Attention-based Text Recognition in the Wild

Zhi-Chen Yan[1] and Stephanie A. Yu[2]
[1]*Facebook Research, 1 Hacker Way, Menlo Park, CA 94025, U.S.A.*
[2]*West Island School, 250 Victoria Road, Pokfulam, Hong Kong, Republic of China*

Keywords: Attention, Convolution, Deep Learning, LSTM, Text Recognition.

Abstract: Recognizing texts in real-world scenes is an important research topic in computer vision. Many deep learning based techniques have been proposed. Such techniques typically follow an encoder-decoder architecture, and use a sequence of feature vectors as the intermediate representation. In this approach, useful 2D spatial information in the input image may be lost due to vector-based encoding. In this paper, we formulate scene text recognition as a spatiotemporal sequence translation problem, and introduce a novel attention based spatiotemporal decoding framework. We first encode an image as a spatiotemporal sequence, which is then translated into a sequence of output characters using the aforementioned decoder. Our encoding and decoding stages are integrated to form an end-to-end trainable deep network. Experimental results on multiple benchmarks, including IIIT5k, SVT, ICDAR and RCTW-17, indicate that our method can significantly outperform conventional attention frameworks.

1 INTRODUCTION

Scene text recognition remains a hot research topic in computer vision (Neumann and Matas, 2012; Jaderberg et al., 2014; Shi et al., 2016a) due to important applications including handwriting recognition and navigation reading.

With the advance of deep learning, many solutions (Lee and Osindero, 2016; Shi et al., 2016b; Yang et al., 2017; Cheng et al., 2017; Wang and Hu, 2017) have been proposed for scene text recognition and promising results have been achieved. In general, such solutions exploit the encoder-decoder network architecture. Specifically, in the encoding stage, a sequence of feature vectors are extracted from an input image using convolutional neural networks (CNNs) (Sainath et al., 2013) and recurrent neural networks with long-short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997). In the decoding stage, the feature vectors are decoded into a character sequence with connectionist temporal classification (CTC) (Graves et al., 2006) or an attention-based temporal decoder (Lee and Osindero, 2016; Shi et al., 2016a; Yang et al., 2017; Cheng et al., 2017). In fact, any character appearing in a natural image contains specific spatial information, such as the stroke layout of the character. However, previous methods, as shown in Figure 1, encode a spatial image using feature vectors and generate an output character sequence by performing classification on the feature vectors, whose one-dimensional layout may miss part of the spatial and structural information in the original two-dimensional image. An ideal network architecture we seek should retain the original spatial structures and thus make the best of spatiotemporal information. Prior to building such a spatiotemporal model, several issues need to be taken into consideration:

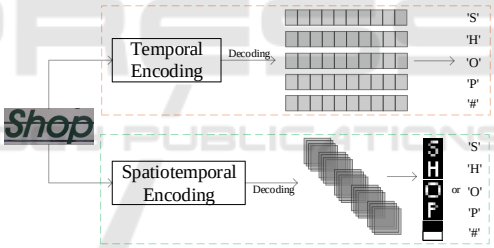

Figure 1: Overview of the proposed model. The brown dashed box refers to the traditional framework for scene text reading: first encoding an image as a sequence of feature vectors (denoted *temporal encoding*), and then performing classification; the green dashed box represents our proposed framework: first encoding an image as a sequence of feature maps (denoted *spatiotemporal encoding*), and then generating a sequence of characters.

Yan, Z. and Yu, S.
Attention-based Text Recognition in the Wild.
DOI: 10.5220/0009970200420049
In *Proceedings of the 1st International Conference on Deep Learning Theory and Applications (DeLTA 2020)*, pages 42-49
ISBN: 978-989-758-441-1

- **Spatiotemporal Encoding:** Unlike the previous methods, our method should have a scheme that efficiently constructs a spatiotemporal sequence from a spatial image. Inspired by (Li et al., 2017), we use a sliding window to build the spatiotemporal sequence.
- **Spatiotemporal Decoding:** Given a constructed spatiotemporal sequence, a spatiotemporal decoder should be used to generate a corresponding character sequence. Unfortunately, no techniques are available for this purpose. In the existing literature, there are two temporal decoders: CTC and attention. CTC was designed for calculating the conditional probability between a predicted character sequence and its corresponding target character sequence, but cannot map a spatiotemporal sequence of feature vectors to the target sequence while a *fully connected* attention (FC-Attention) decoder can only map a sequence of feature vectors (instead of 2D feature maps) to a target sequence. Inspired by convolutional LSTM (ConvLSTM) (Xingjian et al., 2015), we attempt to design a *convolutional* attention mechanism that can build the mapping from an encoded spatiotemporal sequence to the corresponding target sequence.
- **Model Training:** We integrate the encoding and decoding modules to form an end-to-end network, which can be trained with the cross-entropy loss between predicted sequences generated by ConvAttention and their corresponding target sequences, as shown in Figure 1.

As we know that the FC-Attention layer adopted by previous models does not take spatial correlation into consideration, we extend the idea of FC-Attention to ConvAttention, which has convolutional structures in both input-to-state and state-to-state transitions. With ConvAttention, we can build an end-to-end trainable network for describing scene text using character-level labels.

In summary, this paper has the following contributions.

- We propose a novel spatiotemporal deep learning framework with a convolutional attention mechanism, which retains more information about spatial structures. This framework can be trained from end to end.
- Our proposed framework can be configured as an end-to-end spatiotemporal model for robustly reading scene text. Our convolutional attention can effectively transform each scene text image into a sequence of characters or target images, as shown in Figure 1.
- Extensive experiments on public text benchmarks demonstrate that our convolutional attention mechanism significantly outperforms conventional attention frameworks.

2 RELATED WORK

Reading text from natural image is still one of the most important challenges in computer vision and many methods have been proposed. A complete text reading system contains a text detection module and a text recognition module. Our work in this paper focuses on the text recognition task.

Conventional methods typically first locate characters one by one with a sliding window, then recognize characters using a classifier with handcrafted features such as HOG descriptors (Yao et al., 2014), and finally integrate recognized characters into the output text (Neumann and Matas, 2012; Wang and Belongie, 2010; Wang et al., 2011). However, two problems limit the performance of such methods: the low representation capability of handcrafted features and missing contextual information in the pipeline. With the advance in deep learning and convolutional neural networks, researchers use CNNs for extracting high-level feature representations. Jaderberg *et al.* (Jaderberg et al., 2016) carried out a 90k-class classification task with high-level features. In this task, each class represents a character string, therefore, their method cannot recognize out-of-vocabulary words. Wang *et al.* (Wang et al., 2012) developed a CNN-based feature extraction framework for character recognition, and then performed non-maximum suppression for final word prediction. The above models are trained with the segmentation annotation of each character, and do not exploit contextual information in the original text. In addition, annotating segmentations is very labor-intensive especially when the background is cluttered.

Recently, some works regard this problem as a temporal sequence recognition problem, and recurrent neural networks (RNNs) are integrated with CNNs to read character sequences. The CTC (Graves et al., 2006) loss is combined with RNNs in (He et al., 2016; Shi et al., 2016a; Wang and Hu, 2017) to calculate the conditional probability between the predicted and target sequences. Attention-based decoders are used for generating output sequences in (Lee and Osindero, 2016; Shi et al., 2016b; Cheng et al., 2017; Yang et al., 2017). The above methods have achieved promising results on text recognition, but all of them encode a spatial text image into a sequence of feature vectors, which may lose part of

the spatial information. We believe that information about spatial structures is helpful in describing text images.

Different from previous approaches for scene text reading, in this paper we describe scene text from a spatiotemporal perspective, and propose an end-to-end trainable network to build spatiotemporal sequence models for scene text. To the best of our knowledge, this is the first piece of work that applies a spatiotemporal framework to scene text reading.

3 PRELIMINARIES: FC-Attention

An attention-based decoder is a recurrent neural network that directly generates a target sequence $(y_1,...,y_M)$ from input feature vectors $(h_1,...,h_T)$, where the lengths of the input and output sequences may be different Bahdanau *et al.* (Bahdanau et al., 2015) proposed the architecture of FC-Attention, as shown in Figure 2. At the t-th step, the attention module generates an output y_t as follows:

$$\begin{aligned} y_t &\sim softmax(Us_t), \\ s_t &= LSTM(y_{t-1}, g_t, s_{t-1}), \\ g_t &= \textstyle\sum_{k'=1}^{T} \alpha_{t,k'} h_{k'}, \\ \alpha_{t,k} &= \frac{\exp(e_{t,k})}{\sum_{k'=1}^{T} \exp(e_{t,k'})}, \\ e_{t,k} &= w\tanh(Ws_{t-1} + Vh_k + b) \end{aligned} \tag{1}$$

where s_t, g_t, α_t and e_t represent the hidden state of the LSTM, the weighted sum of h, the attention weights and the energy value at the t-th step, respectively. In the above equation, w, W, V, U and b are all trainable parameters.

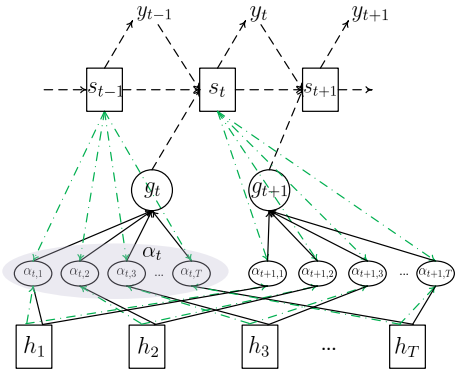

Figure 2: Illustration of FC-Attention. Green dotted and dashed lines correspond to the calculation of attention weights α.

FC-Attention is capable of generating sequences of variable lengths. Following (Sutskever et al., 2014), a special end-of-sequence (EOS) token is added to the target set so that the decoder completes the generation of characters when EOS is emitted.

4 PROPOSED METHOD

In this section, we present our proposed ConvAttention network. Although the FC-Attention layer has proven powerful for handling temporal sequence generation, full connections contain too much redundancy for spatial data. To address this problem, we propose to extend FC-Attention to ConvAttention, which has convolutional structures in both input-to-state and state-to-state transitions. By integrating ConvAttention with a spatiotemporal encoder shown in Figure 3, we are able to build an end-to-end spatiotemporal sequence model.

4.1 Convolutional Attention

Inspired by the processes within FC-Attention and ConvLSTM, we design our convolutional attention mechanism as follows. ConvAttention generates a target sequence $(\mathcal{Y}_1,...,\mathcal{Y}_M)$ from a sequence of input feature maps $\mathcal{H}:(\mathcal{H}_1,...,\mathcal{H}_T)$, where T and M may not be equal. At the t-th step, the convolutional attention module generates an output $\mathcal{Y}_t$ as follows, where $*$ denotes the convolution operator:

$$\mathcal{Y}_t \sim Generate(U * \mathcal{S}_t), \tag{2}$$

where U is a trainable weight template, and $\mathcal{S}_t$ is the hidden state of ConvLSTM (Xingjian et al., 2015) at time t, computed by:

$$\mathcal{S}_t = ConvLSTM(\mathcal{Y}_{t-1}, \mathcal{G}_t, \mathcal{S}_{t-1}), \tag{3}$$

where $\mathcal{G}_t$ is a weighted sum of spatiotemporal feature maps $\mathcal{H}$. That is,

$$\mathcal{G}_t = \textstyle\sum_{k'=1}^{T} \alpha_{t,k'} \mathcal{H}_{k'}, \tag{4}$$

where α_t represents the set of spatiotemporal attention weights for the t-th step. During the computation of attention weights, α_t is often evaluated by scoring each element in $\mathcal{H}$ separately and normalizing the scores as follows:

$$\alpha_t = Attend(\mathcal{S}_{t-1}, \mathcal{H}), \tag{5}$$

where *Attend* denotes the attending process to be elaborated later.

Generate process: The generation function in Equation (2) emphasizes the mapping from ConvLSTM hidden state $\mathcal{S}_t$ to $\mathcal{Y}_t$. For example, the mapping function can be a spatially fully connected layer

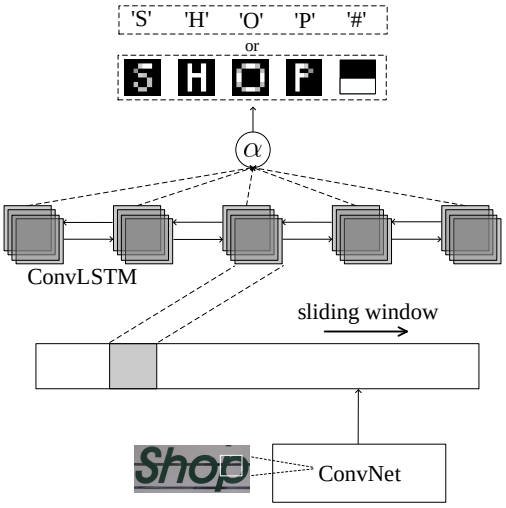

Figure 3: Our spatiotemporal text recognition network. The network consists of an encoder and a decoder. For each image, the encoder uses convolution layers (ConvNet) to extract a high-level feature representation, then generates a spatiotemporal sequence with a sliding window. Afterwards, the decoder with a convolutional attention mechanism generates a predicted image sequence or character sequence conditioned on the output of the encoder.

with the softmax operator for sequentially generating *scalar labels* y (e.g. 'A', 'B', ...) or a spatially invariant convolution kernel for generating 2D *spatial label maps* y_I, as shown in Figure 3.

Attend process: The attending process prescribes the weight of each input feature map $\mathcal{H}_k \in \mathbb{R}^{filters \times width \times height}$, where *filters*, *width* and *height* refers to the number of channels, width and height of the feature map, respectively. This process can be defined as follows:

$$\mathcal{E}_{t,k} = w * \tanh(W * \mathcal{S}_{t-1} + V * \mathcal{H}_k + b), \quad (6)$$

where w, W, and V are trainable 4D weight templates, b is a trainable 3D offset map, and $*$ denotes the convolution operator. Here, there are two modalities for $\mathcal{E}_{t,k}$:

- If $\mathcal{E}_{t,k} \in \mathbb{R}^1$, a scalar weight $\alpha_{t,k}$ is applied to a given feature map, and it is computed by

$$\alpha_{t,k} = \frac{\exp(\mathcal{E}_{t,k})}{\sum_{k'=1}^{T} \exp(\mathcal{E}_{t,k'})}, \quad (7)$$

  which is called *temporal weight*, abbreviated as *tw*.

- If $\mathcal{E}_{t,k} \in \mathbb{R}^{W \times H}$, a pixel-wise weight mask is applied to a given feature map. For each pixel (i, j) in $\mathcal{H}_k$, the weight $\alpha_{t,k}^{(i,j)}$ is computed by

$$\alpha_{t,k}^{(i,j)} = \frac{\exp(\mathcal{E}_{t,k}^{(i,j)})}{\sum_{k'=1}^{T} \exp(\mathcal{E}_{t,k'}^{(i,j)})}, \quad (8)$$

  which is called *spatiotemporal weight*, abbreviated as *stw*.

The attending process of FC-Attention in Equation (1) can be seen as a special case of ConvAttention with the last two dimensions of $\mathcal{H}_k$ being 1. In addition, the padding strategy used by ConvAttention for the convolution operator is the same as ConvLSTM (Xingjian et al., 2015).

4.2 Network Training

We first encode a spatial image into a spatiotemporal sequence using a spatiotemporal feature extractor consisting of convolutional layers (ConvNet) and a sliding window, as shown in Figure 3, and then integrate this spatiotemporal feature extractor with our convolutional attention module.

We can train the entire network with either *spatial label maps* or *scalar labels*. In the case of training with spatial label maps, the following cross-entropy loss is used.

$$\mathcal{L}_c = -\sum_{t,i,j} \hat{y}_I^{t,i,j} log y_I^{t,i,j} + (1 - \hat{y}_I^{t,i,j}) log(1 - y_I^{t,i,j}), \quad (9)$$

where y_I^t and $\hat{y}_I^t$ are the *t-th* predicted image and the *t-th* target image, respectively.

In the case of training with scalar labels, the loss function of the network is defined as follows.

$$\mathcal{L}_s = -\sum_t lnP(\hat{y}_t|\theta), \quad (10)$$

where $\hat{y}_t$ is the ground truth of the *t-th* character and θ includes all the network parameters.

4.3 Evaluation Metrics

If we train our end-to-end network with the cross-entropy loss $\mathcal{L}_c$, we use the cross-entropy value between the predicted images and the target images to evaluate the performance of our framework. The smallest cross-entropy value with respect to the target images indicates the most probable predictions.

On the other hand, if we train our network with the loss $\mathcal{L}_s$, we follow existing works (Lee and Osindero, 2016; Shi et al., 2016b; Yang et al., 2017; Cheng et al., 2017) and compute the accuracy of the generated character sequences with respect to the target character sequences.

5 EXPERIMENTS

We first compare our ConvAttention network with the FC-Attention network on a synthetic dataset released

by Jaderberg *et al.* (Jaderberg et al., 2014) to gain basic understanding of the behavior of our network. We train our network with different numbers of convolution kernels and different kernel sizes. To verify the effectiveness of our network on more challenging scene text reading tasks, we further test our trained model on several benchmarks and compare it with the state of the art.

5.1 Explorative Study

Experimental setting: We randomly choose 200k training images, 10k validation images and 10k testing images from the synthetic dataset released by Jaderberg *et al.* (Jaderberg et al., 2014). All images are resized to 32×256. We have implemented our network in the Caffe framework (Jia et al., 2014). We train all the Attention models by minimizing the cross-entropy loss $\mathcal{L}_c$ or loss $\mathcal{L}_s$ using back-propagation through time (BPTT) (Rumelhart et al., 1988) and stochastic gradient descent with momentum set to 0.95, weight-decay set to 0.0005, and batch-size set to 64. For $\mathcal{L}_c$ and $\mathcal{L}_s$, the base learning rate is set to 0.0001 and 0.01, respectively. Also, we perform the stopping operation on the validation set until 120k iterations. We run all the experiments on a computer with eight NVIDIA Tesla P40 GPUs.

Network details: For the ConvNet in Figure 3, we use 7 convolutional layers. This is similar to the encoder architecture proposed in (Shi et al., 2016b). The {filter size, number of filters, stride, padding size} in these 7 layers are respectively {3,64,1,1}, {3,128,1,1}, {3,256,1,1}, {3,256,1,1}, {3,512,1,1}, {3,512,1,1} and {3,512,1,1}. The 1st and 2nd convolution layers are each followed by a 2×2 max-pooling layer. We adopt batch normalization (BN) and ReLU activation right after the 5th and 6th convolution layers. On top of the convolution layers is a bidirectional ConvLSTM, which forms the basis of our ConvAttention module. In addition, in order to construct the spatiotemporal sequence, a sliding window with a step size set to 2 is used over the output of the 7-th convolutional layer. Note that FC-LSTM/FC-Attention can be seen as a special case of ConvLSTM/ConvAttention with the last two dimensions of $\mathcal{H}_k$ being set to 1.

Performance analysis: The results of 40 experiments are shown in Table 1, where text reading performance with different settings is compared. We evaluate the performance of ConvAttention in capturing spatiotemporal correlations as follows: 1) 20 experiments comparing the cross-entropy values which indicate the quality of generated spatial images, 2) 20 experiments comparing the accuracy of the generated character sequences. We set the number of filters to 512 in FC-Attention, and trained FC-Attention models respectively for generating spatial label maps and scalar character sequences are used as our baseline. We conduct three groups of experiments related to ConvAttention to explore its performance by varying the resolution of feature maps, the kernel size and the number of kernels. The kernel size and the resolution of feature maps in ConvLSTM are the same as their counterparts in ConvAttention, and the number of kernels in ConvLSTM is always set to 64. Given extensive comparative experiments shown in Table 1, we can conclude that

- ConvAttention significantly outperforms FC-Attention in handling spatiotemporal correlations which help boost the recognition performance.
- Making the kernel size bigger than 1 is useful for improving the recognition results.
- By varying the number of kernels from 128 to 8, ConvAttention still maintains a high performance, which demonstrates that ConvAttention is robust and stable.
- The performance of ConvAttention is reasonably affected by the resolution of feature maps because a larger feature map contains more spatial information. This indirectly indicates the importance of spatial information.
- Spatiotemporally weighted ConvAttention is better than temporally weighted ConvAttention.

5.2 Comparison with the State of the Art

Network details and Environments: For further demonstration of the performance of our method, we directly choose a few state-of-the-art networks, such as FC-Att, ConvAtt1_3, and ConvAtt3_2, listed in Table 1 for comparison. In this comparison, we terminate training after 850k iterations.

Training dataset: According to previous work (Cheng et al., 2017), our training set consists of the 8 million synthetic data released by Jaderberg et al. (Jaderberg et al., 2014) and 4 million synthetic instances (excluding the images that contain non-alphanumeric characters) cropped from the 80 thousand images released by (Gupta et al., 2016).

Testing dataset: We have collected three benchmarks as our testing datasets. **IIIT5K-Words** (IIIT5K in short) (Mishra et al., 2012) wass collected from the Internet, containing 3000 cropped word images in its test set. **Street View Text** (SVT in short) (Wang et al., 2011) was collected from

Table 1: Comparison of ConvAttention (ConvAtt) with FC-Attention (FC-Att) on selected datasets. '#MS' and '#Mul-Add' respectively mean the number of parameters in ConvAttention and the number of multiply-add operations in ConvAttention for predicting a character in Equations (3) and (6). 'FS', 'KS' and 'ACC' represent the resolution of feature maps, the kernel size, and the accuracy of a model trained with $\mathcal{L}_c$ or $\mathcal{L}_s$. *tw* and *stw* respectively refer to the results generated with the *temporally weighted* attending operation and the *spatiotemporally weighted* attending operation, which have been separately described in Equations (7) and (8). 'M' means million. Specifically, for FS equal to 1 and 4, the (4th, 6th) and (4th) convolution layers are each followed by a 2×1 max-pooling layer, respectively; for FC-Att, the {kernel size, number of kernels, stride, padding size} of the 7-th convolutional layer is set to {2,512,1,0}.

Model	#MS	#Mul-Add	FS	KS	Filters	ACC by $\mathcal{L}_c$(*tw*/*stw*)	ACC by $\mathcal{L}_s$(*tw*/*stw*)
FC-Att	1.0496 M	5.01 M	1	1×1	512	75.76 ± 0.09/ -	82.29 ± 0.12/ -
ConvAtt1_1	0.1486 M	0.71 M	1	3×3	64	75.32 ± 0.05/ -	82.23 ± 0.10/ -
ConvAtt1_2	0.1486 M	11.35 M	4	3×3	64	78.28 ± 0.02/78.02 ± 0.04	83.02 ± 0.20/83.24 ± 0.04
ConvAtt1_3	0.1486 M	45.40 M	8	3×3	64	78.60 ± 0.03/79.93 ± 0.07	83.50 ± 0.26/83.64 ± 0.08
ConvAtt2_1	0.0024 M	0.75 M	8	3×3	8	77.94 ± 0.03/78.08 ± 0.05	83.23 ± 0.07/83.43 ± 0.03
ConvAtt2_2	0.0095 M	2.91 M	8	3×3	16	78.55 ± 0.01/79.30 ± 0.05	83.55 ± 0.10/83.74 ± 0.02
ConvAtt2_3	0.0374 M	11.45 M	8	3×3	32	79.38 ± 0.05/80.03 ± 0.06	83.64 ± 0.13/83.64 ± 0.05
ConvAtt2_4	0.5921 M	180.81 M	8	3×3	128	80.09 ± 0.07/79.68 ± 0.12	83.68 ± 0.23/83.69 ± 0.09
ConvAtt3_1	0.0165 M	5.09 M	8	1×1	64	79.90 ± 0.11/80.13 ± 0.06	82.97 ± 0.11/84.27 ± 0.03
ConvAtt3_2	0.4128 M	126.03 M	8	5×5	64	79.53 ± 0.05/80.33 ± 0.05	83.61 ± 0.02/84.20 ± 0.17

Google Street View, consists of 647 word images in its test set. Many images in this dataset either are severely corrupted with noise and blur or have a very low resolution. **ICDAR 2003** (IC03 in short) (Lucas et al., 2003) contains 251 scene images, labeled with text bounding boxes. For fair comparison, we discarded images that contain non-alphanumeric characters or have less than three characters, following (Wang et al., 2011). The resulting dataset contains 867 cropped images.

Experiments: We test our method on all three benchmarks (IIIT5k, SVT and IC03), as shown in Table 2.

We first compare the performance of our network against the state of the art. With an encoder that has 7 convolutional layers, ConvAttention (ConvAtt3_2) achieves better performance than existing methods except for Cheng et al.'s work (Cheng et al., 2017) on all benchmarks. Two critical factors make it possible for our network to achieve good performance: a) using extra geometric annotations (location of each character) to help train the attention decoder, and b) exploiting a ResNet-based encoder for obtaining robust feature representations. However, annotating the location of each character is extremely expensive, therefore, it is not a practical solution for real applications. In the ResNet-based encoder (feature extractor), we discard the 3-rd and 4-th pooling layers used in (Cheng et al., 2017), and then integrate it with *ConvAtt3_1* to form a deeper network denoted as *ConvAtt3_1-ResNet*. We find that *ConvAtt3_1-ResNet* achieves a better performance than all existing methods.

Table 2: Comparison of accuracy among state-of-the-art methods on the IIIT5k, SVT and IC03 datasets.

Method	IIIT5k	SVT	IC03
Bissacco (Bissacco et al., 2013)	–	78.0	–
Jaderberg (Jaderberg et al., 2016)	–	80.7	93.1
Jaderberg (Jaderberg et al., 2015)	–	71.7	89.6
Shi (Shi et al., 2016a)	78.2	80.8	89.4
Shi (Shi et al., 2016b)	81.9	81.9	90.1
Lee (Lee and Osindero, 2016)	78.4	80.7	88.7
Cheng (Cheng et al., 2017)	87.4	85.9	94.2
Wang (Wang and Hu, 2017)	79.2	81.5	91.2
FC-Att	82.9	77.7	89.9
ConvAtt1_3	83.7	80.1	90.4
ConvAtt3_2	84.2	81.5	91.7
FC-Att-ResNet (Cheng et al., 2017)	83.7	82.2	91.5
ConvAtt3_1-ResNet	87.6	86.2	94.3

5.3 Challenge on Chinese Character Recognition

Chinese characters contain strokes often in a sophisticated spatial layout. Such a nature indicates complex spatial information. We challenge our method with Chinese character recognition, and believe that our proposed convolutional attention mechanism can help in this task. According to the character generation method proposed in (Gupta et al., 2016), we generate 4 million Chinese text images for training and another 20k Chinese text images as the validation set (*valid-set*). Each image has 1-10 Chinese characters chosen from the set of 3755 most commonly used ones. We also use *RCTW-17* (Shi et al., 2017), which has a training set containing 8034 Chinese scene images labeled with text bounding boxes. We discard

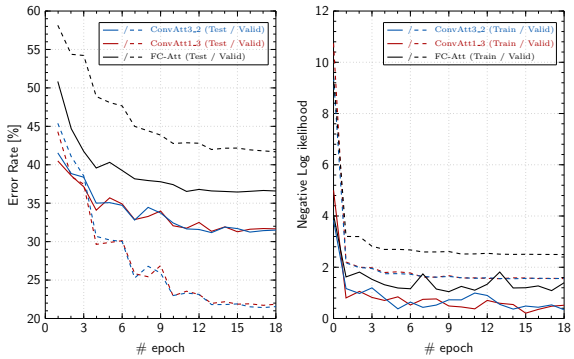

Figure 4: Performance comparison between FC-Attention and ConvAttention. (a) error rates on valid-set (dashed) and test set (solid); (b) loss curves for the training (solid) and validation (dashed) sets.

words that contain non-Chinese characters, have more than 10 characters or have a vertical style, and finally obtain 21781 cropped images as our test set.

Figure 4 gives a performance comparison between FC-Attention and ConvAttention. In subfigure (a), both ConvAtt1_3 and ConvAtt3_2 achieve lower error rates than FC-Attention on *valid-set* and *test set*. For further exploring the effectiveness of ConvAttention, we depict the training process in subfigure (b), and find that ConvAttention has a stronger fitting capacity than FC-Attention. We also note that the error rate of FC-Attention on valid-set is lower than that on the test set because the synthetic dataset has higher complexity than *RCTW-17*. The results here demonstrate that ConvAttention can take on more challenging character recognition tasks than FC-Attention.

5.4 Discussion: Influence of Sliding Window

In most existing literature (Shi et al., 2016a; Shi et al., 2016b; Cheng et al., 2017), $\frac{1}{4}$ down-sampling is used while we perform $\frac{1}{8}$ down-sampling with respect to the width of the input image in both ConvAttention and FC-Attention to lower computational cost, which may have resulted in suboptimal accuracy in Table 2. Therefore, we change the step size of the sliding window from 2 to 1, which increases the length of the resulting spatiotemporal sequence from 29 to 57; we find that the accuracy of *ConvAtt1_3* on average can be further improved by 0.97%. For fair comparison, we also change the stride of the 4-th pooling layer in FC-Attention from 2 to 1, which changes the length of the temporal sequence from 33 to 65; we find that the accuracy of *FC-Att* on average can be further improved by 0.65%. Therefore, ConvAttention outperforms FC-Attention regardless of the down-sampling strategy used.

6 CONCLUSIONS

In this paper, we have presented a novel spatiotemporal deep learning framework with a convolutional attention mechanism (ConvAttention) for retaining more information about spatial structures. ConvAttention not only preserves the advantages of FC-Attention but is also suitable for spatiotemporal data due to its inherent convolutional structure. We have successfully applied ConvAttention to the challenging problem of scene text recognition. By incorporating ConvAttention into text reading, we build an end-to-end trainable deep network for character recognition. Extensive experiments on public benchmarks demonstrate that our method achieves state-of-the-art results. As future work, we will investigate how to apply ConvAttention to image/video captioning.

REFERENCES

Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural Machine Translation by Jointly Learning to Align and Translate. In *ICLR*.

Bissacco, A., Cummins, M., Netzer, Y., and Neven, H. (2013). PhotoOCR: Reading Text in Uncontrolled Conditions. In *ICCV*, pages 785–792.

Cheng, Z., Bai, F., Xu, Y., Zheng, G., Pu, S., and Zhou, S. (2017). Focusing Attention: Towards Accurate Text Recognition in Natural Images. In *ICCV*, pages 5076–5084.

Graves, A., Fernández, S., Gomez, F., and Schmidhuber, J. (2006). Connectionist Temporal Classification : Labelling Unsegmented Sequence Data with Recurrent Neural Networks. In *ICML*, pages 369–376. ACM.

Gupta, A., Vedaldi, A., and Zisserman, A. (2016). Synthetic Data for Text Localisation in Natural Images. In *CVPR*, pages 2315–2324.

He, P., Huang, W., Qiao, Y., Loy, C. C., and Tang, X. (2016). Reading Scene Text in Deep Convolutional Sequences. In *AAAI*, pages 3501–3508.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Jaderberg, M., Simonyan, K., Vedaldi, A., and Zisserman, A. (2014). Synthetic Data and Artificial Neural Networks for Natural Scene Text Recognition. *arXiv preprint arXiv:1406.2227*.

Jaderberg, M., Simonyan, K., Vedaldi, A., and Zisserman, A. (2015). Deep Structured Output Learning for Unconstrained Text Recognition. In *ICLR*.

Jaderberg, M., Simonyan, K., Vedaldi, A., and Zisserman, A. (2016). Reading Text in the Wild with Convolutional Neural Networks. *IJCV*, 116(1):1–20.

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., and Darrell, T. (2014). Caffe: Convolutional Architecture for Fast Feature Embedding. In *ACM-MM*, pages 675–678.

Lee, C. Y. and Osindero, S. (2016). Recursive Recurrent Nets with Attention Modeling for OCR in the Wild. In *CVPR*, pages 2231–2239.

Li, H., Wang, P., and Shen, C. (2017). Towards End-To-End Text Spotting With Convolutional Recurrent Neural Networks. In *ICCV*, pages 5238–5246.

Lucas, S. M., Panaretos, A., Sosa, L., Tang, A., Wong, S., and Young, R. (2003). ICDAR 2003 robust reading competitions. In *ICDAR*, pages 682–687.

Mishra, A., Alahari, K., and Jawahar, C. V. (2012). Scene Text Recognition using Higher Order Language Priors. In *BMVC*, pages 1–11.

Neumann, L. and Matas, J. (2012). Real-time scene text localization and recognition. In *CVPR*, pages 3538–3545.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1988). Learning representations by back-propagating errors. *Cognitive Modeling*, 5(3):1.

Sainath, T. N., Mohamed, A.-r., Kingsbury, B., and Ramabhadran, B. (2013). Deep convolutional neural networks for LVCSR. In *ICASSP*, pages 8614–8618. IEEE.

Shi, B., Bai, X., and Yao, C. (2016a). An End-to-End Trainable Neural Network for Image-based Sequence Recognition and Its Application to Scene Text Recognition. *IEEE TPAMI*, preprint.

Shi, B., Wang, X., Lyu, P., Yao, C., and Bai, X. (2016b). Robust Scene Text Recognition with Automatic Rectification. In *CVPR*, pages 4168–4176.

Shi, B., Yao, C., Liao, M., Yang, M., Xu, P., Cui, L., Belongie, S. J., Lu, S., and Bai, X. (2017). ICDAR2017 competition on reading chinese text in the wild (RCTW-17). *CoRR*, abs/1708.09585.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to Sequence Learning with Neural Networks. In *NIPS*, pages 3104–3112.

Wang, J. and Hu, X. (2017). Gated Recurrent Convolution Neural Network for OCR. In *NIPS*, pages 334–343.

Wang, K., Babenko, B., and Belongie, S. (2011). End-to-end scene text recognition. In *ICCV*, pages 1457–1464.

Wang, K. and Belongie, S. (2010). Word Spotting in the Wild. In *ECCV*, pages 591–604. Springer.

Wang, T., Wu, D. J., Coates, A., and Ng, A. Y. (2012). End-to-end text recognition with convolutional neural networks. In *ICPR*, pages 3304–3308.

Xingjian, S., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-K., and Woo, W.-c. (2015). Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *NIPS*, pages 802–810.

Yang, X., He, D., Zhou, Z., Kifer, D., and Giles, C. L. (2017). Learning to Read Irregular Text with Attention Mechanisms. In *IJCAI*, pages 3280–3286.

Yao, C., Bai, X., Shi, B., and Liu, W. (2014). Strokelets: A Learned Multi-scale Representation for Scene Text Recognition. In *CVPR*, pages 4042–4049.