Show, Attend and Read: A Simple and Strong Baseline for Irregular Text Recognition

Hui Li[*,1] **Peng Wang**[*,2] **Chunhua Shen**[1] **Guyu Zhang**[2]
[1]The University of Adelaide, Australia [2]Northwestern Polytechnical University, China

Abstract

Recognizing irregular text in natural scene images is challenging due to the large variance in text appearance, such as curvature, orientation and distortion. Most existing approaches rely heavily on sophisticated model designs and/or extra fine-grained annotations, which, to some extent, increase the difficulty in algorithm implementation and data collection. In this work, we propose an easy-to-implement strong baseline for irregular scene text recognition, using off-the-shelf neural network components and only word-level annotations. It is composed of a 31-layer ResNet, an LSTM-based encoder-decoder framework and a 2-dimensional attention module. Despite its simplicity, the proposed method is robust and achieves state-of-the-art performance on both regular and irregular scene text recognition benchmarks. The code will be released.

Introduction

Text information in images is of indispensable value in semantic visual understanding. Reading text in natural scene, however, compared to traditional OCR, is still a challenging problem. One of the main reasons is the potential irregularity and diversity of in text shape and layout, which can be curved, oriented or distorted. With the application of deep neural networks, the performance of regular (mostly horizontal) text recognition has been improved rapidly. Taking the ICDAR 2013 benchmark (Karatzas et al. 2013) as example, the best-reported accuracy (Bai et al. 2018) has been 94.4%, to our knowledge. Nonetheless, most regular text recognizers (He et al. 2016b; Shi, Bai, and Yao 2017; Wang and Hu 2017) treat text as horizontal lines, which makes them difficult to be extended directly to irregular text. The performance of existing irregular text recognizers is far from being satisfactory. For instance, the current top-performing approach (Shi et al. 2018) only achieves 76.1% accuracy on the ICDAR 2015 benchmark (Karatzas et al. 2015).

Existing irregular text recognizers can be roughly categorized into three groups: rectification based (Shi et al. 2016; 2018; Liu et al. 2016; Liu, Chen, and Wong 2018), attention based (Yang et al. 2017; Cheng et al. 2017) and multi-direction encoding based (Cheng et al. 2018) approaches.

[*]The first two authors equally contributed to this work. C. Shen is the corresponding author.

(a) Original Image (b) 2D attention (ours) (c) Rectification (Shi et al. 2018)

Figure 1: The comparison of our proposed 2D attention based and the rectification based (Shi et al. 2018) irregular text recognizers. The second column gives the predictions of our approach and the heat map by aggregating attention weights at all character decoding steps; the third column demonstrates the rectified images and the corresponding predictions using the authors' implementation. Rectification based methods may encounter difficulties when the input image is severely curved or distorted. In contrast, we do not transform images and propose a tailored 2D attention module to localize individual characters in a weakly-supervised manner.

The rectification based methods attempt to transform irregular text patches into regular ones and then recognize them using regular text recognizers. However, as shown in Figure 1, severe distortions or curvatures give rise to difficulties for rectification. (Yang et al. 2017) proposed an attention mechanism to select local 2D features when decoding individual characters. Nevertheless, it needs extra character-level annotations to supervise the attention network and a multi-task strategy to learn better visual features. (Cheng et

al. 2018) stated that both rectification and attention based approaches are somewhat difficult to be directly trained on irregular text. They designed a sophisticated framework that needs to encode arbitrarily-oriented text in four directions.

Alternatively, we go back to the conventional attention based encoder-decoder framework. Our proposed model is composed of a 31-layer ResNet, an LSTM-based encoder-decoder framework and a tailored 2-dimensional attention module. In contrast to (Yang et al. 2017), we found that our proposed 2D attention module is able to approximately localize individual characters (see Figures 1 and 6) without additional supervision. Built upon standard NN modules, the main architecture can be implemented by around 100 lines of code. Despite its simplicity, our method outperforms previous methods on irregular text datasets by a large margin, and achieves comparable results on regular text. As demonstrated in Figure 1, our 2D attention module is more flexible and robust in handling sophisticated text layout.

For regular text, it is presented in (Lee and Osindero 2016; Shi et al. 2016) that the 1D attention based encoder-decoder framework is able to align between input subsequences and decoded characters. Our approach extends this framework by replacing 1D attention with a tailored 2D attention mechanism, in order to handle the complicated spatial layout of irregular text. Inspired by the success of the Show-Attend-and-Tell model (Xu et al. 2015) on image captioning, our model is also based on a 2D attention based encoder-decoder structure, which is referred to as Show-Attend-and-Read (SAR). Note that (Xu et al. 2015) is designed for image caption, while ours is used for text recognition.

The main contributions of this work is three-fold:

1) We setup an easy-to-implement strong baseline for recognizing irregular text in natural scene images, which is made up of off-the-shelf neural components such as CNNs, LSTMs and attention mechanisms. The proposed model can be trained end-to-end without pre-training. All the training examples are synthetic or from public real data. We will release the code and data used for training.

2) Compared to existing irregular text recognizers, our proposed approach does not rely on sophisticated designs (including spatial transformation, hierarchical attention or multi-directional encoding) to handle text distortions. Alternatively, we simply use a 2D attention mechanism to deal with irregular text, which selects local features for individual characters. Moreover, our proposed attention module does not require additional pixel-level or character-level supervision information, which is weakly supervised by the cross-entropy loss on the final predictions. The attention mechanism is also tailored to consider neighborhood information and boosts the recognition performance.

3) Note that many irregular text recognizers perform relatively worse on regular text. In contrast, due to its flexibility and robustness, the proposed approach not only significantly outperforms existing approaches on irregular text, but also achieves state-of-the-art performance on regular text.

Related Work

Early work Scene text recognition has drawn lots of attentions during recent years and made significant progress in performance. Early approaches mainly work in a *bottom-up* fashion (Wang, Babenko, and Belongie 2011; Mishra, Alahari, and Jawahar 2012b; Phan et al. 2013; Yao et al. 2014), in which individual characters are detected firstly via sliding window or connected components, and then integrated into a word by dynamic programming or graph models. Character detection or separation by itself, however, is not a completely-solved problem due to complicated background or cursive fonts. Alternatively, (Jaderberg et al. 2015a) considered text recognition as a multi-class classification problem, which assigned a distinct label to each word in a 90k-sized dictionary. Apparently, it is difficult to extend the approach to words out of the dictionary.

Regular Text Recognition (He et al. 2016b) and (Shi, Bai, and Yao 2017) considered words as one-dimensional sequences of varying lengths, and employed RNNs to model the sequences without explicit character separation. A Connectionist Temporal Classification (CTC) layer was adopted to decode the sequences. (Wang and Hu 2017) proposed a Gated Recurrent Convolutional Neural Network (GRCNN) with CTC for regular text recognition. Inspired by the sequence-to-sequence framework for machine translation, (Lee and Osindero 2016) and (Shi et al. 2016) proposed to recognize text using an attention-based encoder-decoder framework. In this manner, RNNs are able to learn the character-level language model hidden in the word strings from the training data. A 1D soft-attention model was adopted to select relevant local features during decoding characters. The RNN+CTC and sequence-to-sequence frameworks serve as two meta-algorithms that are widely used by subsequent text recognition approaches. Both models can be trained end-to-end and achieve considerable improvements on regular text recognition. (Bai et al. 2018) observed that the frame-wise maximal likelihood loss, which is conventionally used to train the encoder-decoder framework, may be confused and misled by missing or superfluity of characters, and thus degrade the recognition accuracy. To this end, they proposed "Edit Probability" to handle this misalignment problem. (Liu et al. 2018) presented a binary convolutional encoder-decoder network (B-CEDNet) together with a bidirectional recurrent neural network (Bi-RNN) for recognizing regular text images, and achieved significant speed-up. The whole framework needs to be trained in two stage, and requires pixel-level annotations. (Li, Wang, and Shen 2017) combined a Faster-RCNN based text detector and a 1D attention based recognizer into an end-to-end trainable system.

Irregular Text Recognition The rapid progress on regular text recognition has given rise to increasing attention on recognizing irregular ones. (Shi et al. 2018) and (Shi et al. 2016) rectified oriented or curved text based on Spatial Transformer Network (STN) (Jaderberg et al. 2015b) and then recognized it using a 1D attentional sequence-to-sequence model. (Liu et al. 2016) also removed text distortions via STN, and used the RNN + CTC framework for sequence recognition. Instead of rectifying the entire distorted text image as in (Shi et al. 2018; Liu et al. 2016), (Liu, Chen, and Wong 2018) presented a Character-Aware Neural Network (Char-Net) to detect and rectify individual characters,

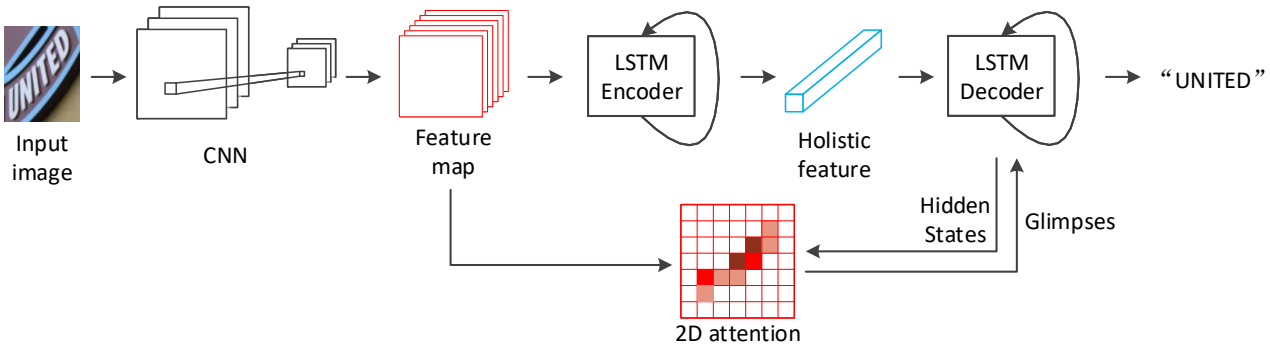

Figure 2: Overview of the proposed framework for irregular text recognition. The input image is firstly fed into a 31-layer ResNet, which results in a 2D feature map. Next, an LSTM model encodes the feature map column by column, and the last hidden state is considered as a holistic feature of the input image. Another LSTM model is used to decode the holistic feature into a sequence of characters. At each time step of decoding, an attention module computes a weighted sum of 2D features (glimpse), depending on the current hidden state of the LSTM decoder. The irregularity of text is implicitly handled by the 2D attention module, in a weakly supervised manner.

which, however, requires extra character-level annotations. Moreover, a sophisticated hierarchical attention mechanism was designed for accurate feature extraction, which consists of a recurrent RoIWarp layer and a character-level attention layer. (Yang et al. 2017) introduced an auxiliary dense character detection task into the encoder-decoder network to handle the irregular text. Pixel-level character/non-character annotations are required to train the network. (Cheng et al. 2017) asserted that there are "attention drifts" in traditional attention model and proposed a Focusing Attention Network (FAN) that is composed of an attention network for character recognition and a focusing network to adjust the attention drift. This work also needs to be trained with character-level bounding box annotations. (Cheng et al. 2018) applied LSTMs in four directions to encode arbitrarily-oriented text. A filtering mechanism was designed to integrate these redundant features and reduce irrelevant ones.

Model

We describe the architecture of our model in this section. As presented in Figure 2, the whole model consists of two main parts: a ResNet CNN for feature extraction and a 2D-attention based encoder-decoder model. It takes an image as input and outputs a varying length sequence of characters.

ResNet CNN

The designed 31-layer ResNet (He et al. 2016a) is presented in Table 1. For each residual block, we use the projection shortcut (done by 1×1 convolutions) if the input and output dimensions are different, and use the identity shortcut if they have the same dimension. All the convolutional kernel size is 3×3. Besides two 2×2 max-pooling layers, we also use a 1×2 max-pooling layer as in (Shi, Bai, and Yao 2017), which reserves more information along the horizontal axis and benefits the recognition of narrow shaped characters (*e.g.*, 'i', 'l'). The resulting 2D feature maps (denoted as $\mathbf{V}$ of size $H \times W \times D$ where D is the number of channels) will be used: 1) to extract holistic feature for the whole image; 2) as the context for the 2D attention network. To keep their original aspect ratios, we resize input images to a fixed height and a varying width. Hence, the width of the obtained feature map, W, also varies w.r.t. aspect ratios.

Layer name	Configuration
Conv	$3 \times 3, 64$
Conv	$3 \times 3, 128$
Max-pooling	k:2 $\times$ 2, s:2 $\times$ 2
Residual block	$\begin{bmatrix} Conv: 3 \times 3, 256 \\ Conv: 3 \times 3, 256 \end{bmatrix} \times 1$
Conv	$3 \times 3, 256$
Max-pooling	k:2 $\times$ 2, s:2 $\times$ 2
Residual block	$\begin{bmatrix} Conv: 3 \times 3, 256 \\ Conv: 3 \times 3, 256 \end{bmatrix} \times 2$
Conv	$3 \times 3, 256$
Max-pooling	k:1 $\times$ 2, s:1 $\times$ 2
Residual block	$\begin{bmatrix} Conv: 3 \times 3, 512 \\ Conv: 3 \times 3, 512 \end{bmatrix} \times 5$
Conv	$3 \times 3, 512$
Residual block	$\begin{bmatrix} Conv: 3 \times 3, 512 \\ Conv: 3 \times 3, 512 \end{bmatrix} \times 3$
Conv	$3 \times 3, 512$

Table 1: The configuration of the 31-layer ResNet for feature extraction. "Conv" stands for Convolutional layers, with kernel size and output channels presented. The stride and padding for convolutional layers are all set to "1". For Max-pooling layers, "k" means kernel size, and "s" represents stride. No padding for Max-pooling layers.

2D Attention based Encoder-Decoder

Sequence-to-sequence models have been widely used in machine translation, speech recognition and text recognition (Sutskever, Vinyals, and Le 2014) (Chorowski et al. 2015) (Cheng et al. 2017). In this work, we adopt a 2D attention based encoder-decoder network for irregular text recognition. Without transforming original text images, the proposed attention module is able to accommodate text of arbitrary shape, layout and orientation.

Encoder As shown in Figure 3, the encoder is a 2-layer

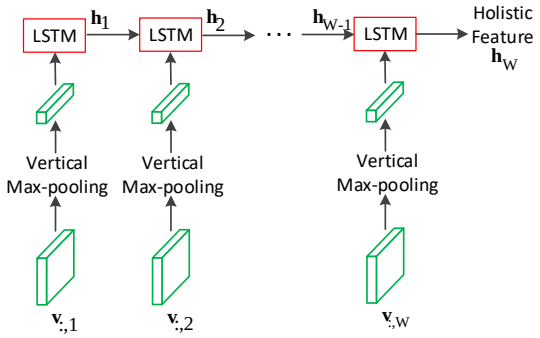

Figure 3: The structure of the LSTM encoder used in this work. $\mathbf{v}_{:,i}$ represents the ith column of the 2D feature map $\mathbf{V}$. At each time step, a column feature is firstly max pooled along the vertical direction, and then fed into LSTM.

LSTM model with 512 hidden state size per layer. At each time step, the LSTM encoder receives one column of the 2D features maps followed by max-pooling along the vertical axis, and updates its hidden state $\mathbf{h}_t$. After W steps, the final hidden state of the second LSTM layer, $\mathbf{h}_W$, is regarded as a fixed-size representation (holistic feature) of the input image, and provided for decoding.

Decoder As shown in Figure 4, the decoder is another LSTM model with 2 layers and 512 hidden state size per layer. The encoder and decoder do not share parameters. Initially, the holistic feature $\mathbf{h}_W$ is fed into the decoder LSTM, at time step 0. Then a "START" token is input into LSTM at step 1. From step 2, the output of the previous step is fed into LSTM until the "END" token is received. All the LSTM inputs are represented by one-hot vectors, followed by a linear transformation $\Psi()$. During training, the inputs of decoder LSTMs are replaced by the ground-truth character sequence. The outputs are computed by the following transformation:

$$\mathbf{y}_t = \varphi(\mathbf{h}'_t, \mathbf{c}_t) = \text{softmax}(\mathbf{W}_o[\mathbf{h}'_t; \mathbf{c}_t]) \tag{1}$$

where $\mathbf{h}'_t$ is the current hidden state and $\mathbf{c}_t$ is the output of the attention module. $\mathbf{W}_o$ is a linear transformation, which embeds features into the output space of 94 classes, in corresponding to 10 digits, 52 case sensitive letters, 31 punctuation characters, and an "END" token.

2D Attention Traditional 2D attention modules (Xu et al. 2015) treat each location independently, neglecting their 2D spatial relationships. In order to take neighborhood information into account, we propose a tailored 2D attention mechanism as follows:

$$\begin{cases} \mathbf{g}_{ij} = \tanh(\mathbf{W}_v \mathbf{v}_{ij} + \sum\limits_{p,q \in \mathcal{N}_{ij}} \tilde{\mathbf{W}}_{p-i,q-j} \cdot \mathbf{v}_{pq} + \mathbf{W}_h \mathbf{h}'_t), \\ \alpha_{ij} = \text{softmax}(\mathbf{w}_g^T \cdot \mathbf{g}_{ij}), \\ \mathbf{c}_t = \sum\limits_{i,j} \alpha_{ij} \mathbf{v}_{ij}, \quad i = 1, \ldots, H, \quad j = 1, \ldots, W. \end{cases} \tag{2}$$

where $\mathbf{v}_{ij}$ is the local feature vector at position (i, j) in $\mathbf{V}$, and $\mathcal{N}_{ij}$ is the eight-neighborhood around this position; $\mathbf{h}'_t$ is the hidden state of decoder LSTMs at time step t, to be used as the guidance signal; $\mathbf{W}_v$, $\mathbf{W}_h$ and $\tilde{\mathbf{W}}$s are linear transformations to be learned; α_{ij} is the attention weight at location (i, j); and $\mathbf{c}_t$ is the weighted sum of local features, denoted as a *glimpse*. Compared to traditional attention mechanisms, we add a term $\sum_{p,q \in \mathcal{N}_{ij}} \tilde{\mathbf{W}}_{p-i,q-j} \cdot \mathbf{v}_{pq}$ when computing the weight of $\mathbf{v}_{ij}$. We can see from Figure 5 that the computation of (2) can be accomplished by a series of convolution operations. Hence it is easy to implement.

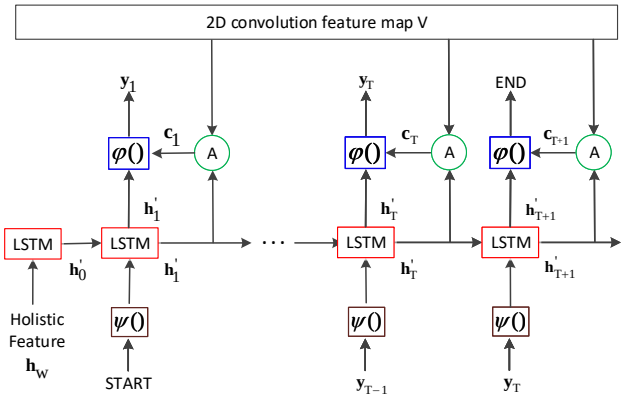

Figure 4: The structure of the LSTM decoder used in this work. The holistic feature $\mathbf{h}_W$, a "START" token and the previous outputs are input into LSTM subsequently, terminated by an "END" token. At each time step t, the output y_t is computed by $\varphi()$ with the current hidden state and the attention output as inputs.

Experiments

In this section, we perform extensive experiments to verify the effectiveness of the proposed method. We first show the datasets used for training and test, and then demonstrate the implementation details. Our model is compared with state-of-the-art methods on a number of public benchmark datasets, including both regular and irregular text in natural scene images. We also conduct ablation studies to analyze the impact of model hyper-parameters on performance.

Datasets

The following datasets are used in our experiments:

Synthetic Datasets There are two public available synthetic datasets that are widely used to train text recognizers: the 9-million synthetic data (refer to as **Syn90k**) released by (Jaderberg et al. 2015a) and the 8-million synthetic words (refer to as **SynthText**) proposed by (Gupta, Vedaldi, and Zisserman 2016). Images in **Syn90k** are generated based on 90k generic English words, while word instances in **SynthText** are from the Newsgroup20 lexicon (Lang 1995). Although they cover a huge number of word instances, the proportion of special characters like punctuations is relatively small. To compensate the lack of spatial characters, we synthesize additional 1.6-million word images (denoted as **SynthAdd**) using the synthetic engine proposed by (Gupta, Vedaldi, and Zisserman 2016). Special characters are randomly inserted to the words in the aforementioned two lexicons.

IIIT 5K-words (IIIT5K) (Mishra, Alahari, and Jawahar 2012a) contains 5000 word patches cropped from natural scene images found by Google image search, 2000 for training and 3000 for test. Text instances in these images are

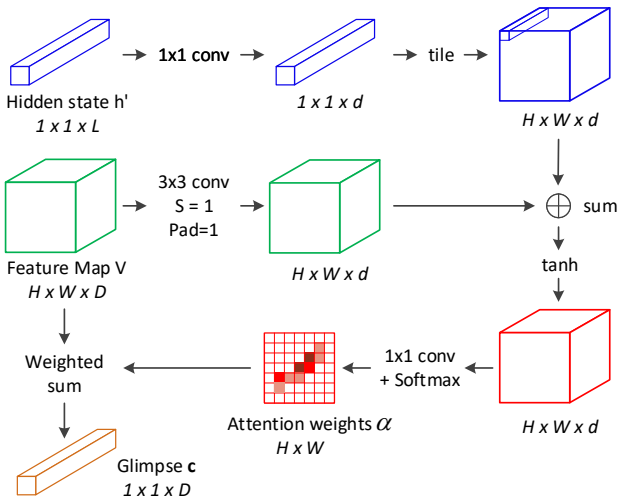

Figure 5: The computation of the proposed 2D attention mechanism can be simply implemented by convolutions, where $\mathbf{W}_v\mathbf{v}_{ij} + \sum_{p,q\in\mathcal{N}_{ij}} \tilde{\mathbf{W}}_{p-i,q-j} \cdot \mathbf{v}_{pq}$ is accomplished by a 3×3 convolution. The sizes of intermediate results are also demonstrated. The operation "tile" duplicates the input $1 \times 1 \times d$ vector $H \times W$ times.

nearly horizontal. Each image associates with a 50-word lexicon and a 1000-word lexicon individually.

Street View Text (SVT) (Wang, Babenko, and Belongie 2011) consists of 647 word patches cropped from Google Street View for test. They are nearly horizontal, but with noise, blur and low-resolution. Each image is associated with a 50-word lexicon.

ICDAR 2013 (IC13) (Karatzas et al. 2013) has 848 cropped word patches for training and 1095 for test. To fairly compare with previous results, we remove images that contain non-alphanumeric characters, which results in 1015 test patches. Words in this dataset are also mostly regular. No lexicon is provided.

ICDAR 2015 (IC15) (Karatzas et al. 2015) contains word patches cropped from incidental scene images captured under arbitrary angles. Hence most word patches in this dataset are irregular (oriented, perspective or curved). It contains 4468 patches for training and 2077 for test. No lexicon is associated.

Street View Text Perspective (SVTP) (Phan et al. 2013) consists of 639 word patches, which are cropped from side-view snapshots in Google Street View and encounter severe perspective distortions. All patches are used for test, with a 50-word lexicon and a Full lexicon for each image.

CUTE80 (CT80) (Risnumawan et al. 2014) contains 288 curved text images for test, with high resolution. No lexicon is associated.

COCO-Text (COCO-T) (Veit et al. 2016) contains more than 62k legible word patches cropped from more than 63k COCO images, including machine printed and handwritten, regular and irregular text. There are 42618 patches for training, 9896 for validation and 9837 for test. No lexicon is provided.

Implementation Details

The proposed model is implemented in Torch. All experiments are conducted on an NVIDIA Titan X GPU with 12GB memory. We simply use the cross-entropy loss for training. Without any pre-training, the whole network is end-to-end trained using the ADAM optimizer (Kingma and Ba 2014). We use a batch size of 32 at training time. The learning rate is set to 10^{-3} initially, with a decay rate of 0.9 every 10000 iterations until it reaches 10^{-5}.

Iteratively, we construct distinct data groups with 120k patches randomly sampled from **Syn90k**, 120k from **SynthText**, 80k from **SynthAdd** and approximately 50k training data from all the aforementioned public real datasets. Each group is trained for 2 epochs, and our algorithm converges after using 20 groups. In total, 2.4 million patches from **Syn90k**, 2.4 million from **SynthText** and 1.6 million from **SynthAdd** are used in the whole training process. The height of input images is resized to 48 pixels, and the width is calculated according to the original aspect ratio, but no longer than 160 and no smaller than 48 pixels.

At test time, for images with height larger than width, we will rotate the image by 90 degrees clockwise and anticlockwise respectively, and recognize them together with the original image. A recognition score will be calculated by averaging the output probabilities. The top-scored one will be chosen as the final recognition result. We use beam search for LSTM decoding, which keeps the top-k candidates with the highest accumulative scores, where k is empirically set to 5 in our experiments. Compared to greedy decoding that only picks the highest scored character at each time step, beam search brings an approximately 0.5% improvement to the recognition accuracy, in our practice. The test speed is 15ms per patch in average.

Experimental results

In this section, we evaluate our model on several regular and irregular text benchmarks, and compare the performance with other state-of-the-art methods. For datasets with lexicons provided, we simply select from lexicon the one with the minimum edit distance to the predicted word. The recognition results are summarized in Table 2.

On irregular text datasets (*i.e.*, IC15, SVTP, CT80 and COCO-T), our approach outperforms the compared methods by a large margin. In particular, our approach gives accuracy increases of 7.5% (78.9% to 86.4%) on SVTP-None and 10.1% (79.5% to 89.6%) on CT80. Note that neither SVTP or CT80 provides training data, which lowers the chance of over-fitting. Meanwhile, the proposed method still achieves state-of-the-art performance on regular text datasets (*i.e.*, IIIT5K, SVT and IC13). Actually, our model performs the best or the second best on 5 of the 6 evaluated regular text settings. The superiority of our method is more significant when there is no lexicon, such as in IIIT5K and SVTP. It demonstrates the practicality of our proposed approach in realistic scenarios where lexicon is rarely provided.

Examples of 2D attention heat maps when decoding individual characters are visualized in Figure 6. Although learned in a weakly supervised manner, the attention mod-

Table 2: Recognition accuracy (in percentages) on public benchmarks, including both regular and irregular text. "50", "1k", and "Full" are lexicon sizes, where "Full" means a combined lexicon of all images in the dataset. "None" means lexicon-free. The approaches marked with "*" are trained with both word-level and character-level annotations. In each column, the best performing result is shown in **bold** font, and the second best result is shown in *Italic* font. Our approach outperforms all the compared methods on all irregular text benchmarks, and achieves comparable performance on regular text.

Method	Regular Text						Irregular Text					
	IIIT5K			SVT		IC13	IC15	SVTP			CT80	COCO-T
	50	1k	None	50	None	None	None	50	Full	None	None	None
(Wang, Babenko, and Belongie 2011)	–	–	–	57.0	–	–	–	40.5	21.6	–	–	–
(Mishra, Alahari, and Jawahar 2012b)	64.1	57.5	–	73.2	–	–	–	45.7	24.7	–	–	–
(Phan et al. 2013)	–	–	–	73.7	–	–	–	75.6	67.0	–	–	
(Yao et al. 2014)	80.2	69.3	–	75.9	–	–	–	–	–	–	–	
(Jaderberg et al. 2015a)	97.1	92.7	–	95.4	80.7	90.8	–	–	–	–	42.7	
(He et al. 2016b)	94.0	91.5	–	93.5	–	–	–	–	–	–	–	
(Lee and Osindero 2016)	96.8	94.4	78.4	96.3	80.7	90.0	–	–	–	–	–	
(Wang and Hu 2017)	98.0	95.6	80.8	96.3	81.5	–	–	–	–	–	–	
(Shi et al. 2016)	96.2	93.8	81.9	95.5	81.9	88.6	–	91.2	77.4	71.8	59.2	–
(Liu et al. 2016)	97.7	94.5	83.3	95.5	83.6	89.1	–	*94.3*	83.6	73.5	–	
(Shi, Bai, and Yao 2017)	97.8	95.0	81.2	97.5	82.7	89.6	–	92.6	72.6	66.8	54.9	–
(Yang et al. 2017)*	97.8	96.1	–	95.2	–	–	–	93.0	80.2	75.8	69.3	–
(Cheng et al. 2017)*	99.3	97.5	87.4	97.1	85.9	93.3	70.6	92.6	81.6	71.5	63.9	–
(Liu et al. 2018)*	97.0	94.1	87.0	95.2	–	92.9	–	–	–	–	–	
(Liu, Chen, and Wong 2018)*	–	–	92.0	–	85.5	91.1	74.2	–	–	*78.9*	–	*59.3*
(Bai et al. 2018)*	*99.5*	97.9	88.3	96.6	87.5	**94.4**	73.9	–	–	–	–	
(Cheng et al. 2018)	**99.6**	98.1	87.0	96.0	82.8	–	68.2	94.0	*83.7*	73.0	76.8	
(Shi et al. 2018)	**99.6**	**98.8**	*93.4*	**99.2**	**93.6**	91.8	*76.1*	–	–	78.5	*79.5*	
SAR (Ours)	99.4	*98.2*	**95.0**	*98.5*	*91.2*	*94.0*	**78.8**	**95.8**	**91.2**	**86.4**	**89.6**	**66.8**

Table 3: Ablation studies by changing model hyper-parameters. The firstly row corresponds to the original proposed model configuration. Those changed parameters are shown in **bold** font. The models are evaluated on benchmarks without using any lexicon. Reducing the size of CNN and LSTM models has negative impacts on the recognition performance. Using the traditional $2D$ attention or 1D attention modules, instead of our proposed attention mechanism, also degrades the accuracy.

Model Configuration					IIIT5K	SVT	IC13	IC15	SVTP	CT80	COCO-T
CNN channels	Down-sampling ratio	Attention module	LSTM layers	Hidden state size							
$\times 1$	$1/8, 1/4$	2D proposed	2	512	95.0	91.2	94.0	78.8	86.4	89.6	66.8
$\times \mathbf{1/2}$	$1/8, 1/4$	2D proposed	2	512	92.7	88.7	92.0	75.6	81.3	86.8	62.6
$\times 1$	$\mathbf{1/16}, 1/4$	2D proposed	2	512	93.8	90.3	92.7	77.4	84.5	89.2	64.8
$\times 1$	$\mathbf{1/16}, \mathbf{1/8}$	2D proposed	2	512	94.0	90.6	93.1	76.2	83.7	87.5	63.7
$\times 1$	$1/8, \mathbf{1/8}$	2D proposed	2	512	93.6	89.3	92.5	76.1	82.8	87.5	63.3
$\times 1$	$1/8, 1/4$	**2D traditional**	2	512	94.0	90.1	92.3	77.2	84.3	87.5	64.2
$\times 1$	$1/8, 1/4$	**1D**	2	512	93.0	89.9	90.2	76.6	83.6	84.7	65.4
$\times 1$	$1/8, 1/4$	2D proposed	**1**	512	89.7	87.2	87.4	70.6	76.4	80.6	60.1
$\times 1$	$1/8, 1/4$	2D proposed	2	**256**	94.0	89.3	92.8	76.8	83.7	86.5	63.8

ule can still approximately localize characters being decoded, extract discriminative local features and finally help text recognition. Note that the proposed attention module is trained without character-level annotations.

Some failure cases are also presented in Figure 8. There are a variety of reasons for failure, such as blurry, partial occlusion, extreme distortion, uneven lighting condition, uncommon fonts, vertical text, *etc*. Scene text recognition still has a long way to be completely solved.

Ablation Studies

In order to analyze the impact of different model hyper-parameters on the recognition performance, we perform a series of ablation studies as presented in Table 3. All the evaluated models in this section are trained from scratch with the same data and tested on benchmarks without lexicon.

CNN Parameters We firstly reduce by 50% the number of channels of all convolutional layers expect the last layer, which lowers the accuracy by 2 to 4 percentages. The down-sampling ratio of the proposed ResNet is $1/8$ vertically and $1/4$ horizontally, which results in a feature map of maximum size 6×40. Here we further divide the vertical and/or horizontal down-sampling ratios by 2, and obtain worse performance. These results shows that the volume of feature maps should be sufficiently large to encode a large variety of visual information for text recognition.

Attention Modules The proposed 2D attention model is respectively replaced by the traditional 2D counterpart with

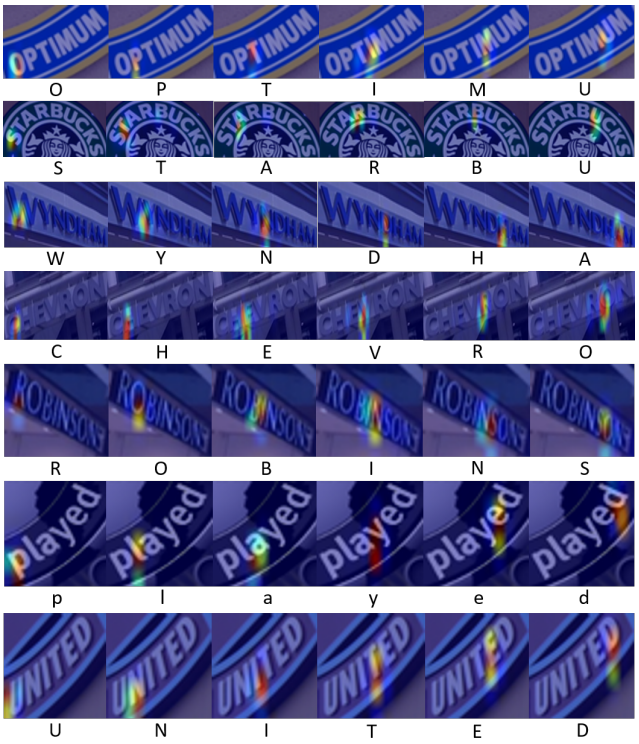

Figure 6: Visualization of 2D attention weights at individual decoding time steps, which shows that our 2D attention model can be trained to approximately localize characters without character-level annotations. For space reasons, some of the decoding results are truncated.

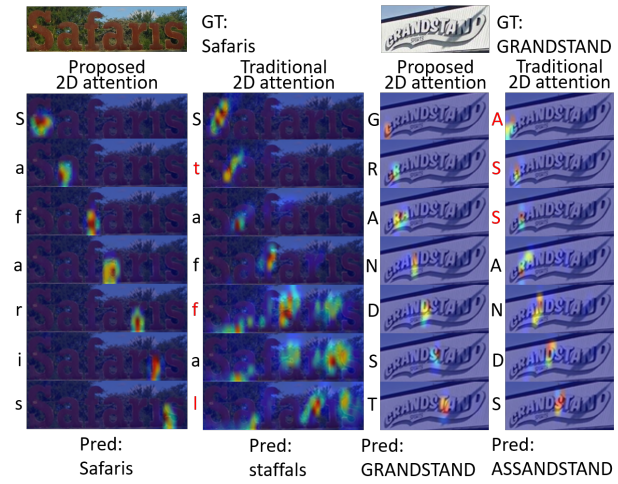

Figure 7: Comparison of our proposed 2D attention model and the traditional 2D attention model. The decoded characters are shown to the left of the corresponding attention heat maps, with incorrect ones marked in red. The proposed model shows more accurate localization and better recognition results.

the term $\sum_{p,q\in\mathcal{N}_{ij}}\tilde{\mathbf{W}}_{p-i,q-j}\cdot\mathbf{v}_{pq}$ removed from Equation (2) and a 1D attention module that considers feature maps as 1D sequences. By aggregating neighborhood information, the proposed 2D attention model outperforms the traditional 2D one by 1 to 2 percentages. Both of the 2D attention modules performs better than the 1D one in most cases, which shows their robustness for both regular and irregular text recognition. We compare the proposed and the traditional 2D attention heat maps in Figure 7. The proposed model presents better performance on character localization and recognition.

Figure 8: Failure cases of our model. "GT" stands for the ground-truth annotation, and "Pred" denotes the predicted results.

LSTM Parameters By cutting down by half the hidden state size of both encoder and decoder LSTMs, we receive degraded recognition accuracies. The performance degradation is more serious when we use 1 layer of LSTMs, instead of 2 layers. Relatively, the number of LSTM layers presents a stronger impact on the performance.

Conclusion

In this work, we present a simple yet strong baseline for irregular text recognition. The proposed framework is built upon off-the-shelf neural network modules, including a ResNet CNN, an LSTM encoder-decoder and a 2D tailored attention module. Without any extra supervision information, the proposed attention mechanism is capable of select local features for decoding characters. Being robust to different forms of text layouts, our approach performs well for both regular and irregular text.

As to future works, the proposed framework can be extended in several ways. Firstly, the LSTM encoder-decoder is possible to be replaced by CNNs for sequence modeling, which will further ease the training process. Secondly, the proposed 2D attention module can be seen as a special case of graph neural networks, where edges of the graph are defined on 8-neighborhoods. Straightforwardly, we can apply the attention mechanism on graphs with more complex structures, to incorporate with the rich context information. Finally, to better learn visual features and speedup the training process, we can also add a word classification head apart from the LSTM decoder.

References

Bai, F.; Cheng, Z.; Niu, Y.; Pu, S.; and Zhou, S. 2018. Edit probability for scene text recognition. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*

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

Cheng, Z.; Xu, Y.; Bai, F.; Niu, Y.; Pu, S.; and Zhou, S. 2018. AON: Towards arbitrarily-oriented text recognition. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*

Chorowski, J.; Bahdanau, D.; Serdyuk, D.; Cho, K.; and Bengio, Y. 2015. Attention-based models for speech recognition. In *Proc. Adv. Neural Inf. Process. Syst.*, 577–585.

Gupta, A.; Vedaldi, A.; and Zisserman, A. 2016. Synthetic data for text localisation in natural images. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2315–2324.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016a. Deep residual learning for image recognition. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*

He, P.; Huang, W.; Qiao, Y.; Loy, C. C.; and Tang, X. 2016b. Reading scene text in deep convolutional sequences. In *Proc. National Conf. Artificial Intell.*

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

Jaderberg, M.; Simonyan, K.; Zisserman, A.; et al. 2015b. Spatial transformer networks. In *Proc. Adv. Neural Inf. Process. Syst.*, 2017–2025.

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

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

Kingma, D., and Ba, J. 2014. Adam: A method for stochastic optimization. In *Proc. Int. Conf. Learn. Representations*.

Lang, K. 1995. Newsweeder: Learning to filter netnews. In *Proceedings of the Twelfth International Conference on Machine Learning*, 331–339.

Lee, C.-Y., and Osindero, S. 2016. Recursive recurrent nets with attention modeling for ocr in the wild. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*

Li, H.; Wang, P.; and Shen, C. 2017. Towards end-to-end text spotting with convolutional recurrent neural networks. In *Proc. IEEE Int. Conf. Comp. Vis.*, 5238–5246.

Liu, W.; Chen, C.; Wong, K.-Y. K.; Su, Z.; and Han, J. 2016. Star-net: A spatial attention residue network for scene text recognition. In *Proc. British Mach. Vis. Conf.*

Liu, Z.; Li, Y.; Ren, F.; Goh, W. L.; and Yu, H. 2018. Squeezedtext: A real-time scene text recognition by binary convolutional encoder-decoder network. In *Proc. National Conf. Artificial Intell.*

Liu, W.; Chen, C.; and Wong, K.-Y. K. 2018. Char-net: A character-aware neural network for distorted scene text recognition. In *Proc. National Conf. Artificial Intell.*

Mishra, A.; Alahari, K.; and Jawahar, C. V. 2012a. Scene text recognition using higher order language priors. In *Proc. British Mach. Vis. Conf.*, 1–11.

Mishra, A.; Alahari, K.; and Jawahar, C. V. 2012b. Top-down and bottom-up cues for scene text recognition. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*

Phan, T. Q.; Shivakumara, P.; Tian, S.; and Tan, C. L. 2013. Recognizing text with perspective distortion in natural scenes. In *Proc. IEEE Int. Conf. Comp. Vis.*, 569–576.

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

Shi, B.; Bai, X.; and Yao, C. 2017. 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):2298–2304.

Shi, B.; Wang, X.; Lv, P.; Yao, C.; and Bai, X. 2016. Robust scene text recognition with automatic rectification. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*

Shi, B.; Yang, M.; Wang, X.; Lyu, P.; Yao, C.; and Bai, X. 2018. Aster: An attentional scene text recognizer with flexible rectification. *IEEE Trans. Pattern Anal. Mach. Intell.* 1–1.

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to sequence learning with neural networks. In *Proc. Adv. Neural Inf. Process. Syst.*

Veit, A.; Matera, T.; Neumann, L.; Matas, J.; and Belongie, S. 2016. Coco-text: Dataset and benchmark for text detection and recognition in natural images. In *arXiv preprint arXiv:1601.07140*.

Wang, J., and Hu, X. 2017. Gated recurrent convolution neural network for ocr. In *Proc. Adv. Neural Inf. Process. Syst.*

Wang, K.; Babenko, B.; and Belongie, S. 2011. End-to-end scene text recognition. In *Proc. IEEE Int. Conf. Comp. Vis.*, 1457–1464.

Xu, K.; Ba, J.; Kiros, R.; Courville, A.; Salakhutdinov, R.; Zemel, R.; and Bengio, Y. 2015. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In *Proc. Int. Conf. Mach. Learn.*

Yang, X.; He, D.; Zhou, Z.; Kifer, D.; and Giles, C. L. 2017. Learning to read irregular text with attention mechanisms. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, 3280–3286.

Yao, C.; Bai, X.; Shi, B.; and Liu, W. 2014. Strokelets: A learned multi-scale representation for scene text recognition. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*