoded character. Figure 3 demonstrates a case study of the 2D attention maps generated with and without the global representation. We can see that the algorithm focuses more accurately on the characters to be decoded with the context information provided by the global representation. In addition, we study the effects of the number of bottlenecks that are used to generate the global representation. As shown in Table 3, using more bottleneck modules can improve the performance both for regular and irregular word recognition. When the bottleneck number reachs 6, the performance is almost saturated. So we set the number of bottlenecks B to 6 by default.

Impact of Self-Attention. Self-attention plays a key role in many sequence-to-sequence tasks (*e.g.*, machine translation), due to its ability of modeling long-range dependen-

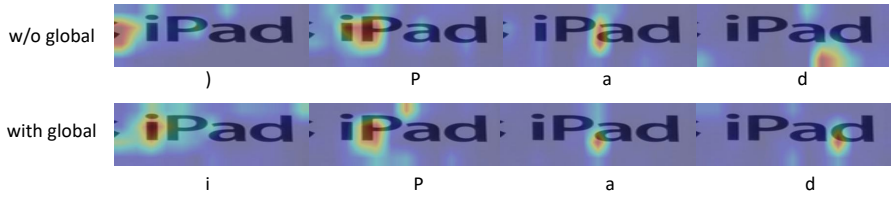

Figure 3: Case study of attention maps with or without global representation. With global representation, our algorithm tends to focus more correctly on the regions of characters to be decoded.

Bottleneck Number(B)	Accuracy	
	IIIT5K	IC15
No Global	93.3	74.0
2	93.4	74.6
4	93.4	**75.1**
6	**94.2**	74.8
8	94.2	75.0

Table 3: The performance with different bottleneck number for global representation. It shows that using deeper bottleneck can improve the performance both for regular and irregular datasets, and when the bottleneck number is 6 , the performance is almost saturated.

cies. In the context of image processing, self-attention share a similar spirit with non-local neural networks [40]. In this section, we examine the impact of self-attention in our proposed model for irregular text recognition. We firstly add a self-attention module on top of the convolutional feature maps, to enhance the representation of dependencies between distant image regions. However, the results in Table 4 show that the addition of self-attention in the encoder does not bring improvement. On the other hand, to examine the impact of self-attention on the decoder side, we remove the self-attention modules from the decoder. The recognition performance of the resulting model just moderately drops compared with the original model (0.8% for IIIT5K containing regular text and 0.6% for IC15 consisting of irregular text), which is still comparable to previous methods.

In contrast to machine translation, we find that the usage of self-attention in our irregular text recognizer has a relatively small impact on the performance. We analyze

Encoder Self-attention	Decoder Self-attention	Accuracy	
		IIIT5K	IC15
×	×	93.4	74.2
×	√	**94.2**	**74.8**
√	√	93.9	74.8

Table 4: The performance with or without self-attention in the encoder and decoder. Comparing Rows 1 and 2, removing self-attention in the decoder from our model results in a moderate performance drop. From Rows 2 and 3, we can see that adding self-attention in the encoder does not show significant improvement.

that the reasons may be three-fold. First, the lengths of sequences to be modeled in the task of irregular text recognition is typically smaller than that in machine translation. For example, in the Multi30K [41] dataset for English-German translation, the average lengths of input and output sequences are 11.8 and 11.1 respectively. While the average length of output sequences is 5.3, in the test set of IC15 [36] for irregular text recognition. Apparently, it is less important to model long-range dependencies for short sequences. Second, the deep CNN encoder already models a certain level of long-range dependencies, given that the receptive field of the final feature layer of ResNet34 is 889 that is larger than the input image size (128×400). Last, in machine translation, self-attention is typically used to model the dependencies between words in a sentence or even a paragraph. There are still rich semantic and syntactic relationships between words that are far from each other. While for irregular text recognition, each input image usually contains a particular word, and the self attention is only used to model character dependencies in a word. The dependencies between characters of a word are typically weaker than that between words in a sentence or paragraph. That may be why self-attention does not empirically improve a lot to the performance of irregular text recognition.

4.4. Comparison with State-of-the-art

In this section we evaluate our model with $N = 1$, $H = 16$ and $d = 1024$, in comparison with state-of-the-art approaches on several benchmarks. For fair comparison, we only demonstrate the performance of the SAR [10] model trained with the same synthetic data. As shown in Table 6, our proposed method outperforms other

Method	Model Size	Forward Time per Batch	Backward Time per Batch
Shi *et al*. 2018 [7]	22M	65ms	143ms
Li *et al*. 2019 [10]	61M	404ms	903ms
Ours	59M	45ms	85ms

Table 5: The comparison on training speed and model size. The speed is evaluated with 20-sized batches in average. Our model is $1.7\times$ to $10.6\times$ faster in backward pass and $1.4\times$ to $9.0\times$ faster in forward pass.

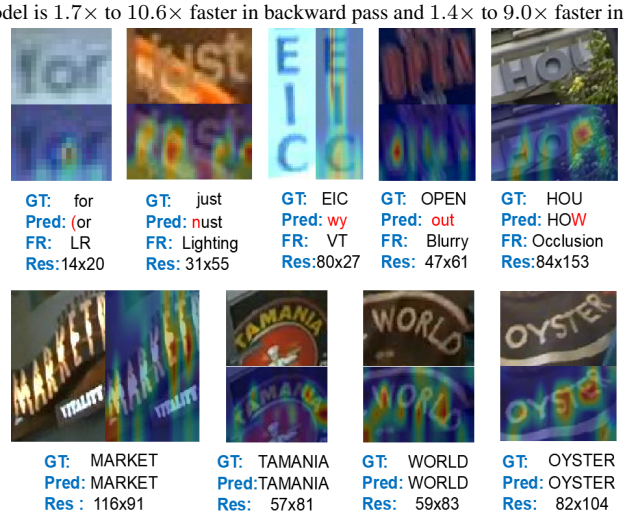

Figure 4: Some success and failure cases by our approach. The 2D attention weights combining all decoding steps are also illustrated. "GT": Ground Truth, "Pred": Prediction, "FR": Failure Reason, "Res": Original Image Resolution. The reasons for failure include blurry, low resolution (LR), lighting, vertical text (VT), and occlusion *etc* .

approaches on **all of** evaluated settings for irregular text recognition. In particular, it achieves accuracy increases of 3.0% (from 76.1% to 79.1%) on IC15-1811 and 2.1% (from 79.6% to 81.7%) on SVTP-None.

And for regular text datasets, our performance is also competitive. On the IIIT5K dataset which contains the largest number of test images over the three evaluated regular datasets, our model is 0.8% better than the best model (94.2% v.s. 93.4%).

We also compare the model size and computation speed of our model with a simple yet strong baseline [10] and a state-of-the-art model [7]. The experiment is performed on a 1080ti GPU with a batch size of 20. Due to the non-recurrence property, our model

Method	Regular Text			Irregular Text						
	IIIT5K	SVT	IC13	IC15			SVTP			CT80
	None	None	None	None	Char&Digit	1811	50	Full	None	None
Wang *et al*. 2011 [18]	-	-	-	-	-	-	40.5	21.6	-	-
Mishra *et al*. 2012 [34]	-	-	-	-	-	-	45.7	24.7	-	-
Phan *et al*. 2013 [19]	-	-	-	-	-	-	75.6	67.0	-	-
Jaderberg *et al*. 2015 [24]	-	80.7	90.8	-	-	-	-	-	-	42.7
Lee and Osindero 2016 [16]	78.4	80.7	90.0	-	-	-	-	-	-	-
Wang and Hu 2017 [42]	80.8	81.5	-	-	-	-	-	-	-	-
Shi *et al*. 2016 [13]	81.9	81.9	88.6	-	-	-	91.2	77.4	71.8	59.2
Liu *et al*. 2016 [43]	83.3	83.6	89.1	-	-	-	94.3	83.6	73.5	-
Shi *et al*. 2017 [21]	81.2	82.7	89.6	-	-	-	92.6	72.6	66.8	54.9
Yang *et al*. 2017 [11]*	-	-	-	-	-	-	93.0	80.2	75.8	69.3
Cheng *et al*. 2017 [12]*	87.4	85.9	*93.3*	-	-	70.6	-	-	71.5	63.9
Liu *et al*. 2018 [44]*	87.0	-	92.9	-	-	-	92.6	81.6	-	-
Liu *et al*. 2018 [26]*	92.0	85.5	91.1	*74.2*	-	-	-	-	78.9	-
Bai *et al*. 2018 [23]	88.3	87.5	**94.4**	-	-	73.9	-	-	-	-
Cheng *et al*. 2018 [8]	87.0	82.8	-	-	68.2	-	-	-	-	76.8
Shi *et al*. 2018 [7]	*93.4*	*89.5*	91.8	-	-	*76.1*	94.0	83.7	78.5	79.5
Gao *et al*. 2019 [14]	81.8	82.7	88.0	-	-	-	-	-	-	-
Liao *et al*. 2019 [9]*	91.9	86.4	91.5	-	-	-	-	-	-	79.9
Li *et al*. 2019 [10]	91.5	84.5	91.0	69.2	-	-	-	-	76.4	*83.3*
Luo *et al*. 2019 [33]	91.2	88.3	92.4	-	68.8	-	*94.3*	*86.7*	76.1	77.4
Zhan *et al*. 2019 [25]	93.3	**90.2**	91.3	-	*76.9*	-	-	-	*79.6*	*83.3*
Ours	**94.2**	89.0	92.0	**74.8**	**77.1**	**79.1**	**95.7**	**90.1**	**81.7**	**83.7**

Table 6: Scene text recognition performance on public datasets. "Char&Digit" means discard non-alphanumeric characters in the prediction and annotation, "1811" means discard the images which have any non-alphanumeric characters, and there are 1811 images left. "50" and "Full" are lexicon sizes, "None" means no lexicon. For datasets with lexicons, we select from lexicon the one with the minimum edit distance to the predicted word. "*" indicates models trained with both word-level and character-level annotations. **Bold** and *Italic* fonts represent the best and second best performance respectively.

is significantly faster than these two RNN-based models.

Some success and failure cases are also presented in Figure 4. It shows that our model is capable of dealing with text of complex shapes. There are several reasons for our method to make wrong decisions, including blurry images, low resolution, vertical text, lighting and occlusion.

5. Conclusion

In this work, we propose a simple and robust model for scene text recognition. The simplicity of our model is reflected in three aspects. 1) Simple architecture: the proposed model directly connects a CNN encoder to an attention-based encoder. We do not convert input images into sequences as in many existing irregular text recognizers. 2) Parallel training: as a non-recurrent network, our model can be trained in parallel. Compared with two state-of-the-art RNN-based irregular text recognizers, the computational speed of our model is significantly faster. 3) Simple training data: our model only relies on the word-level annotations. As a simple meta-algorithm, this model can be extended in multiple ways, such as incorporating multi-scale image features via stacked 2D attention and resizing input images while keeping aspect ratios. We leave them for further work.

References

[1] S. Karaoglu, R. Tao, T. Gevers, A. W. Smeulders, Words matter: Scene text for image classification and retrieval 19 (5) (2017) 1063–1076.

[2] S. Karaoglu, R. Tao, J. C. van Gemert, T. Gevers, Con-text: Text detection for fine-grained object classification, IEEE transactions on image processing 26 (8) (2017) 3965–3980.

[3] X. Bai, M. Yang, P. Lyu, Y. Xu, J. Luo, Integrating scene text and visual appearance for fine-grained image classification, IEEE Access 6 (2018) 66322–66335.

[4] L. Gómez, A. Mafla, M. Rusinol, D. Karatzas, Single shot scene text retrieval, in: Proc. Eur. Conf. Comp. Vis., 2018.

[5] D. Gurari, Q. Li, A. J. Stangl, A. Guo, C. Lin, K. Grauman, J. Luo, J. P. Bigham, Vizwiz grand challenge: Answering visual questions from blind people, in: Proc. IEEE Conf. Comp. Vis. Patt. Recogn., 2018.

[6] ICDAR 2019 robust reading challenge on scene text visual question answering, `http://rrc.cvc.uab.es/?ch=11`, accessed: 2019-03-09.

[7] B. Shi, M. Yang, X. Wang, P. Lyu, C. Yao, X. Bai, ASTER: An attentional scene text recognizer with flexible rectification, IEEE Trans. Pattern Anal. Mach. Intell. (2018) 1–1.

[8] Z. Cheng, Y. Xu, F. Bai, Y. Niu, S. Pu, S. Zhou, AON: Towards arbitrarily-oriented text recognition, in: Proc. IEEE Conf. Comp. Vis. Patt. Recogn., 2018.

[9] M. Liao, J. Zhang, Z. Wan, F. Xie, J. Liang, P. Lyu, C. Yao, X. Bai, Scene text recognition from two-dimensional perspective, in: Proc. AAAI Conf. Artificial Intell., 2019.

[10] H. Li, P. Wang, C. Shen, G. Zhang, Show, attend and read: A simple and strong baseline for irregular text recognition, in: Proc. AAAI Conf. Artificial Intell., 2019.

[11] X. Yang, D. He, Z. Zhou, D. Kifer, C. L. Giles, Learning to read irregular text with attention mechanisms, in: Proc. Int. Joint Conf. Artificial Intell., 2017.

[12] Z. Cheng, F. Bai, Y. Xu, G. Zheng, S. Pu, S. Zhou, Focusing attention: Towards accurate text recognition in natural images, in: Proc. IEEE Int. Conf. Comp. Vis., 2017, pp. 5086–5094.

[13] B. Shi, X. Wang, P. Lv, C. Yao, X. Bai, Robust scene text recognition with automatic rectification, in: Proc. IEEE Conf. Comp. Vis. Patt. Recogn., 2016.

[14] Y. Gao, Y. Chen, J. Wang, H. Lu, Reading scene text with attention convolutional sequence modeling, Neurocomputing.

[15] F. Sheng, Z. Chen, B. Xu, NRTR: A no-recurrence sequence-to-sequence model for scene text recognition, arXiv:1806.00926.

[16] C.-Y. Lee, S. Osindero, Recursive recurrent nets with attention modeling for ocr in the wild, in: Proc. IEEE Conf. Comp. Vis. Patt. Recogn., 2016.

[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Proc. Adv. Neural Inf. Process. Syst., 2017.

[18] K. Wang, B. Babenko, S. Belongie, End-to-end scene text recognition, in: Proc. IEEE Int. Conf. Comp. Vis., 2011, pp. 1457–1464.

[19] T. Q. Phan, P. Shivakumara, S. Tian, C. L. Tan, Recognizing text with perspective distortion in natural scenes, in: Proc. IEEE Int. Conf. Comp. Vis., 2013, pp. 569–576.

[20] M. Jaderberg, K. Simonyan, A. Vedaldi, A. Zisserman, Reading text in the wild with convolutional neural networks, Int. J. Comp. Vis. 116 (1) (2015) 1–20.

[21] B. Shi, X. Bai, C. Yao, An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition., IEEE Trans. Pattern Anal. Mach. Intell. 39 (11) (2017) 2298–2304.

[22] H. Li, P. Wang, C. Shen, Towards end-to-end text spotting with convolutional recurrent neural networks, in: Proc. IEEE Int. Conf. Comp. Vis., 2017, pp. 5238–5246.

[23] F. Bai, Z. Cheng, Y. Niu, S. Pu, S. Zhou, Edit probability for scene text recognition, in: Proc. IEEE Conf. Comp. Vis. Patt. Recogn., 2018.

[24] M. Jaderberg, K. Simonyan, A. Zisserman, et al., Spatial transformer networks, in: Proc. Adv. Neural Inf. Process. Syst., 2015, pp. 2017–2025.

[25] F. Zhan, S. Lu, ESIR: End-to-end scene text recognition via iterative rectification, in: Proc. IEEE Conf. Comp. Vis. Patt. Recogn., 2019.

[26] W. Liu, C. Chen, K.-Y. K. Wong, Char-Net: A character-aware neural network for distorted scene text recognition, in: Proc. AAAI Conf. Artificial Intell., 2018.

[27] J. Gehring, M. Auli, D. Grangier, D. Yarats, Y. N. Dauphin, Convolutional sequence to sequence learning, in: Proc. Int. Conf. Mach. Learn., 2017.

[28] L. Dong, S. Xu, B. Xu, Speech-transformer: A no-recurrence sequence-to-sequence model for speech recognition, in: Proc. IEEE Int. Conf. Acoustics, Speech & Signal Processing, 2018, pp. 5884–5888.

[29] A. W. Yu, D. Dohan, M.-T. Luong, R. Zhao, K. Chen, M. Norouzi, Q. V. Le, QANet: Combining local convolution with global self-attention for reading comprehension, in: Proc. Int. Conf. Learn. Representations, 2018.

[30] M. Dehghani, S. Gouws, O. Vinyals, J. Uszkoreit, Ł. Kaiser, Universal transformers, in: Proc. Int. Conf. Learn. Representations, 2019.

[31] Y.-C. Wu, F. Yin, X.-Y. Zhang, L. Liu, C.-L. Liu, SCAN: Sliding convolutional attention network for scene text recognition, arXiv:1806.00578.

[32] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proc. IEEE Conf. Comp. Vis. Patt. Recogn., 2016.

[33] L. J. Luo, Canjie, Z. Sun, MORAN: A multi-object rectified attention network for scene text recognition, in: Pattern Recogn., 2019.

[34] A. Mishra, K. Alahari, C. V. Jawahar, Scene text recognition using higher order language priors, in: Proc. British Mach. Vis. Conf., 2012, pp. 1–11.

[35] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L. G. i Bigorda, S. R. Mestre, J. Mas, D. F. Mota, J. A. Almazan, L. P. de las Heras, ICDAR 2013 robust reading competition, in: Proc. Int. Conf. Doc. Anal. Recog., 2013.

[36] D. Karatzas, L. Gomez-Bigorda, A. Nicolaou, S. Ghosh, A. Bagdanov, M. Iwamura, J. Matas, L. Neumann, V. R. Chandrasekhar, S. Lu, F. Shafait, S. Uchida, E. Valveny, ICDAR 2015 robust reading competition, in: Proc. Int. Conf. Doc. Anal. Recog., 2015.

[37] A. Risnumawan, P. Shivakumara, C. S. Chan, C. L. Tan, A robust arbitrary text detection system for natural scene images, Expert Systems with Applications 41 (18) (2014) 8027–8048.

[38] A. Gupta, A. Vedaldi, A. Zisserman, Synthetic data for text localisation in natural images, in: Proc. IEEE Conf. Comp. Vis. Patt. Recogn., 2016, pp. 2315–2324.

[39] M. D. Zeiler, ADADELTA: an adaptive learning rate method, arXiv preprint arXiv:1212.5701.

[40] X. Wang, R. Girshick, A. Gupta, K. He, Non-local neural networks, in: Proc. IEEE Conf. Comp. Vis. Patt. Recogn., 2018.

[41] D. Elliott, S. Frank, K. Sima'an, L. Specia, Multi30k: Multilingual english-german image descriptions (2016) 70–74.

[42] J. Wang, X. Hu, Gated recurrent convolution neural network for ocr, in: Proc. Adv. Neural Inf. Process. Syst., 2017.

[43] W. Liu, C. Chen, K.-Y. K. Wong, Z. Su, J. Han, STAR-Net: A spatial attention residue network for scene text recognition, in: Proc. British Mach. Vis. Conf., 2016.

[44] Z. Liu, Y. Li, F. Ren, W. L. Goh, H. Yu, SqueezedText: A real-time scene text recognition by binary convolutional encoder-decoder network, in: Proc. AAAI Conf. Artificial Intell., 2018.