Scene Text Recognition by Attention Network with Gated Embedding

Cong Wang [1,2], Cheng-Lin Liu [1,2,3]
[1]*National Laboratory of Pattern Recognition, Institute of Automation of Chinese Academy of Sciences,* Beijing 100190, China
[2]*School of Artificial Intelligence, University of Chinese Academy of Sciences,* Beijing 100049, China
[3]*CAS Center for Excellence of Brain Science and Intelligence Technology,* Beijing 100190, China
{cong.wang, liucl}@nlpr.ia.ac.cn

***Abstract*—Recurrent attention based encoder-decoder model is one of the most popular frameworks for scene text recognition. However, most methods in this category only use standard recurrent attention network as the decoder. In this paper, in order to alleviate the problem that standard attention network relies on the previous output character overmuch, we propose an attention network with gated embedding for scene text recognition. The proposed attention network with gated embedding (GEAN) adopts a gated embedding to adaptively reset the input information from the embedding vector of previous output character for recurrent attention network. The gated embedding is constructed by adding an adaptive embedding gate based on the degree of correlation between the hidden state vector and the embedding vector of the corresponding character at the same time step. We verify the effectiveness of GEAN for scene text recognition through extensive experiments on both regular and irregular scene text datasets. The performance of GEAN is shown to be superior to the standard recurrent attention based decoder and is comparable compared with state-of-the-art methods.**

***Index Terms*—Scene text recognition, Attention network, Gated embedding**

I. Introduction

Texts in natural images convey rich high-level semantic information. Reading texts in images plays an important role in numerous real-world applications such as scene understanding, image and video retrieval, and driver assistance. Consequently, scene text recognition has drawn much attention from computer vision and document analysis communities. Despite several decades of research on Optical Character Recognition (OCR), recognizing texts from natural scene images is still a challenging task. The main challenges stem from the following factors. First, scene text has high variation in character color, font, size and languages. Second, most scene images undergo uneven illumination, blurring, perspective distortion, low contrast, low resolution and occlusion, etc. Moreover, text lines in the wild may have irregular shape, such as curved shape.

In recent years, benefiting from the development of deep learning, notable advances in scene text recognition have been achieved. Recent works model scene text recognition as a sequence recognition problem, which allows lexicon-free recognition and have yielded better performance. Particularly, Connectional Temporal Classification (CTC) [7] or attention mechanism [1] are widely adopted at the decoder stage in these methods.

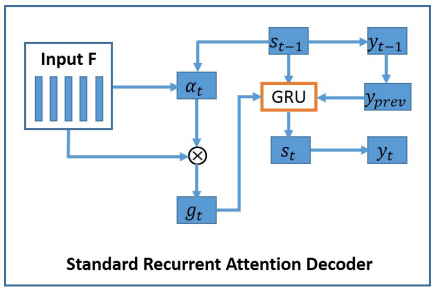


Fig. 1. The flowchart of standard recurrent attention network.

Most of recurrent attention based methods for scene text recognition [4], [5], [15], [18], [22], [27], [28], [35], [40], [41] adopt standard recurrent attention as the decoder module. Fig. 1 shows the flowchart of the adopted standard recurrent attention network, which is described in detail in Section III-B. Standard recurrent attention network recurrently outputs the prediction sequence and learns the alignment between the input sequence and the output sequence. In addition, some recent works [2], [3], [16], [31] further exploit the potential of recurrent attention model by modifying standard recurrent attention network in different perspectives. In the context of scene text recognition, Bai et al. [2] proposed an edit probability (EP), which estimates the probability of generating a string from the output sequence and needs an extra dictionary. Wang et al. [31] proposed a memory-augmented attention model, which feeded the part of character sequence already generated and all attended alignment history to the attention model when predicting the character at current time step. Li et al. [16] proposed a tailored 2D attention mechanism which considered the neighborhood information of each position for alignment module. Chen et al. [3] proposed an adaptive embedding gate to control the information transmission between adjacent characters by introducing high-order character language model and using a specific dictionary or a specific root table in a supervised way.

Existing recurrent attention based methods for scene text recognition all feed the hidden state vector at last time step to alignment module and use the hidden state vector at current time step to predict character label. Thus, the hidden state vector plays an important role in recurrent attention network.

The hidden state vector is obtained through a recurrent neural network (RNN) cell in a recurrent way. It can significantly improve the performance of recurrent attention based text recognition to feed the embedding vector of previous character to the input information of the RNN cell. The embedding vector of previous character is from the ground truth character label in the training phase, while in the test phase, it is from the predicted character label. Due to the gap between the training phase and the test phase, it is not appropriate to rely on the embedding vector of previous output character overmuch for recurrent attention network. To alleviate this problem and further exploit the potential of recurrent attention network, we propose a recurrent attention network with gated embedding (GEAN), which adaptively controls how much recurrent attention network relies on the previous output character. The proposed gated embedding is constructed by adding an adaptive embedding gate to recurrent attention network based on the degree of correlation between the hidden state vector and the embedding vector of the corresponding character label at the same time step. Compared with the standard recurrent attention based scene text recognition, the proposed attention network with gated embedding adaptively resets the input information from the previous output character and further improve the performance of scene text recognition. In the training stage, the objective function can be optimized end-to-end in a weak-supervised way, which only requires images and the corresponding text labels.

We conducted extensive experiments to verify the effectiveness of GEAN for scene text recognition. The performance of GEAN is shown to be superior to standard recurrent attention based decoder and is comparable compared with state-of-the-art methods on both regular datasets and irregular datasets, including the IIIT5K, SVT, ICDAR2003, ICDAR2013, ICDAR2015, SVT-Perspective, and CUTE80 datasets.

The rest of the paper is organized as follows. Section II reviews related work. Section III gives the details of the proposed method. Experimental results are given in Section IV, and the conclusions are presented in Section V.

II. Related Work

Numerous works for scene text recognition have been published in recent years. Comprehensive surveys can be found in [37], [42].

Early works usually follow the bottom-up pipeline: candidate characters detected in text detection stage or generated in an over-segmentation stage are classified by a character classifier, and the classification results are fused, possibly with contexts, to infer the character label sequence.

Recent works mainly follow top-down pipeline, where the entire text from the original image is directly recognized without detecting and recognizing individual characters. Currently, the popular works for regular text recognition can be roughly categorized into CTC based methods and attention based methods according to the decoding mechanism.

A. CTC based scene text recognition

CTC based methods for scene text recognition are briefly outlined as follows. Sun and Lu [29] extract sequences of HOG features to represent images, and then combine a RNN with CTC to predict the corresponding character sequence. He et al. [10] and Shi et al. [26] propose an end-to-end neural network that combines convolutional neural network (CNN) and RNN for visual feature representation, and then adopt CTC as the decoder. Yin et al. [38] propose a sliding convolutional character model in which the character classifier outputs on the sliding windows are normalized and decoded with CTC based algorithm. Gao et al. [6] incorporate the residual attention modules into a small densely connected network to encode the input text image and then adopt CTC as the decoder to generate label sequence. Liu et al. [20] design a multi-task network with an encoder-discriminator-generator architecture to guide the feature of the original image toward that of the corresponding clean image, and then adopt CTC as the decoder to output the predicted character sequence.

B. Attention based scene text recognition

By means of RNN with attention mechanism, recurrent attention model as a decoder module is widely adopted for regular and irregular text recognition. Most of recurrent attention based methods for scene text recognition [4], [5], [15], [18], [22], [27], [28], [35], [40], [41] adopt the standard recurrent attention as the decoder module.

Recent attention based methods [15], [41] have achieved substantial performance improvement for regular text recognition. Lee and Osindero [15] use a recursive CNN for image feature extraction and then adopt the attention-based decoder for sequence generation. Zhang et al. [41] propose a sequence-to-sequence domain adaptation network for robust text image recognition, exploiting unsupervised sequence data by an attention-based sequence encoder-decoder network.

Other recent attention based methods [4], [5], [16], [22], [27], [28], [35], [40] aim to recognize irregular text. Shi et al. [27], [28] introduce an end-to-end neural network model that comprises a rectification network and a recognition network. The rectification network predicts a flexible Thin-Plate Spline transformation. The recognition network is an attentional sequence-to-sequence model that predicts a character sequence directly from the rectified image. Zhan et al. [40] present an end-to-end trainable scene text recognition system that iteratively removes perspective distortion and text line curvature as driven by better scene text recognition performance. Luo et al. [22] propose a multi-object rectified attention network (MORAN), which consists of a multi-object rectification network and an attention-based sequence recognition network. Cheng et al. [5] propose the arbitrary orientation network (AON), which extracts four-direction features and the character placement clues and then adopts recurrent attention model as the decoder module. Yang et al. [35] propose an auxiliary Fully Convolutional Network for dense character detection and an alignment loss to guide the training of an attention model. Cheng et al. [4] propose a focusing attention

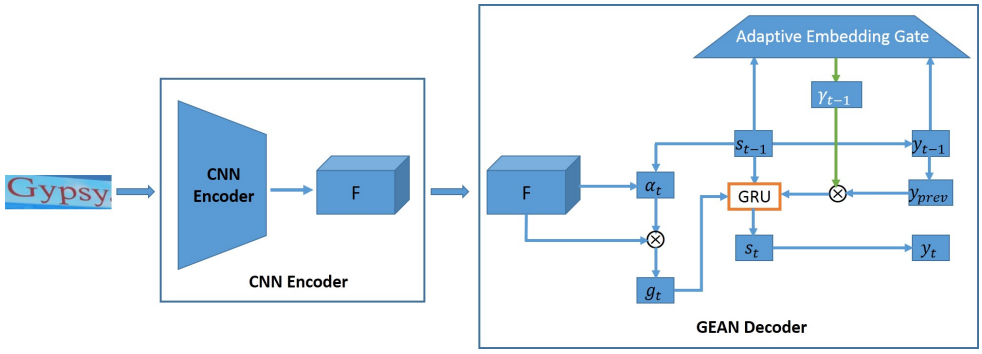


Fig. 2. The framework of the proposed GEAN for scene text recognition. It firstly encodes an input text image into two dimensional feature maps by a CNN encoder and then the two dimensional feature maps are decoded into one dimensional character sequence by the proposed attention network with gated embedding.

network (FAN) that automatically draws back the drifted attention through an auxiliary focusing network.

Some recent works [2], [3], [16], [31] further exploit the potential of recurrent attention based scene text recognition model by modifying the standard recurrent attention network in different perspectives. Bai et al. [2] propose an edit probability (EP), which estimates the probability of generating a string from the output sequence and needs an extra lexicon to be given. Wang et al. [31] propose a memory-augmented attention model, which feeds the part of character sequence already generated and all attended alignment history to the attention model when predicting the character at current time step. Li et al. [16] propose a tailored 2D attention mechanism which considers the neighborhood information of each position when computing the alignment factors. Chen et al. [3] propose an adaptive embedding gate to control the information transmission between adjacent characters by introducing high-order character language model to attentional decoder.

Besides the recurrent attention based methods for scene text recognition, the attention based approach of [33] directly connects a CNN-based 2D image encoder to a self-attention [30] based decoder.

The proposed attention network with gated embedding (GEAN) in this paper also adaptively resets the embedding vector of the previous output character for recurrent attention network. However, the proposed GEAN is motivated by the gap that the embedding vector of the previous output character is from different sources in training and test phases. Different from the method [3] which considers high-order character language model and uses a specific dictionary or a specific root table in a supervised way, the proposed GEAN constructs a gated embedding based on the degree of correlation between the hidden state vector and the embedding vector of the corresponding character label at the same time step without the need of a specific dictionary or a specific root table.

III. Proposed Method

The framework of the proposed GEAN for scene text recognition is shown in Fig. 2. It firstly encodes an input text image into two dimensional feature maps by a CNN encoder and then the two dimensional feature maps are decoded into one dimensional character sequence by the proposed attention network with gated embedding.

TABLE I
THE ARCHITECTURE OF THE CNN ENCODER. "S" DENOTES THE STRIDE OF THE FIRST CONVOLUTIONAL LAYER IN EACH BLOCK.

Layer	Configurations	Output size
Block 0	3×3, s $1 \times 1, 32$	32×100
Block 1	$\begin{bmatrix} 1 \times 1, 32 \\ 3 \times 3, 32 \end{bmatrix} \times 3$, s 2×2	16×50
Block 2	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 4$, s 2×2	8×25
Block 3	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 6$, s 2×1	4×25
Block 4	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$, s 1×1	4×25
Block 5	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$, s 1×1	4×25

A. CNN Encoder

In order to improve the performance of arbitrary-shaped text recognition, the input image is encoded into two dimensional feature maps by an adopted CNN encoder. Inspired by [28] and for fair comparison, the CNN encoder we adopt is also based on ResNet [9] and similar to the architecture of [28]. The architecture of the CNN encoder is shown in Table I. The residual unit in each residual block comprises a 1×1 convolution followed by 3×3 convolution. Feature maps are downsampled by the stride 2×2 in the first two residual blocks. The stride in the third residual block is changed to 2×1 to reserve more resolution along the horizontal axis. In order to recognize arbitrary-shaped text better, the stride in the last two residual blocks is changed to 1×1. A two dimensional feature maps are fed to the following decoder module. Considering that RNN is difficult to compute in a parallel way, we do not use the Bidirectional Long-Short Term Memory [11] to further encode the output of the CNN encoder.

B. Standard Recurrent Attention Network

The standard recurrent attention based decoder we adopt is a RNN that directly generates an output sequence $y = (y_1, \cdots, y_T)$ from an input. In this work, the input is the two dimensional feature maps F which are the output of the above described CNN encoder. the probability of the output character y_t at the t-th time step is given by

$$P(y_t|y_1, \cdots, y_{t-1}, F) = softmax(W_o s_t + b_o) \in \mathbb{R}^N, \quad (1)$$

where, W_o and b_o are the trainable parameters, and N is the number of character classes. In English text recognition, the character label space includes all 26 English letters (upper/lower case not discriminated), 10 digits, plus a special end-of-sequence (EOS) token, namely $N = 37$. When EOS is emitted, the decoder ends the generation of characters. In addition, s_t is the hidden state vector of the Gated Recurrent Unit (GRU) cell at the t-th time step. We compute s_t as

$$s_t = GRU([y_{prev}, g_t], s_{t-1}), \quad (2)$$

where y_{prev} denotes the embedding vector of the previous output y_{t-1} at the $(t-1)$-th time step. Note that y_{t-1} denotes the ground truth character label in the training phase, while in the test phase, it denotes the predicted character label. And g_t denotes the glimpse vector at the t-th time step. y_{prev} and g_t are computed as follows:

$$y_{prev} = Embedding(y_{t-1}), \quad (3)$$

$$g_t = \sum_{i=1}^{H}\sum_{j=1}^{W} \alpha_t(i,j)F(i,j), \quad (4)$$

where W and H are the width and height of the feature map F. α_t is a matrix of attention weights, also called as alignment factors, which effectively controls where the decoder focuses on at the current time step and is computed as follows:

$$e_t(i,j) = v^T tanh(W_s s_{t-1} + W_f F(i,j) + b), \quad (5)$$

$$\alpha_t(i,j) = \frac{exp(e_t(i,j))}{\sum_{k=1}^{H}\sum_{q=1}^{W} exp(e_t(k,q))}, \quad (6)$$

where v, W_s, W_f and b are the trainable parameters.

C. Recurrent Attention Network with Gated Embedding

The hidden state vector at each time step plays an important role in standard recurrent attention network for scene text recognition. Firstly, the hidden state vector at each time step is a representation of the predicted character and gives the probability of the output character through a fully connected layer. Secondly, standard recurrent attention network adopts the hidden state vector at last time step as the query to compute alignment factors at current time step. A confusing or ambiguous query can cause the inaccurate alignment factors and then make error in predicting character label.

In standard recurrent attention network, the hidden state vector at each time step is obtained through a RNN cell, which concatenates the glimpse vector at current time step and the embedding vector of the previous output character as the input information. It can significantly improve the performance of text recognition to introduce the embedding vector of the previous output character to recurrent attention network, as shown by the experimental results in Table II. However, unlike that the embedding vector of the previous output label is from the predicted character label in the test phase, it is from the ground truth character label in the training phase. Due to the gap between the training phase and the test phase, it is not appropriate to rely on the embedding vector of the previous output character overmuch for recurrent attention network.

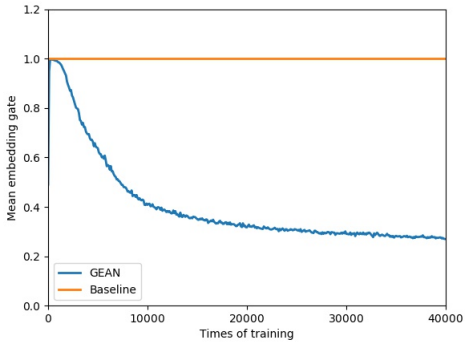


Fig. 3. The mean embedding gate of output sequences in a mini-batch during the first 40 thousands times of training.

In order to further exploit the potential of recurrent attention network and introduce the embedding vector of the previous output character to the recurrent attention network in a more elaborate way, we propose a recurrent attention network with gated embedding (GEAN), which adaptively controls how much recurrent attention network relies on the previous output label. The proposed gated embedding is constructed by adding an adaptive embedding gate to recurrent attention network based on the degree of correlation between the hidden state vector and the embedding vector of the corresponding character label at the same time step. The adaptive embedding gate at the t-th time step is defined as

$$\gamma_t = \sigma(v_g^T tanh(W_{g_1} s_{t-1} + W_{g_2} y_{prev} + b_g)), \quad (7)$$

where, v_g, W_{g_1}, W_{g_2} and b_g are the trainable parameters, σ denotes sigmoid function, and γ_t denotes the embedding gate at the t-th time step.

Thus, the hidden state vector in GEAN is computed as

$$s_t = GRU([\gamma_t * y_{prev}, g_t], s_{t-1}). \quad (8)$$

Compared with standard attention network, the proposed GEAN adaptively alleviates the problem that standard attention network relies on the previous output character overmuch. Fig. 3 demonstrates the mean embedding gate of output sequences in a mini-batch during the first 40 thousands times of training. According to Fig. 3, we can see that the proposed GEAN places too much reliance on the previous output character only in the beginning of training phase and then significantly decreases the reliance on the previous output character with the increase of training times.

D. Training Objective

For training the proposed GEAN for scene text recognition, the objective function is defined as

$$L_{reg} = -\sum_{t=1}^{T} lnP(\hat{y}_t|I,\theta), \tag{9}$$

where, I is the given input image, $\hat{y}_t$ is the ground truth of the t-th character in the character label sequence, and θ is a vector that combines all the network parameters.

IV. Experiments

We evaluate the performance of the proposed model on various benchmarks, including both regular and irregular text datasets.

A. Datasets

Following many published works, the proposed GEAN model is trained on two synthetic datasets, namely Synth90k [12] and SynthText [8]. And the proposed GEAN model is tested on four regular text datasets and three irregular text datasets, namely IIIT5K-Words [23], Street View Text [32], ICDAR 2003 [21], ICDAR 2013 [14], ICDAR 2015 Incidental Text [13], Street View Text Perspective [24] and CUTE80 [25]).

Synth90k [12] contains 8-million training images and theirs corresponding ground truth words. Such images are generated by a synthetic text engine and are highly realistic.

SynthText [8] is generated for text detection. Therefore, words are rendered onto full images. We crop 6-million word images using the groundtruth word bounding boxes.

IIIT5K-Words [23] (IIIT5K) was collected from the Internet. The test set contains 3000 cropped word images. Each word image has a 50-word lexicon and a 1000-word lexicon.

Street View Text [32] (SVT) was collected from Google Street View. The test set contains 249 images, from which 647 word images are cropped. Each word image has a 50-word lexicon.

ICDAR 2003 [21] (IC03) contains 251 scene images in its test dataset. Following [32], we discard images that contain non-alphanumeric characters or have less than three characters, and then get a test set with 859 cropped word images. Each word image is associated with a 50-word lexicon and a full lexicon which contains all label words.

ICDAR 2013 [14] (IC13) inherits most text images from IC03. The dataset is filtered by removing words that contain non-alphanumeric characters. and contains 1015 cropped text images.

ICDAR 2015 Incidental Text [13] (IC15) is from the Challenge 4 of the ICDAR 2015 Robust Reading Competition. It contains 2077 cropped text images for testing. And some of them are irregular. To fairly compare with some previous methods, we also test our method on a subset of IC15, which discards the images containing non-alphanumeric characters and has 1811 images (refer to as IC15-1811).

Street View Text Perspective [24] (SVT-P) is from side-view angle snapshots in Google Street View. The dataset consists of 645 cropped images for testing. Many of them are heavily distorted by the non-frontal view angle. Each word image is associated with a 50-word lexicon and a full lexicon which contains all label words.

CUTE80 [25] contains 80 high-resolution images taken in natural scenes. It was specifically collected for evaluating the performance of curved text recognition. It contains 288 cropped natural images for testing.

B. Implementation Details

Network: The architecture of the CNN encoder is given in Table I. The number of hidden units of GRU in the decoder module is set to 256. Specifically, "Baseline" method in the following sections refers to the standard recurrent attention based text recognition, which has no the gated embedding.

Model training: We apply ADADELTA [39] to train our proposed model. We set the learning rate to 1.0 in the beginning and decrease it to 0.1 after the third epoch. The mini-batch size of each branch is set to 64. All images in both training set and test set are resized to 32×100. And we randomly rotate the input images in a certain angle range of $[-30°, 30°]$ in the training phase.

Implementation: We implement the proposed model with Pytorch and conduct all experiments on a NVIDIA TITAN Xp GPU with 12GB memory.

Transcription: In the test phase, we obtain the final recognition result by straightforwardly selecting the most probable character at each time step for lexicon-free transcription. When a lexicon is given, the final recognition result chooses the sequence in the lexicon that has smallest edit distance with recognition result via lexicon-free transcription.

C. Effect of Embedding Vector y_{prev}

We examine how the embedding vector of the previous output character y_{prev} impacts the performance of recurrent attention based text recognition. The corresponding results are shown in Table II. We can see that the performance of recurrent attention based text recognition significantly decreases when the input information of RNN cell does not use the embedding vector y_{prev}. In addition, compared with the standard recurrent network, the proposed recurrent attention network with gated embedding can further improve the performance.

TABLE II
THE PERFORMANCE OF RECURRENT ATTENTION BASED TEXT RECOGNITION WHEN USING EMBEDDING VECTOR y_{prev} IN DIFFERENT WAYS.

Variants	IIIT5K	SVT	IC13	SVT-P	CUTE80
Baseline	92.0	**87.9**	91.4	79.7	80.9
without y_{prev}	90.4	85.5	89.8	74.6	78.8
GEAN	**92.5**	87.5	**91.9**	**80.9**	**82.6**

D. Performance on Regular Text Datasets

We evaluate the performance of the proposed GEAN compared with other representative methods on four regular text datasets. The results are shown in Table III.

TABLE III
THE RESULTS ON FOUR PUBLIC REGULAR DATASETS. "50", "1K" AND "FULL" DENOTE THE LEXICON SIZES, "NONE" MEANS NO LEXICON. "*" INDICATES THE MODELS TRAINED WITH BOTH WORD-LEVEL AND CHARACTER-LEVEL ANNOTATIONS. "†" INDICATES THE MODELS TRAINED WITH EXTRA DICTIONARY OR ROOT TABLE.

Method	IIIT5K			SVT		IC03			IC13
	50	1k	None	50	None	50	Full	None	None
Yao et al. [36]	80.2	69.3	-	75.9	-	88.5	80.3	-	-
Su and Lu [29]	-	-	-	83.0	-	92.0	82.0	-	-
Shi et al. [26]	97.8	95.0	81.2	97.5	82.7	98.7	**98.0**	91.9	89.6
Shi et al. [27]	96.2	93.8	81.9	95.5	81.9	98.3	96.2	90.1	88.6
Lee et al. [15]	96.8	94.4	78.4	96.3	80.7	97.9	97.0	88.7	90.0
Yin et al. [38]	98.9	96.7	81.6	95.1	76.5	97.7	96.4	84.5	85.2
Cheng et al. [4]*	99.3	97.5	87.4	97.1	85.9	99.2	97.3	94.2	93.3
Cheng et al. [5]	99.6	98.1	87.0	96.0	82.8	98.5	97.1	91.5	-
Bai et al. [2]†	99.5	97.9	88.3	96.6	87.5	98.7	97.9	94.6	**94.4**
Liu et al. [18]	-	-	92.0	-	85.5	-	-	-	91.1
Liu et al. [20]	97.3	96.1	89.4	96.8	87.1	98.1	97.5	94.7	94.0
Shi et al. [28]	99.6	**98.8**	93.4	97.4	89.5	**98.8**	**98.0**	94.5	91.8
Gao et al. [6]	99.1	97.9	81.8	97.4	82.7	98.7	96.7	89.2	88.0
Li et al. [16]	-	-	91.5	-	84.5	-	-	-	91.0
Liao et al. [17]*	**99.8**	**98.8**	91.9	**98.8**	86.4	-	-	-	91.5
Luo et al. [22]	97.9	96.2	91.2	96.6	88.3	98.7	97.8	**95.0**	92.4
Xie et al. [34]	-	-	82.3	-	82.6	-	-	92.1	89.7
Zhan et al. [40]	99.6	98.8	93.3	96.9	**90.2**	-	-	-	91.3
Chen et al. [3]†	99.4	98.3	**93.6**	96.9	89.2	98.8	**98.0**	94.8	92.9
Baseline	99.2	98.1	92.0	97.4	87.9	98.1	96.4	93.7	91.4
GEAN	99.3	98.2	92.5	96.6	87.5	97.8	95.9	93.6	91.9

From Table III, we can see that GEAN achieves a comparable performance on regular text datasets compared with the state-of-the-art methods . The method of [16] also used extra synthetic and public real data in training besides Synth90k and SynthText. Referring to [33], the result of [16] we compare with is only based on the model trained with Synth90k and SynthText for fair comparison. The methods of [2], [3] both train the models with extra dictionary or root table. And the performance in [3] we compare is that without combining with the rectification based methods for fair comparison.

As shown in Table III, GEAN can statistically improve the performance on regular text datasets under lexicon-free (None) condition compared with "Baseline" method. This verifies the effectiveness of the proposed GEAN for regular text recognition.

E. Performance on Irregular Text Datasets

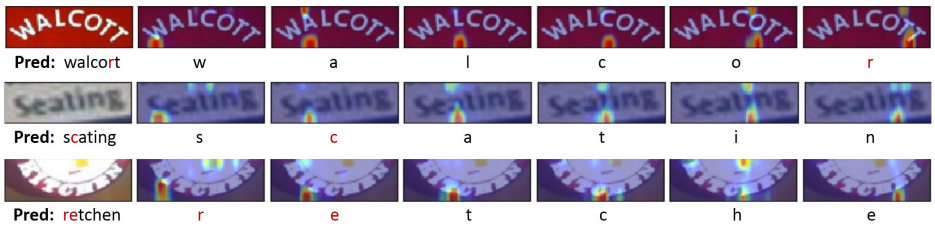
We also evaluate the performance of the proposed GEAN on three irregular text datasets compared with other representative methods. The results are shown in Table IV.

From Table IV, we can see that GEAN achieves a comparable or superior performance on irregular text datasets compared with the state-of-the-art methods. In addition, GEAN can be flexibly combined with other recurrent attention based methods to further improve the performance of scene text recognition, such as the rectification based methods [28] [40] [22].

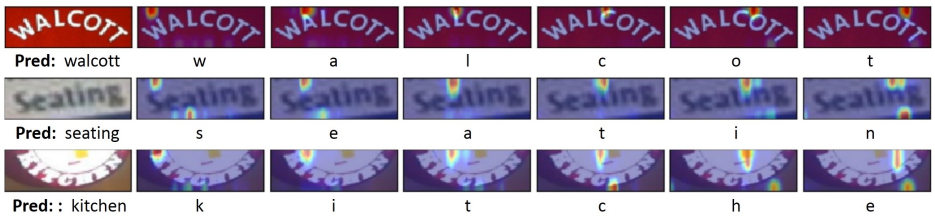
As shown in Table IV, GEAN can improve the performance on the three irregular text datasets compared with "Baseline" method. Specifically, it gives accuracy increases of 1.2% (from 79.7% to 80.9%) on SVT-P, 1.7% (from 80.9% to 82.6%) on CUTE80, 1.3% (from 72.9% to 74.2%) on IC15 and 0.8% (from 77.7% to 78.5%) on IC15-1811. The experiment verifies that the proposed GEAN is effective for irregular text recognition. Some irregular text images which are recognized by GEAN correctly but not recognized by "Baseline" correctly are shown in Fig. 4. Fig. 4(a) shows the images and the corresponding 2D attention maps at each time step based on "Baseline", while Fig. 4(b) shows that based on GEAN.

TABLE IV
THE RESULTS ON THREE PUBLIC IRREGULAR DATASETS. "NONE" MEANS NO LEXICON. "*" INDICATES THE MODELS TRAINED WITH BOTH WORD-LEVEL AND CHARACTER-LEVEL ANNOTATIONS. "†" INDICATES THE MODELS TRAINED WITH EXTRA DICTIONARY OR ROOT TABLE.

Method	SVT-P	CUTE80	IC15	IC15-1811
	None	None	None	None
Shi et al. [26]	71.8	59.2	-	-
Yang et al. [35]	75.8	69.3	-	-
Liu et al. [19]	73.5	-	-	-
Cheng et al. [4]*	71.5	63.9	-	66.2
Cheng et al. [5]	73.0	76.8	68.2	-
Bai et al. [2]†	-	-	-	73.9
Liu et al. [20]	73.9	62.5	-	-
Liu et al. [18]	78.9	-	74.2	-
Shi et al. [28]	78.5	79.5	76.1	-
Liao et al. [17]*	-	79.9	-	-
Li et al. [16]	76.4	**83.3**	69.2	-
Luo et al. [22]	76.1	77.4	68.8	-
Xie et al. [34]	70.1	82.6	68.9	-
Zhan et al. [40]	79.6	**83.3**	**76.9**	-
Chen et al. [3]†	80.0	80.2	75.5	-
Baseline	79.7	80.9	72.9	77.7
GEAN	**80.9**	82.6	74.2	**78.5**



(a) Visualization of the 2D attention maps at each time step based on "Baseline" method.



(b) Visualization of the 2D attention maps at each time step based on GEAN.

Fig. 4. Some irregular text images which are recognized by GEAN correctly but not recognized by "Baseline" method correctly. (a) shows the images and the corresponding 2D attention maps at each time step based on "Baseline" method, while (b) shows the images and the corresponding 2D attention maps at each time step based on GEAN.

V. Conclusion

In this paper, in order to alleviate the problem that standard attention network relies on the previous output character overmuch, we propose an attention network with gated embedding for scene text recognition. The proposed attention network with gated embedding (GEAN) adopts a gated embedding to adaptively reset the input information from embedding vector of previous output character. The gated embedding is constructed by adding an adaptive embedding gate based on the degree of correlation between the hidden state vector and the embedding vector of the corresponding character label at the same time step. The proposed GEAN can improve the performance of recurrent attention based scene text recognition. We verify the effectiveness of GEAN for scene text recognition through extensive experiments. The performance of GEAN is shown to be superior to the standard recurrent attention based decoder and is comparable compared with state-of-the-art methods. In addition, the proposed GEAN can be flexibly combined with other recurrent attention based models for scene text recognition.

Acknowledgments

This work has been supported by the Major Project for New Generation of AI under Grant No. 2018AAA0100400, the National Natural Science Foundation of China (NSFC) grants 61733007 and 61721004.

References

[1] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[2] F. Bai, Z. Cheng, Y. Niu, S. Pu, and S. Zhou, "Edit probability for scene text recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1508–1516.

[3] X. Chen, T. Wang, Y. Zhu, L. Jin, and C. Luo, "Adaptive embedding gate for attention-based scene text recognition," *Neurocomputing*, vol. 381, pp. 261–271, 2020.

[4] Z. Cheng, F. Bai, Y. Xu, G. Zheng, S. Pu, and S. Zhou, "Focusing attention: Towards accurate text recognition in natural images," in *Proceedings of the International Conference on Computer Vision*, 2017, pp. 5076–5084.

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

[6] Y. Gao, Y. Chen, J. Wang, M. Tang, and H. Lu, "Reading scene text with fully convolutional sequence modeling," *Neurocomputing*, vol. 339, pp. 161–170, 2019.

[7] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 369–376.

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

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2016, pp. 770–778.

[10] P. He, W. Huang, Y. Qiao, C. C. Loy, and X. Tang, "Reading scene text in deep convolutional sequences," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[11] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

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

[13] D. Karatzas, L. Gomez-Bigorda, A. Nicolaou, S. Ghosh, A. Bagdanov, M. Iwamura, J. Matas, L. Neumann, V. R. Chandrasekhar, S. Lu *et al.*, "Icdar 2015 competition on robust reading," in *Proceedings of the 13th International Conference on Document Analysis and Recognition*, 2015, pp. 1156–1160.

[14] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L. G. i Bigorda, S. R. Mestre, J. Mas, D. F. Mota, J. A. Almazan, and L. P. de las Heras, "Icdar 2013 robust reading competition," in *Proceedings of the 12th International Conference on Document Analysis and Recognition*, 2013, pp. 1484–1493.

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

[16] H. Li, P. Wang, C. Shen, and G. Zhang, "Show, attend and read: A simple and strong baseline for irregular text recognition," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 8610–8617.

[17] M. Liao, J. Zhang, Z. Wan, F. Xie, J. Liang, P. Lyu, C. Yao, and X. Bai, "Scene text recognition from two-dimensional perspective," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 8714–8721.

[18] W. Liu, C. Chen, and K.-Y. K. Wong, "Char-net: A character-aware neural network for distorted scene text recognition," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[19] W. Liu, C. Chen, K.-Y. K. Wong, Z. Su, and J. Han, "Star-net: a spatial attention residue network for scene text recognition." in *Proceedings of the British Machine Vision Conference*, vol. 2, 2016, p. 7.

[20] Y. Liu, Z. Wang, H. Jin, and I. Wassell, "Synthetically supervised feature learning for scene text recognition," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 435–451.

[21] S. M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, and R. Young, "Icdar 2003 robust reading competitions," in *Proceedings of the 7th International Conference on Document Analysis and Recognition*, 2003, pp. 682–687.

[22] C. Luo, L. Jin, and Z. Sun, "Moran: A multi-object rectified attention network for scene text recognition," *Pattern Recognition*, vol. 90, pp. 109–118, 2019.

[23] A. Mishra, K. Alahari, and C. Jawahar, "Scene text recognition using higher order language priors," in *Proceedings of the British Machine Vision Conference*, 2012.

[24] T. Quy Phan, P. Shivakumara, S. Tian, and C. Lim Tan, "Recognizing text with perspective distortion in natural scenes," in *Proceedings of the International Conference on Computer Vision*, 2013, pp. 569–576.

[25] A. Risnumawan, P. Shivakumara, C. S. Chan, and C. L. Tan, "A robust arbitrary text detection system for natural scene images," *Expert Systems with Applications*, vol. 41, no. 18, pp. 8027–8048, 2014.

[26] B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 11, pp. 2298–2304, 2017.

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

[28] B. Shi, M. Yang, X. Wang, P. Lyu, C. Yao, and X. Bai, "Aster: An attentional scene text recognizer with flexible rectification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

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

[30] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.

[31] C. Wang, F. Yin, and C.-L. Liu, "Memory-augmented attention model for scene text recognition," in *Proceedings of the 16th International Conference on Frontiers in Handwriting Recognition*, 2018, pp. 62–67.

[32] K. Wang, B. Babenko, and S. Belongie, "End-to-end scene text recognition," in *Proceedings of the International Conference on Computer Vision*, 2011, pp. 1457–1464.

[33] P. Wang, L. Yang, H. Li, Y. Deng, C. Shen, and Y. Zhang, "A simple and robust convolutional-attention network for irregular text recognition," *arXiv preprint arXiv:1904.01375*, 2019.

[34] Z. Xie, Y. Huang, Y. Zhu, L. Jin, Y. Liu, and L. Xie, "Aggregation cross-entropy for sequence recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6538–6547.

[35] X. Yang, D. He, Z. Zhou, D. Kifer, and C. L. Giles, "Learning to read irregular text with attention mechanisms." in *IJCAI*, vol. 1, no. 2, 2017, p. 3.

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

[37] Q. Ye and D. Doermann, "Text detection and recognition in imagery: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 7, pp. 1480–1500, 2014.

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

[39] M. D. Zeiler, "Adadelta: an adaptive learning rate method," *arXiv preprint arXiv:1212.5701*, 2012.

[40] F. Zhan and S. Lu, "Esir: End-to-end scene text recognition via iterative image rectification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2059–2068.

[41] Y. Zhang, S. Nie, W. Liu, X. Xu, D. Zhang, and H. T. Shen, "Sequence-to-sequence domain adaptation network for robust text image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2740–2749.

[42] Y. Zhu, C. Yao, and X. Bai, "Scene text detection and recognition: Recent advances and future trends," *Frontiers of Computer Science*, vol. 10, no. 1, pp. 19–36, 2016.