Semi-Supervised Scene Text Recognition

Yunze Gao, Yingying Chen, Jinqiao Wang, *Member, IEEE*, and Hanqing Lu, *Senior Member, IEEE*

***Abstract*— Scene text recognition has been widely researched with supervised approaches. Most existing algorithms require a large amount of labeled data and some methods even require character-level or pixel-wise supervision information. However, labeled data is expensive, unlabeled data is relatively easy to collect, especially for many languages with fewer resources. In this paper, we propose a novel semi-supervised method for scene text recognition. Specifically, we design two global metrics, i.e., edit reward and embedding reward, to evaluate the quality of generated string and adopt reinforcement learning techniques to directly optimize these rewards. The edit reward measures the distance between the ground truth label and the generated string. Besides, the image feature and string feature are embedded into a common space and the embedding reward is defined by the similarity between the input image and generated string. It is natural that the generated string should be the nearest with the image it is generated from. Therefore, the embedding reward can be obtained without any ground truth information. In this way, we can effectively exploit a large number of unlabeled images to improve the recognition performance without any additional laborious annotations. Extensive experimental evaluations on the five challenging benchmarks, the Street View Text, IIIT5K, and ICDAR datasets demonstrate the effectiveness of the proposed approach, and our method significantly reduces annotation effort while maintaining competitive recognition performance.**

***Index Terms*— Semi-supervised scene text recognition, embedding, reinforcement learning.**

I. Introduction

RECOGNIZING text in the natural images has attracted increasing interests in computer vision. Text is ubiquitous in our daily life. There exists text in the road sign, billboard, poster and license plate, and so on. As an important element of scene understanding, scene text carries rich and precise semantic information. Therefore, scene text has vital significance for numerous practical applications, including robot navigation, road sign recognition, product search and virtual reality. Although much effort has been made towards scene text recognition [1]- [6], it is still challenging to read text in the wild, due to the large variations including text fonts, irregular shape, image qualities and background noise.

Deep learning methods for scene text recognition have yielded impressive results in recent years. In particular, the sequence learning based methods have been advancing rapidly. Current approaches [1], [3]–[5] typically follow an encoder-decoder framework to recognize text in the natural image. They usually combine the convolutional neural network (CNN) and recurrent neural network (RNN) to encode the input image, then utilize RNN with attention mechanism [7] or connectionist temporal classification (CTC) [8] to decode the encoded information into the target string. At present, the attention-based encoder-decoder paradigm has become the core of most state-of-the-art scene text recognition methods. Cheng *et al.* [9] employed a focusing attention mechanism to alleviate the drifted attention and improve the recognition performance. More recently in [13], aiming at the misalignment problem between the ground truth strings and the attention's output probability distribution sequences caused by missing or superfluous characters, Bai *et al.* proposed the edit probability to estimate the probability of generating a string that might contain missing or superfluous characters. Although shown promising results on standard benchmarks, these approaches were generally trained end-to-end to learn the mapping between input images and target strings in a purely data-driven manner. Most existing methods are starved of a large amount of labeled images during training process. Especially, Cheng *et al.* [9] and Liu *et al.* [14] needed extra pixel-wise annotation information. Acquiring such data requires expensive and laborious annotation efforts and costs. By contrast, unlabeled data is far easier to collect, especially for many low resourced languages.

In this paper, we propose a novel semi-supervised architecture for scene text recognition, which requires only a small number of labeled images and has the ability to exploit a large number of unlabeled images. Moreover, we only supply the textual labels for labeled data. Specifically, we design two global word-level metrics and directly optimize these metrics using reinforcement learning. First, we propose an edit reward for the labeled data, which can measure the distance between the ground truth label and the generated string. Then we design an embedding reward defined by visual-semantic embedding [10]–[12] without the requirement for ground truth. Through embedding word image and word string into a common space, the embedding reward can evaluate the correctness of the generated string according to the similarity between input image and target string. Naturally, the predicted string should be close to the corresponding input image. Therefore, we can obtain the embedding reward easily for all the data, including the unlabeled data, without the need for the time-consuming annotation effort. With the embedding reward, we can exploit a large amount of unlabeled data to improve the performance.

Manuscript received August 28, 2018; revised October 14, 2019 and December 22, 2020; accepted December 31, 2020. Date of publication January 20, 2021; date of current version February 19, 2021. This work was supported in part by the Research and Development Projects in the Key Areas of Guangdong Province under Grant 2019B010153001 and in part by the National Natural Science Foundation of China under Grant 61772527, Grant 61806200, Grant 62006230, and Grant 61876086. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Guo-Jun Qi. *(Yunze Gao and Yingying Chen contributed equally to this work.) (Corresponding author: Yingying Chen.)*

The authors are with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: yunze.gao@nlpr.ia.ac.cn; yingying.chen@nlpr.ia.ac.cn; jqwang@nlpr.ia.ac.cn; luhq@nlpr.ia.ac.cn).

Digital Object Identifier 10.1109/TIP.2021.3051485

In addition, traditional methods are optimized by the character-level cross-entropy loss, which aims at maximizing the probability of each ground truth character. However, the missing or redundant characters could cause that the generated string is close to the ground truth but the large error is back propagated [13]. By contrast, the proposed word-level rewards can serve as the reasonable global metrics to optimize for scene text recognition. Thus, we design a hybrid loss function to mix the cross-entropy loss and reward-based losses, which benefits from the combination of the character-level and word-level optimization methods. Considering the computation of edit reward and embedding reward is based on the generated string, we need to sample the characters from the probability distributions, which is non-differentiable. Hence, we adopt reinforcement learning to measure the reward-based losses and compute the gradients. Moreover, typical attention-based encoder-decoder trained with cross-entropy loss often exhibits exposure bias. During training, the model predicts the next character given the previous ground truth character at each step. However, in the process of testing, the previously predicted character is provided to predict the next character. Therefore, the model is trained based on the training data distribution, but is tested based on the model distribution. The disagreement between training and testing causes the error accumulation at test time. Recently, Ranzato *et al*. [15] have shown that the exposure bias issue can be addressed by incorporating reinforcement learning. Therefore, our approach not only has the ability to utilize the unlabeled data, but also overcomes the exposure bias.

The main contributions are summarized as follows:

(1) A novel semi-supervised approach is proposed for scene text recognition, which can exploit a large amount of unlabeled data to improve recognition performance.

(2) Two global metrics are designed as the optimization objectives for reinforcement learning. The edit reward can measure the agreement between the ground truth and generated string, while the embedding reward can evaluate the similarity between the input image and generated string.

(3) The training with reinforcement learning can overcome the exposure bias for the attention-based encoder-decoder framework.

(4) Extensive experiments on the challenging benchmarks show that the proposed approach can significantly reduce annotation effort while maintaining competitive recognition performance.

II. Related Work

Many scene text recognition approaches have been proposed in the literature. Comprehensive surveys can be found in [16], [17]. Traditional methods [18]–[20] tackled this problem using bottom-up paradigm, which first detected individual characters by sliding window, connected components or Hough voting, then used heuristic rules or language model to combine recognized characters or character hypotheses into words. These methods relied on explicit character detection and were sensitive to the detection performance. However, in many cases character detection is not reliable and will degrade the overall performance.

Instead of the bottom-up approaches, numerous previous works have attempted to recognize words without explicitly detecting characters, but learn the mapping between the global image representation and the target string directly. Jaderberg *et al*. [21] used two CNNs to classify the characters at each position in the word and detect the N-grams contained within the word separately. Jaderberg *et al*. [22] assigned a class label to each word and conducted a 90k-class classification with CNN. Recent studies formulated the scene text recognition as a sequence recognition problem. Shi *et al*. [1] and He *et al*. [3] combined CNN and RNN to extract image feature and applied CTC loss for model learning. As another powerful paradigm, attention-based encoder-decoder is proposed by [4], [5], which utilized an attention-based decoder to perform feature selection and generate characters one by one. Furthermore, Cheng *et al*. [9] employed a focusing attention mechanism to address the attention drift. This method improved the performance greatly, however, it required additional pixel-wise labeled data. In addition, most existing approaches, especially the neural network based methods, require a large number of labeled images in the training process. But labeled data is expensive to provide, especially the character-level and pixel-wise labeled data.

Different from the existing methods, in this paper a novel semi-supervised approach is proposed for scene text recognition. We are able to utilize a large amount of unlabeled data to improve the recognition performance, which saves the labor-intensive and time-consuming annotation costs. In addition, the word-level rewards can effectively complement with the traditional character-level cross-entropy to better measure the prediction quality. To the best of our knowledge, this is the first work to perform semi-supervised scene text recognition.

III. The Proposed Approach

The overview of our semi-supervised architecture for scene text recognition is shown in Figure 1. First, we adopt an attention network [4] to generate the target string for input image. For the attention network, we use the basic attention-based encoder-decoder pipeline. A convolutional-recurrent encoder extracts the visual representation of input image, and then a RNN with attention mechanism decodes the feature representation and generates the characters recurrently. After generating the word string, the edit reward can be measured according to the edit distance between ground truth and generated string. Through the embedding network, the embedding reward can be obtained conditioned on the similarity between input image and generated string. The embedding network is the key component of our semi-supervised architecture, which is able to compute the embedding reward without the requirement of any labels. In this way, the attention network can be further improved by optimizing the edit reward and the embedding reward with reinforcement learning.

A. Attention Network

The attention network is composed of an image encoder and an attention decoder, which aims at generating the target

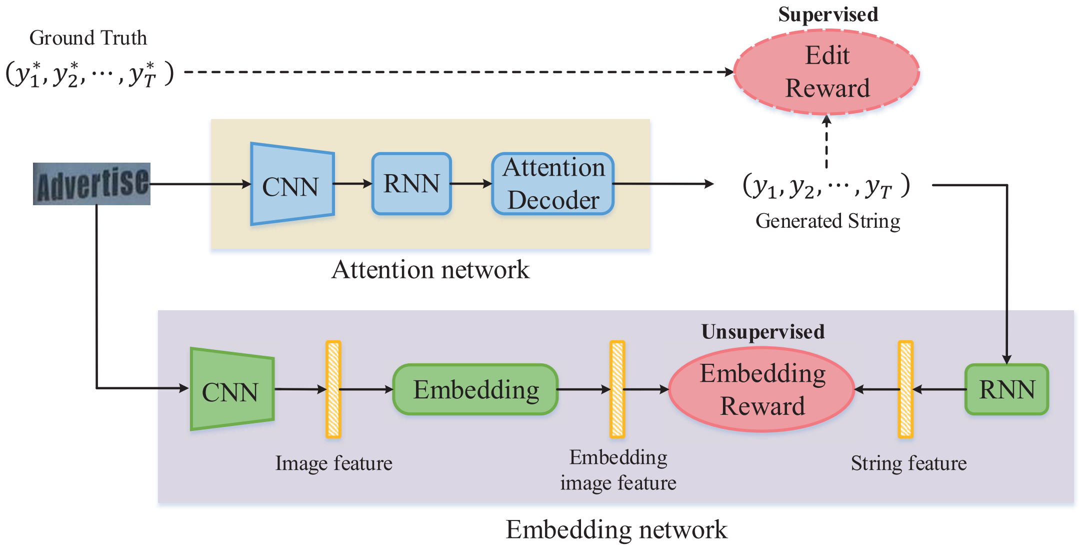

Fig. 1. Overview of our semi-supervised architecture for scene text recognition. The attention network generates the word string conditioned on the input image. The edit reward is obtained according to the distance between the ground truth and the generated string. Through embedding network, we can get the embedding reward, which measures the similarity between the input image and the generated string. Dashed connections are only for the labeled data.

strings for the input images. The encoder E combines CNN and RNN to extract a sequence of feature vectors $h = E(I)$ from the input image I. The decoder D recurrently generates a sequence of characters $y = (y_1, y_2, \ldots, y_T)$ given the encoded sequential representation h. The beginning of each sequence is marked with a special begin-of-sequence (BOS) token, and the end with an end-of-sequence (EOS) token. At each step t, the decoder dynamically weights the image feature and focuses on the most relevant content to generate the output probability distribution p_t:

$$p_t = softmax(V^T s_t), \tag{1}$$

$$s_t = RNN(y_{t-1}, s_{t-1}, g_t), \tag{2}$$

$$g_t = \sum_{j=1}^{|h|} \alpha_{t,j} h_j, \tag{3}$$

$$\alpha_{t,j} = \frac{\exp(e_{t,j})}{\sum_{j=1}^{|h|} \exp(e_{t,j})} \tag{4}$$

$$e_{t,j} = v^T tanh(W s_{t-1} + U h_j + b) \tag{5}$$

in which s_t, g_t, $\alpha_{t,j}$ and $e_{t,j}$ represent the RNN hidden state, the weighted sum of sequential feature vectors h, the attention weight and the alignment score, respectively. And W, U, V, v, b are the parameters to be learned.

Traditionally, the model parameters are learned by maximizing the likelihood of each ground truth character. At step t, the decoder outputs the probability distribution conditioned on the encoded feature representation and the previous ground truth character. Given the ground truth string $y^* = (y_1^*, y_2^*, \ldots, y_T^*)$, the objective is to minimize the cross-entropy loss:

$$L(\theta) = -\sum_{t=1}^{T} \log p(y_t^* | h, y_{t-1}^*), \tag{6}$$

where θ denotes the parameters of the attention network. During the phase of testing, the ground truth is unavailable, hence the previous predicted character is provided instead. At each step, the character with the highest probability is chosen.

B. Edit Reward

The traditional cross-entropy loss is optimized in the character-level style. However, sometimes the generated string only has some missing or redundant characters. In this case, the generated result is close to the ground truth but the misalignment causes the large error, which may confuse the training process. Therefore, we propose the word-level edit reward for the labeled data to further improve the optimization process.

The edit reward measures the distance between the ground truth string $y^* = (y_1^*, y_2^*, \ldots, y_T^*)$ and the generated string $y = (y_1, y_2, \ldots, y_T)$. We define the edit reward of the generated string y to be the negative normalized edit distance [30] between y and y^*:

$$r_{edit} = -\frac{edit\_distance(y, y^*)}{|y^*|}. \tag{7}$$

The closer the generated string is to the ground truth, the smaller the edit distance and the higher the reward. Through optimizing the edit reward, we can improve the correctness of the generated results effectively.

C. Embedding Reward

The embedding reward can be easily obtained through the embedding network, relying only on the correlation between the input image and the generated string, without the requirement of any ground truth information. This component is of crucial importance for the semi-supervised training. We define the embedding reward by the similarity between the generated string y and the input image I. As illustrated in Figure 2, we first use a CNN and a RNN to encode word images and word strings into feature vectors, respectively. But the feature

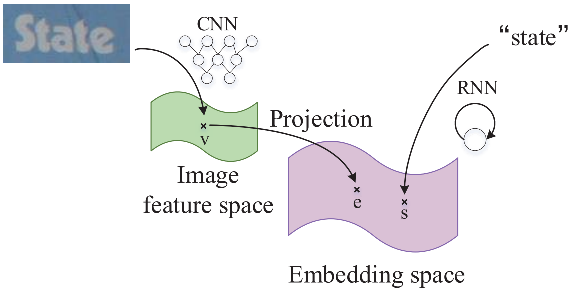

Fig. 2. Illustration of the embedding process. First, a CNN is used to extract the holistic feature of the word image. Then an RNN is adopted to encode feature of the word string, where the characters in the string serve as the input of the RNN one by one and the last hidden state of the RNN acts as the feature representation. After that, we project the image feature into the semantic space of the string feature through several linear layers to make them comparable.

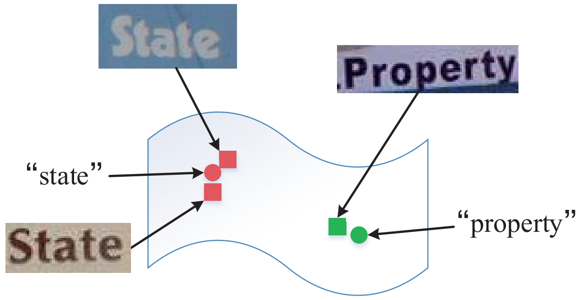

Fig. 3. Illustration of the matching image-string pairs and non-match image-string pairs in the embedding space. The images are close to their corresponding word strings with larger similarities, while far from the non-corresponding word strings with lower similarities. Squares and circles represent images and strings, respectively. Different colors distinguish between different image-string pairs. For example, the red items are the image and string corresponding to "state", while the green items correspond to "property."

representations have different dimensionality and semantics. In order to make them comparable, we project the image features into the space of string features by the linear mapping layers. Through learning to map the word images and word strings into a common embedding space, the similarity can be measured naturally by the inner product of the normalized feature representations.

The embedding process is implemented by the embedding network, which consists of a CNN image encoder E_i, a RNN string encoder E_s and linear mapping layers F_e. Given the word string y, the feature is represented by the last hidden state s of the string encoder. The image encoder extracts the visual representation v of the input image, and then the feature v is projected to the embedding space by the linear mapping layers. We use e to denote the embedded image feature. The embedding network is trained using the labeled images. As shown in Figure 3, the embedding network is learned using the ranking loss [23] by enforcing the matching image-string pairs to be closer than the non-matching pairs by $margin$:

$$L(\theta_e) = \sum_i \sum_{j \neq i} max(0, margin - e_i \cdot s_i + e_i \cdot s_j) + \sum_i \sum_{j \neq i} max(0, margin - s_i \cdot e_i + s_i \cdot e_j), \quad (8)$$

where θ_e represents the parameters of the embedding network, (e_i, s_i) is the corresponding image-string pair, e_j is the embedding of other image term, and s_j is the embedding of other string term. In such a way, we can learn the mapping to force the similarity of the positive pair is higher than those of negative pairs in the common embedding space. After training, the embedding network retains unchanged and is able to compute the similarity of arbitrary image-string pair.

The generated string naturally corresponds to the given input image. Hence the embedding reward is obtained in the unsupervised setting. More closer to the input image, the generated string is more accurate and the embedding reward is higher. We define the embedding reward by the cosine similarity between the generated string with feature s and the input image with feature e:

$$r_{embedding} = \frac{e \cdot s}{||e|| \cdot ||s||}. \quad (9)$$

With the embedding reward, the generated string is encouraged to be closer to the corresponding image than other distractor images.

D. Reinforcement Learning

Since the computation of edit reward and embedding reward is based on the generated string, we need to sample character from the probability distribution, which is non-differentiable. Moreover, the model trained with cross-entropy loss suffers from exposure bias. Because the attention decoder has knowledge of the previous ground truth character during training but uses the previously generated character from the model distribution when testing. Therefore, the model is only exposed to the training data distribution, rather than its own predictions. The disagreement will result in the error accumulation and reduce the performance. Both problems can be overcome by reinforcement learning [24]. Therefore, we adopt the reinforcement learning to optimize the designed rewards directly.

In order to maximize the reward, there is an agent to interact with the given environment and perform actions in reinforcement learning. To formulate scene text recognition as a reinforcement learning process, the attention network can be viewed as the agent, at the same time, images and strings act as the environment. At each step, the agent predicts the next character, that is the action, according to the policy p_θ defined by the parameters of the attention network. After generating the EOS token, the reward r can be observed and the correctness of the generated string can be indicated effectively. Before sampling the characters, we adjust the probability distribution as follows:

$$p_t^a = \frac{p_t}{\tau}, \quad t = 1, 2, \ldots, T, \quad (10)$$

where τ is a temperature hyper-parameter. Low temperature will result in a more concentrated distribution, that is, the larger differences between the values of the probability distribution. Through adjusting the probability distribution, we can get different sampled strings with different rewards to further adjust the network optimization. Denote the $y^s = (y_1^s, y_2^s, \ldots, y_T^s)$ as the sampled string from the model generated distributions. Our training objective is to minimize the negative expected reward:

$$L(\theta) = -\mathbb{E}_{y^s \sim p_\theta}[r(y^s)]. \tag{11}$$

The expectation is typically estimated by a single Monte-Carlo sample from p_θ in practice. We use the REINFORCE algorithm [25] to compute the policy gradient $\nabla_\theta L(\theta)$ of the non-differentiable, reward-based loss function. The expected gradient can be computed as follows:

$$\nabla_\theta L(\theta) = -\mathbb{E}_{y^s \sim p_\theta}[r(y^s)\nabla_\theta \log p_\theta(y^s)]. \tag{12}$$

The expected gradient is approximated with a single Monte-Carlo sample as well:

$$\nabla_\theta L(\theta) \approx -r(y^s)\nabla_\theta \log p_\theta(y^s). \tag{13}$$

In order to reduce the variance of the gradient estimate, a baseline b is reduced from the reward, without changing the expected gradient [24]. The estimate of the expected gradient with baseline b is given by:

$$\nabla_\theta L(\theta) \approx -(r(y^s) - b)\nabla_\theta \log p_\theta(y^s). \tag{14}$$

The baseline can be an arbitrary function as long as it does not depend on the selected action. Here, we formulate the baseline as the reward of the string obtained by the current model under the greedy decoding used in the inference process [26]. Given the baseline output $\hat{y} = (\hat{y_1}, \hat{y_2}, \ldots, \hat{y_T})$, in which $\hat{y_t}$ represents the character with the highest probability at time step t, Eq. 14 can be rewritten as follows:

$$\nabla_\theta L(\theta) \approx -(r(y^s) - r(\hat{y}))\nabla_\theta \log p_\theta(y^s). \tag{15}$$

The sampled string with the higher reward than the baseline will be encouraged, and conversely the sampled result with the lower reward will be suppressed.

E. Model Training

Since both the rewards for reinforcement learning are word-level, and the global optimization cannot provide specific character correction information. Therefore, we use a hybrid loss function, which is a weighted sum of the cross-entropy loss and the reward-based losses. The character-level and word-level measurements are complementary with each other and their combination provides a better optimization method.

In the process of training, the labeled images and the unlabeled images are mixed in each mini-batch. For labeled images, the objective function $L_{labeled}$ is constructed by considering the cross-entropy loss $L_{CE}(\theta)$, the edit reward based loss $L_{edit}(\theta)$ and the embedding reward based loss $L_{embedding}(\theta)$ as follows:

$$L_{labeled} = \lambda_1 L_{CE}(\theta) + \lambda_2 L_{edit}(\theta) + \lambda_3 L_{embedding}(\theta), \tag{16}$$

where λ_1, λ_2 and λ_3 are the manually set hyper-parameters to balance the three loss terms. For unlabeled images, we do not provide any ground truth, therefore the optimization objective $L_{unlabeled}$ is only the embedding reward based loss:

$$L_{unlabeled} = L_{embedding}(\theta). \tag{17}$$

In this way, we can train our model in the semi-supervised setting, with a small number of labeled images and a large number of unlabeled images. For labeled image, the generated string is forced to be consistent with its corresponding image and ground truth. And we only need the textual label, without any character-level or pixel-wise label. As for unlabeled image, it is natural that the generated result is encouraged to be close to the corresponding input image, without the requirement of any ground truth information.

We first train the embedding network with the labeled images. The CNN image encoder has the same architecture with the CNN in attention network. Hence, we pre-train the attention network with cross-entropy loss and inherit the parameters from its CNN. Then we train the embedding network with ranking loss. We gradually increase the difficulty during training with ranking loss. First, we consider all the negative image-string pairs in the mini-batch. Then the top three difficult negative pairs are taken into account. Finally, we adopt the hardest negative pair to further improve the discriminability of the embedding network. After training, the embedding network remains unchanged and can be used to measure the embedding reward during training of the attention network.

To train the attention network, we first pre-train it using cross-entropy loss as well. The pre-trained model serves as a good initialization for reinforcement learning. And then the network is trained with the hybrid loss of cross-entropy loss and reinforcement learning losses, using the mixed data of labeled images and unlabeled images.

IV. Experiments

In this section, we conduct extensive experiments to evaluate the proposed semi-supervised architecture on the challenging benchmark datasets commonly used in the literature. We evaluate the effectiveness of the designed rewards. In addition, we conduct the experiments to compare the performance of networks trained with the proposed semi-supervised training method and the traditional fully-supervised training method. Besides, we also explore the effect of different experimental settings and hyper-parameters.

A. Dataset

We evaluate our approach on the following public datasets:

- **Street View Text** [18] contains 647 word images which are cropped from 249 street-view images collected from Google Street View. Each image is associated with a 50 words lexicon defined by [18], denoted as SVT-50.
- **IIIT5K** [27] contains 3000 cropped word images collected from the Internet. Each image has a 50 words lexicon and a 1000 words lexicon, denoted as IIIT5k-50 and IIIT5k-1k separately.
- **ICDAR 2003** [28] contains 251 scene images and 860 cropped word images. Each image is specified with a 50 words lexicon defined by [18], which is denoted as IC03-50. And a full lexicon is composed of all the words that appear in the test set, denoted as IC03-Full.

TABLE I

THE CNN ARCHITECTURE OF THE ATTENTION NETWORK. THE CNN IN THE EMBEDDING NETWORK IS THE SAME AND IS INITIALIZED BY THE CNN IN THE PRE-TRAINED ATTENTION NETWORK

Layer name	Configurations
conv1	3×3, 64
conv2_X	$\begin{bmatrix} 3\times 3, 64 \\ 3\times 3, 64 \end{bmatrix} \times 2$
	pool: 2×2, stride 2×2
conv3_X	$\begin{bmatrix} 3\times 3, 128 \\ 3\times 3, 128 \end{bmatrix} \times 2$
	pool: 2×2, stride 2×2
conv4_X	$\begin{bmatrix} 3\times 3, 256 \\ 3\times 3, 256 \end{bmatrix} \times 2$
	pool: 2×1, stride 2×1
conv5_X	$\begin{bmatrix} 3\times 3, 512 \\ 3\times 3, 512 \end{bmatrix} \times 2$
	pool: 2×1, stride 2×1
conv6	3×3, 512

- **ICDAR 2013** [29] contains 1015 cropped word images without any lexicon, which derives from the ICDAR 2003.
- **ICDAR 2015** [30] contains 2077 word images including plenty of irregular text. No lexicon is specified.

Following the evaluation protocol in [18], we perform recognition on word images that contain only alphanumeric characters and at least three characters. For training data, our model is trained on the dataset released by [31], which contains around 8.9 million images. We divide about one percent of the dataset as the labeled data. For other data, we do not provide any ground truth. Besides, we will further explore the influence of the ratio of labeled data.

Except for the word-level accuracy, we also evaluate the total normalized edit distance (NED) (lower is better) on the standard benchmarks. Following [9], the NED is defined as $edit\_distance(pred, gt)/|gt|$, in which $pred$ and gt denote the generated string and ground truth, respectively.

In the process of testing, we report the results in both lexicon-free and lexicon-based setting. For lexicon-free recognition, we select the character with the highest probability at each step. For lexicon-based recognition, the word in the pre-defined lexicon with the smallest edit distance from the generated string is selected as the output string.

B. Implementation Details

For the attention network, the image encoder is composed of a CNN and a bidirectional RNN. Specifically, the architecture of CNN is described in Table I. The size of filters and channels for convolutional layers are shown. And the height, weight, stride of height, stride of weight are also specified for max pooling layers. There exists shortcut connection between each pair of 3×3 convolutional filters. All the convolution are performed with zero padding, ReLU activation function and batch normalization [32]. The bidirectional RNN has two layers and the hidden state dimension is set to be 1024. Besides, in order to accelerate the training process, we adopt the Simple Recurrent Unit (SRU) [33], the variant of RNN, which operates faster than traditional recurrent implementations. For the attention decoder, we use a GRU [34] with 512 hidden states and 37 output states (36 alphanumeric characters and 1 EOS token).

For the embedding network, the image encoder has the same architecture with the CNN in the attention network, that is, the structure in Table I. In addition, the string encoder is a SRU with 512 hidden states and the feature of word string is represented by the hidden state at the last time step. The image feature is extracted from the last layer of image encoder and then is projected to the embedding space by the linear mapping layers. The linear mapping layers contain three fully connected layers with 4096-dimension, 4096-dimension and 512-dimension separately.

In the process of training and testing, the input images are resized to 32 × 100 with gray scale. Firstly, the attention network is pre-trained using Adam [35] with a mini-batch size of 64. The pre-trained model is trained using cross-entropy loss with labeled images. The initial learning rate is set to 5×10^{-4} and is decreased by a factor of 0.8 every thirty epochs. Moreover, the gradient clipping is used at the magnitude of 5. The pre-trained model serves as a good initialization for reinforcement learning. Then we train the embedding network using Adam [35] with labeled images. We use the CNN in the pre-trained attention network as the initialization of the CNN in embedding network and fix the parameters of the CNN. Then we train the RNN string encoder and the linear mapping layers with a learning rate of 2×10^{-4} for 5 epoch and lower the learning rate to 2×10^{-5} for another 5 epoch. All the negative pairs are considered and the margin is 1. After that, we increase the difficulty and finetune the whole embedding network. We choose the top three hard negative pairs with margin 0.7. Finally, we only take the hardest negative pair into account with margin 0.3. The ranking loss decays to 1e-2. The average similarity of positive pairs is over 90 percent and the average similarity of negative pairs is below 50 percent. After that, we keep the parameters of the embedding network unchanged and train the attention network in the semi-supervised setting. Given the pre-trained attention network, we run the hybrid loss training using SGD with a learning rate of 5×10^{-4}. The batchsize is set to 64 and the unlabeled images accounted for half of each mini-batch. And we apply gradient clipping at the magnitude of 0.1.

The proposed network is implemented with Pytorch [36]. Most parts of our model are GPU-accelerated based on the CUDA backend. All the experiments are carried out on a workstation which has one Inter(R) Xeon(R) E5-2630 2.20Ghz CPU, an NVIDIA TITAN X GPU and 256GB RAM.

C. The Effect of Different Loss Terms

We investigate the effectiveness of our proposed rewards and summarize the results in Table II. The base model is trained only with cross-entropy loss using labeled data. After mixing the edit reward based loss for labeled data, the performance is significantly improved compared with the base model. The edit reward is designed for word-level

TABLE II

LEXICON-FREE PERFORMANCE OF THE DIFFERENT LOSS MIXING STRATEGIES ON PUBLIC BENCHMARKS. "TNED" REPRESENTS THE TOTAL NORMALIZED EDIT DISTANCE DEFINED IN SECTION IV.A

loss			SVT		IIIT5k		IC03		IC13		IC15	
L_{CE}	L_{edit}	$L_{embedding}$	accuracy	TNED	accuracy	TNED	accuracy	TNED	accuracy	TNED	accuracy	TNED
$\surd$			55.5	136.0	55.0	593.6	63.6	157.0	64.2	144.9	32.0	733.6
$\surd$	$\surd$		69.2	84.8	63.8	415.4	74.9	100.4	74.1	97.6	41.9	566.4
$\surd$		$\surd$	70.8	79.8	65.4	402.2	76.6	95.1	**77.1**	91.0	44.0	544.5
$\surd$	$\surd$	$\surd$	**72.2**	**75.9**	**68.3**	**354.4**	**78.7**	**85.0**	76.7	**83.2**	**46.0**	**539.4**

TABLE III

LEXICON-FREE ACCURACY RESULTS ON PUBLIC BENCHMARKS WITH DIFFERENT LOSS WEIGHTS λ_1, λ_2 AND λ_3 VALUES

λ_1	λ_2	λ_3	SVT	IIIT5k	IC03	IC13	IC15
0.25	0.25	0.5	71.5	66.3	78.1	**77.3**	45.6
0.25	0.375	0.375	**72.6**	67.6	**78.5**	76.6	**46.1**
0.25	0.5	0.25	70.2	67.2	76.8	76.4	45.7
0.5	0.15	0.35	71.2	66.6	76.7	75.9	42.6
0.5	0.25	0.25	70.7	**68.7**	77.0	75.7	44.1
0.5	0.35	0.15	71.1	67.3	76.3	76.5	43.1
0.75	0.1	0.15	70.8	66.1	76.1	76.7	43.6
0.75	0.125	0.125	70.5	67.1	75.8	76.4	42.7
0.75	0.15	0.1	71.5	65.9	75.6	76.6	42.0

TABLE IV

LEXICON-FREE TOTAL NED RESULTS ON PUBLIC BENCHMARKS WITH DIFFERENT LOSS WEIGHTS λ_1, λ_2 AND λ_3 VALUES

λ_1	λ_2	λ_3	SVT	IIIT5k	IC03	IC13	IC15
0.25	0.25	0.5	81.9	368.8	86.4	87.3	538.7
0.25	0.375	0.375	**75.4**	**351.1**	84.9	**84.1**	538.2
0.25	0.5	0.25	83.4	372.8	94.2	95.7	543.1
0.5	0.15	0.35	82.9	380.5	92.2	90.3	551.2
0.5	0.25	0.25	84.8	364.8	94.6	89.9	540.3
0.5	0.35	0.15	78.7	379.7	**84.0**	89.4	541.2
0.75	0.1	0.15	86.4	376.7	91.8	91.5	549.1
0.75	0.125	0.125	82.7	378.9	95.7	97.4	547.4
0.75	0.15	0.1	85.8	382.1	93.1	94.3	**537.9**

optimization. The results show that the model gains benefits from the combination of character-level and word-level optimization methods. Then we add the embedding reward based loss solely and evaluate the performance. With the mixing of embedding reward, the unlabeled data can also be utilized. Compared with the base model, the embedding reward brings about beyond ten percentages improvement. The results suggest that when there is only a small amount of labeled training data, our algorithm is able to effectively exploit the rich information of unlabeled images and significantly improve the performance without any ground truth information. Both the edit reward based loss and the embedding reward based loss can result in remarkable improvement. Therefore, both the rewards in our semi-supervised architecture are useful and we can effectively utilize a large number of unlabeled images. We also evaluate the performance of adding both the rewards. It is observed that the combination of the edit reward and embedding reward achieves the best performance, which verifies the effectiveness of our method.

Furthermore, in order to trade off the influence of different loss terms, we also explore how the weights among them should be set. We tune the hyper-parameters λ_1, λ_2 and λ_3 in our training objective function to adjust the ratios of the three loss items. To clearly show the impact of each loss, we set $\lambda_1 + \lambda_2 + \lambda_3 = 1$ in the ablation experiments. We conduct experiments on the standard benchmarks under different weights settings and the lexicon-free results are given in Table III and IV. We observed that the performance on different datasets is not exactly the same, but performs stably. In general, $\lambda_1 = 0.25$, $\lambda_2 = 0.375$ and $\lambda_3 = 0.375$ results in the relatively higher performance. Therefore, more emphasis on reinforcement learning losses performs better and both the proposed rewards are important.

TABLE V

LEXICON-FREE ACCURACY RESULTS ON PUBLIC BENCHMARKS WITH DIFFERENT τ VALUES

τ	SVT	IIIT5k	IC03	IC13	IC15
0.2	69.4	65.4	75.9	75.6	42.1
0.5	69.7	66.3	77.2	**77.0**	43.4
0.7	69.9	**66.5**	76.4	75.1	43.5
1	69.9	65.6	77.0	74.9	**44.8**
2	**71.3**	65.7	**78.0**	76.8	**44.8**
5	70.9	64.8	77.8	74.9	44.5

TABLE VI

LEXICON-FREE TOTAL NED RESULTS ON PUBLIC BENCHMARKS WITH DIFFERENT τ VALUES

τ	SVT	IIIT5k	IC03	IC13	IC15
0.2	91.6	413.9	101.4	92.5	578.6
0.5	87.0	393.5	95.3	87.7	580.6
0.7	83.9	395.3	94.3	93.1	568.9
1	86.3	398.9	**91.3**	91.9	561.9
2	**79.9**	**391.5**	92.6	**87.4**	**543.4**
5	81.0	424.6	98.9	97.4	550.8

D. The Effect of the Temperature Parameter

The hyper-parameter temperature τ adjusts the difference within the probability distribution to get the different sampled strings with different rewards and further adjust the network optimization. A larger value results in smaller difference between the values of the probability distribution. $\tau = 1$ means that the original probability distribution remains unchanged. We vary the value of the temperature τ and analyze the effect of the different values. The accuracy and total NED on the standard benchmarks in the lexicon-free setting are shown in Table V and Table VI, respectively. For different parameter setting, our approach performs stably and achieves the best performance with $\tau = 2$ in most situations. Therefore, narrowing the difference appropriately will lead to a better performance on some datasets.

E. The Effect of Mini-Batch Form

In the semi-supervised experimental setting, we mix labeled images and unlabeled images in each mini-batch. We explore

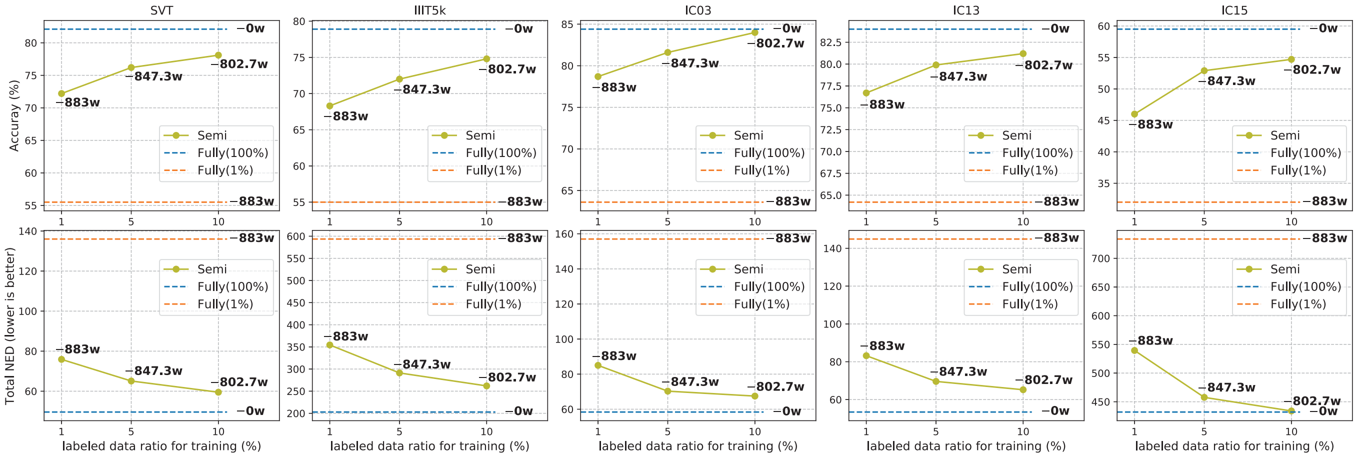

Fig. 4. The comparison between our semi-supervised method (Semi) and the traditional fully-supervised method (Fully) on standard benchmarks under different ratios of labeled samples in the training dataset. The first row is the accuracy, while the second row is the total normalized edit distance (NED) (lower is better). Each column corresponds to a dataset. The numbers after "-" represent the reduced amount of label data. With the same 1% labeled images, our semi-supervised approach significantly outperforms the fully-supervised method. Compared to the fully-supervised method using the entire dataset, we can achieve almost comparable performance with only 10% labeled images.

TABLE VII

LEXICON-FREE ACCURACY RESULTS ON PUBLIC BENCHMARKS WITH DIFFERENT PROPORTIONS OF FORMING A MINI-BATCH WITH LABELED IMAGES AND UNLABELED IMAGES. ALL LOSS WEIGHTS ARE SET TO 1 IN THE MIXED LOSS

ratio labeled : unlabeled	SVT	IIIT5k	IC03	IC13	IC15
3:1	70.6	65.3	77.0	75.0	41.0
2:1	70.6	**66.1**	77.6	75.4	43.7
1:1	**71.3**	65.7	**78.0**	**76.8**	**44.8**
1:2	70.3	66.0	76.7	75.6	43.6
1:3	69.9	66.0	76.5	76.1	44.2

TABLE VIII

LEXICON-FREE TOTAL NED RESULTS ON PUBLIC BENCHMARKS WITH DIFFERENT PROPORTIONS OF FORMING A MINI-BATCH WITH LABELED IMAGES AND UNLABELED IMAGES. ALL LOSS WEIGHTS ARE SET TO 1 IN THE MIXED LOSS

ratio labeled : unlabeled	SVT	IIIT5k	IC03	IC13	IC15
3:1	88.2	440.1	96.4	94.4	578.6
2:1	83.4	**379.6**	**91.3**	89.3	555.3
1:1	**79.9**	391.5	92.6	**87.4**	543.4
1:2	81.1	389.7	94.0	90.6	545.6
1:3	82.2	390.6	95.1	91.5	**536.8**

the proportion of forming a mini-batch with labeled images and unlabeled images. We train the network using five different proportions, 3:1, 2:1, 1:1, 1:2 and 1:3, while the other settings are the same. Then we summarize the lexicon-free accuracy and total NED on several benchmark datasets in Table VII and Table VIII, respectively. The results show that the proportion of 1:1 makes the accuracy relatively higher in most cases. But the total NED on the five benchmarks perform differently. Thus, the labeled images and unlabeled images need to be balanced properly in a mini-batch.

F. Comparisons With Traditional Training Method

For the attention network, traditional methods use cross-entropy loss to optimize it in fully-supervised setting. So we also conduct experiments using traditional fully-supervised training method with the same recognition network for comparison. The fully-supervised training requires the ground truth, therefore, only the labeled images can be used. In our semi-supervised setting, we use one percent of the dataset as the labeled data. So we adopt the same labeled images to train the network in fully-supervised setting and the results are shown in Figure 4, Table IX and Table X. With the same labeled images, our semi-supervised approach outperforms the traditional training method. It is proved that our method is able to effectively utilize the rich information of unlabeled images. And the training process benefits from the unlabeled data and the combination of character-level and word-level optimization methods. Then we use the entire dataset to train the network in fully-supervised setting. By contrast, our semi-supervised approach achieves promising performance, although only one percent of images are provided with labels in the dataset. Especially in the lexicon-based situations, our semi-supervised method obtains highly competitive performance, which demonstrates the effectiveness of our approach. Fuhermore, we also explore the effect of the ratio of labeled images and evaluate the performance in different ratio settings. As shown in Table IX and Table X, as the ratio of labeled data increases, the recognition results also gradually perform better. When we increase the ratio of labeled images to ten percent, our semi-supervised method can achieve nearly comparable performance compared with the traditional fully-supervised training method using the entire dataset.

G. Comparisons With State-of-the-Art

We compare our semi-supervised method trained using ten percent labeled images with state-of-the-art algorithms in the challenging benchmarks, including SVT, IIIT5k and ICDAR datasets. As shown in Table XI, all existing

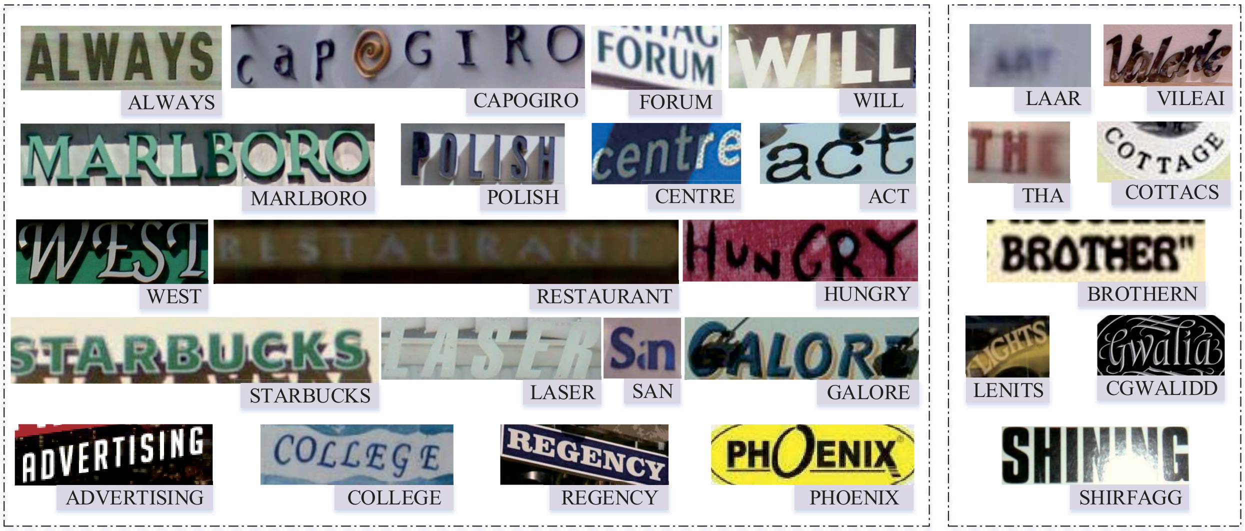

Fig. 5. Examples of lexicon-free scene text recognition results from SVT, IIIT5k and ICDAR datasets. The left displays the correctly recognized samples while the right shows the incorrect ones.

TABLE IX

Scene Text Recognition Accuracies on the Standard Benchmarks. "50", "1000" and "Full" Represent the Size of Dictionary Used for Lexicon-Based Recognition, and "None" Represents Lexicon-Free Recognition. "Fully" and "Semi" Represent the Network Trained With Traditional Fully-Supervised Training Method and Our Semi-Supervised Training Method, Respectively. The Percentage in the Brackets Represents the Ratio of Labeled Data in the Training Dataset. The Approximate Amounts of Labeled and Unlabeled Data Are Also Given

Methods	Labeled images	Unlabeled images	SVT		IIIT5k			IC03			IC13	IC15
			50	None	50	1k	None	50	Full	None	None	None
Fully (1%)	8.9w	-	90.6	55.5	90.6	85.0	55.0	94.1	83.8	63.6	64.2	32.0
Fully (100%)	891.9w	-	95.8	82.1	97.7	95.8	78.9	97.7	94.3	84.4	84.0	59.5
Semi (1%)	8.9w	883w	94.9	72.2	95.5	91.9	68.3	96.4	92.1	78.7	76.7	46.0
Semi (5%)	44.6w	847.3w	94.7	76.2	96.3	93.6	72.0	97.3	93.7	81.6	79.9	52.9
Semi (10%)	89.2w	802.7w	95.2	78.1	97.2	94.2	74.8	97.1	93.1	84.0	81.2	54.7

TABLE X

The Total NED on the Standard Benchmarks. "50", "1000" and "Full" Represent the Size of Dictionary Used for Lexicon-Based Recognition, and "None" Represents Lexicon-Free Recognition. "Fully" and "Semi" Represent the Network Trained With Traditional Fully-Supervised Training Method and Our Semi-Supervised Training Method, Respectively. The Percentage in the Brackets Represents the Ratio of Labeled Data in the Training Dataset. The Approximate Amounts of Labeled and Unlabeled Data Are Also Given

Methods	Labeled images	Unlabeled images	SVT		IIIT5k			IC03			IC13	IC15
			50	None	50	1k	None	50	Full	None	None	None
Fully (1%)	8.9w	-	53.7	136.0	236.3	350.0	593.6	51.7	117.7	157.0	144.9	733.6
Fully (100%)	891.9w	-	23.3	49.6	55.8	90.6	202.9	20.9	41.0	58.4	53.4	432.1
Semi (1%)	8.9w	883w	28.5	75.9	113.9	182.3	354.4	29.7	55.9	85.0	83.2	539.4
Semi (5%)	44.6w	847.3w	29.0	65.1	92.1	139.9	291.2	23.5	42.5	70.3	69.6	457.7
Semi (10%)	89.2w	802.7w	26.0	59.5	71.7	122.2	261.9	23.6	47.3	67.5	65.2	434.0

approaches are fully-supervised, some of which even require the character-level or pixel-wise annotations. By contrast, our method is the first work for semi-supervised scene text recognition, which saves expensive and time-consuming annotation efforts and costs. In lexicon-free setting, our method presents promising performance with significantly fewer labeled images compared with the state-of-the-art algorithms. And we just use the basic attention-based encoder-decoder framework as the recognition network, without the focusing attention mechanism [9] or edit probability [13]. It also should be remarked that the models in [9], [13], [53] are trained using large additional training dataset with about 4-million images. It is worth noting that our semi-supervised approach already achieves highly competitive performance in lexicon-based setting. Furthermore, our method even performs better than some fully-supervised approaches. For example, we outperform [21] by a margin of 6.4% on SVT. Note that we use significantly fewer labeled images compared with most existing approaches, the promising results suggest the significance of our method. In such a way, we are able to utilize more real word images without the requirement of annotation efforts.

Besides, following [51], considering [22] benefits from the pre-defined 90K dictionary, we also include the results using the same dictionary to post-process the predictions of

TABLE XI

SCENE TEXT RECOGNITION ACCURACIES ON THE BENCHMARK DATASETS. "50", "1000" AND "FULL" REPRESENT THE SIZE OF LEXICON USED FOR LEXICON-BASED RECOGNITION. "∗" [22] IS NOT LEXICON-FREE STRICTLY, DUE TO THE OUTPUT SEQUENCE IS CONSTRAINED TO A 90K DICTIONARY. "†" REPRESENTS THAT ADDITIONAL 4-MILLION DATA IS USED

Methods	SVT-50	SVT	IIIT5k-50	IIIT5k-1k	IIIT5k	IC03-50	IC03-Full	IC03	IC13	IC15
	Fully-supervised methods with word-level labels and character-level or pixel-wise labels									
† Cheng *et al.* [9]	97.1	85.9	99.3	97.5	87.4	99.2	97.3	94.2	93.3	70.6
Liu *et al.* [14]	96.1	-	96.9	94.3	86.6	98.4	97.9	93.1	92.7	-
Yang *et al.* [48]	95.2	-	97.8	96.1	-	97.7	-	-	-	-
Yao *et al.* [43]	81.0	-	85.6	72.7	-	90.3	82.6	-	-	-
Gordo [47]	91.8	-	93.3	86.6	-	-	-	-	-	-
	Fully-supervised methods with word-level labels									
† Bai *et al.* [13]	96.6	87.5	99.5	97.9	88.3	98.7	97.9	94.6	94.4	73.9
† Cheng et al. [53]	96.0	82.8	99.6	98.1	87.0	98.5	97.1	91.5	-	68.2
Jaderberg *et al.* [21]	93.2	71.7	95.5	89.6	-	97.8	97.0	89.6	81.8	-
Liu *et al.* [51] (unconstrained)	-	84.4	-	-	83.6	-	-	91.5	90.8	-
Liu *et al.* [51] (90k Dict)	-	87.6	-	-	-	-	-	93.3	93.7	-
∗ Jaderberg *et al.* [22]	95.4	80.7	97.1	92.7	-	98.7	98.6	93.1	90.8	-
ABBYY [18]	35.0	-	24.3	-	-	56.0	55.0	-	-	-
Wang *et al.* [18]	57.0	-	-	-	-	76.0	62.0	-	-	-
Mishra *et al.* [27]	73.2	-	64.1	57.5	-	81.8	67.8	-	-	-
Novikova et al. [37]	72.9	-	64.1	57.5	-	82.8	-	-	-	-
Wang *et al.* [38]	70.0	-	-	-	-	90.0	84.0	-	-	-
Goel *et al.* [39]	77.3	-	-	-	-	89.7	-	-	-	-
Bissacco *et al.* [40]	90.4	78.0	-	-	-	-	-	-	87.6	-
Alsharif and Pineau [41]	74.3	-	-	-	-	93.1	88.6	-	-	-
Almazán *et al.* [42]	89.2	-	91.2	82.1	-	-	-	-	-	-
Rodriguez-Serrano *et al.* [44]	70.0	-	76.1	57.4	-	-	-	-	-	-
Jaderberg *et al.* [45]	86.1	-	-	-	-	96.2	91.5	-	-	-
Su and Lu [46]	83.0	-	-	-	-	92.0	82.0	-	-	-
He *et al.* [3]	92.0	-	94.0	91.6	-	97.0	94.4	-	-	-
Shi *et al.* [1]	97.5	82.7	97.8	95.0	81.2	98.7	98.0	91.9	89.6	-
Shi *et al.* [4]	95.5	81.9	96.2	93.8	81.9	98.3	96.2	90.1	88.6	-
Lee and Osindero [5]	96.3	80.7	96.8	94.4	78.4	97.9	97.0	88.7	90.0	-
Liu *et al.* [49]	95.5	83.6	97.7	94.5	83.3	96.9	95.3	89.9	89.1	-
Ghosh *et al.* [50]	95.2	80.4	-	-	-	95.7	94.1	92.6	-	-
Wang and Hu [52]	96.3	81.5	98.0	95.6	80.8	98.8	97.8	91.2	-	-
	Semi-supervised methods with word-level labels									
Ours	95.2	78.1	97.2	94.2	74.8	97.1	93.1	84.0	81.2	54.7
Ours (ensemble)	95.8	80.8	97.3	94.2	76.8	97.3	94.5	85.8	84.5	57.6
Ours (90k Dict)	-	84.9	-	-	-	-	-	86.9	87.5	-
Ours (ensemble+90k Dict)	-	85.5	-	-	-	-	-	89.4	89.5	-

our model in lexicon-free setting on SVT, ICDAR03 and ICDAR13. The performance is further improved with the 90k dictionary, which is more competitive and validates the effectiveness of our method. In particular, using the same 90K dictionary, our method outperforms [22] by 4.2 percentages on SVT. Additionally, considering the analysis in [54] that the outputs of deep neural networks at different iterations always demonstrate diversity and complementarity, we also report the performance of an ensemble of models at different iterations from one learning process. Specifically, we average the last three checkpoints which are obtained from the same training process without increasing any additional training cost. The ensemble approach brings about improvement nearly on all benchmarks. When testing with both the 90k dictionary and ensemble approach, we can obtain highly competitive performance even compared with the fully-supervised methods that use character-level or pixel-wise labels. Therefore, our semi-supervised method can achieve comparative performance with the state-of-the-arts using some testing strategies, requiring no extra training cost. Furthermore, the proposed semi-supervised training method can be generalized to other scene text recognition network. The attention network can be replaced with other structures, such as CRNN [1] and Squeezedtext [14].

In addition, some examples predicted by our model in lexicon-free setting are shown in Figure 5. The correctly recognized images are presented in the left panel. We can see that our model can recognize some words suffered from noise and slight distortion. However, there are also some words that cannot be distinguished precisely and some incorrectly recognized samples are shown in the right. It is observed that the model fails to recognize the text images with low contrast, severe distortion and some special symbols disturbance.

V. CONCLUSION

In this work, we propose a novel semi-supervised method for scene text recognition. We design two word-level rewards and optimize them using reinforcement learning. The edit reward evaluates the distance between the ground truth and the generated string. Besides, the embedding reward measures the similarity between the input image and the generated string. By combining the reinforcement learning losses and traditional cross-entropy loss, the network can be effectively optimized in both character-level and word-level. It is worth noting that the computation of embedding reward does not need any ground truth information. Therefore, different from the existing approaches, our method has the ability to utilize a large number of unlabeled images. The semi-supervised way significantly reduces the requirement of expensive and laborious annotation efforts and costs. We validate the performance of our approach on the challenging benchmarks and extensive experimental results demonstrate the effectiveness.

REFERENCES

[1] 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 Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 11, pp. 2298–2304, Nov. 2017.

[2] C. Yi and Y. Tian, “Scene text recognition in mobile applications by character descriptor and structure configuration,” *IEEE Trans. Image Process.*, vol. 23, no. 7, pp. 2972–2982, Jul. 2014.

[3] P. He, W. Huang, Y. Qiao, C. C. Loy, and X. Tang, “Reading scene text in deep convolutional sequences,” in *Proc. AAAI Conf. Artif. Intell.*, 2016, pp. 3501–3508.

[4] B. Shi, X. Wang, P. Lyu, C. Yao, and X. Bai, “Robust scene text recognition with automatic rectification,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4168–4176.

[5] C.-Y. Lee and S. Osindero, “Recursive recurrent nets with attention modeling for OCR in the wild,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2231–2239.

[6] X. Cao, W. Ren, W. Zuo, X. Guo, and H. Foroosh, “Scene text deblurring using text-specific multiscale dictionaries,” *IEEE Trans. Image Process.*, vol. 24, no. 4, pp. 1302–1314, Apr. 2015.

[7] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, “Attention-based models for speech recognition,” in *Proc. Neural Inf. Process. Syst.*, vol. 2015, pp. 577–585.

[8] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, “Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks,” in *Proc. 23rd Int. Conf. Mach. Learn. (ICML)*, 2006, pp. 369–376.

[9] Z. Cheng, F. Bai, Y. Xu, G. Zheng, S. Pu, and S. Zhou, “Focusing attention: Towards accurate text recognition in natural images,” in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5086–5094.

[10] G.-J. Qi, X.-S. Hua, and H.-J. Zhang, “Learning semantic distance from community-tagged media collection,” in *Proc. ACM Int. Conf. Multimedia*, 2009, pp. 243–252.

[11] G.-J. Qi, C. Aggarwal, and T. Huang, “Towards semantic knowledge propagation from text corpus to Web images,” in *Proc. 20th Int. Conf. World wide web*, 2011, pp. 297–306.

[12] X. Shu, G.-J. Qi, J. Tang, and J. Wang, “Weakly-shared deep transfer networks for heterogeneous-domain knowledge propagation,” in *Proc. 23rd ACM Int. Conf. Multimedia*, Oct. 2015, pp. 35–44.

[13] F. Bai, Z. Cheng, Y. Niu, S. Pu, and S. Zhou, “Edit probability for scene text recognition,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1508–1516.

[14] Z. Liu, Y. Li, F. Ren, H. Yu, and W. Goh, “Squeezedtext: A real-time scene text recognition by binary convolutional encoder-decoder network,” in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 7194–7201.

[15] M. Ranzato, S. Chopra, M. Auli, and W. Zaremba, “Sequence level training with recurrent neural networks,” in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–16.

[16] X.-C. Yin, Z.-Y. Zuo, S. Tian, and C.-L. Liu, “Text detection, tracking and recognition in video: A comprehensive survey,” *IEEE Trans. Image Process.*, vol. 25, no. 6, pp. 2752–2773, Jun. 2016.

[17] Y. Zhu, C. Yao, and X. Bai, “Scene text detection and recognition: Recent advances and future trends,” *Frontiers Comput. Sci.*, vol. 10, no. 1, pp. 19–36, Feb. 2016.

[18] K. Wang, B. Babenko, and S. Belongie, “End-to-end scene text recognition,” in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 1457–1464.

[19] A. Mishra, K. Alahari, and C. V. Jawahar, “Top-down and bottom-up cues for scene text recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2687–2694.

[20] C. Yao, X. Bai, and W. Liu, “A unified framework for multioriented text detection and recognition,” *IEEE Trans. Image Process.*, vol. 23, no. 11, pp. 4737–4749, Nov. 2014.

[21] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, “Deep structured output learning for unconstrained text recognition,” in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–10.

[22] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, “Reading text in the wild with convolutional neural networks,” *Int. J. Comput. Vis.*, vol. 116, no. 1, pp. 1–20, Jan. 2016.

[23] A. Frome *et al.*, “Devise: A deep visual-semantic embedding model,” in *Proc. NIPS*, 2013, pp. 2121–2129.

[24] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 1998.

[25] R. J. Williams, “Simple statistical gradient-following algorithms for connectionist reinforcement learning,” *Mach. Learn.*, vol. 8, nos. 3–4, pp. 229–256, May 1992.

[26] S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, and V. Goel, “Self-critical sequence training for image captioning,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, p. 3.

[27] A. Mishra, K. Alahari, and C. Jawahar, “Scene text recognition using higher order language priors,” in *Proc. Brit. Mach. Vis. Conf.*, 2012, pp. 1–11.

[28] S. M. Lucas *et al.*, “ICDAR 2003 robust reading competitions: Entries, results, and future directions,” *Int. J. Document Anal. Recognit.*, vol. 7, nos. 2–3, pp. 105–122, Jul. 2005.

[29] D. Karatzas *et al.*, “ICDAR 2013 robust reading competition,” in *Proc. 12th Int. Conf. Document Anal. Recognit.*, Aug. 2013, pp. 1484–1493.

[30] D. Karatzas *et al.*, “ICDAR 2015 competition on robust reading,” in *Proc. 13th Int. Conf. Document Anal. Recognit. (ICDAR)*, Aug. 2015, pp. 1156–1160.

[31] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, “Synthetic data and artificial neural networks for natural scene text recognition,” in *Proc. NIPS Deep Learn. Workshop*, 2014, pp. 1–10.

[32] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

[33] T. Lei, Y. Zhang, S. I. Wang, H. Dai, and Y. Artzi, “Simple recurrent units for highly parallelizable recurrence,” 2017, *arXiv:1709.02755*. [Online]. Available: http://arxiv.org/abs/1709.02755

[34] K. Cho *et al.*, “Learning phrase representations using RNN encoder–decoder for statistical machine translation,” in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1724–1734.

[35] D. Kingma, J. Ba, and A. Adam, “Adam: A method for stochastic optimization,” in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–15.

[36] N. Ketkar, “Introduction to pytorch,” in *Proc. Deep Learn. Python*, vol. 2017, pp. 195–208.

[37] T. Novikova, O. Barinova, P. Kohli, and V. Lempitsky, “Large-lexicon attribute-consistent text recognition in natural images,” in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 752–765.

[38] T. Wang, D. J. Wu, A. Coates, and A. Y. Ng, “End-to-end text recognition with convolutional neural networks,” in *Proc. Int. Conf. Pattern Recog.*, 2012, pp. 3304–3308.

[39] V. Goel, A. Mishra, K. Alahari, and C. V. Jawahar, “Whole is greater than sum of parts: Recognizing scene text words,” in *Proc. 12th Int. Conf. Document Anal. Recognit.*, Aug. 2013, pp. 398–402.

[40] A. Bissacco, M. Cummins, Y. Netzer, and H. Neven, “PhotoOCR: Reading text in uncontrolled conditions,” in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 785–792.

[41] O. Alsharif and J. Pineau, “End-to-end text recognition with hybrid HMM maxout models,” in *Proc. Int. Conf. Learn. Represent.*, 2014, pp. 1–10.

[42] J. Almazan, A. Gordo, A. Fornes, and E. Valveny, “Word spotting and recognition with embedded attributes,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 12, pp. 2552–2566, Dec. 2014.

[43] X. Bai, C. Yao, and W. Liu, “Strokelets: A learned multi-scale mid-level representation for scene text recognition,” *IEEE Trans. Image Process.*, vol. 25, no. 6, pp. 2789–2802, Jun. 2016.

[44] J. A. Rodriguez-Serrano, A. Gordo, and F. Perronnin, “Label embedding: A frugal baseline for text recognition,” *Int. J. Comput. Vis.*, vol. 113, no. 3, pp. 193–207, Jul. 2015.

[45] M. Jaderberg, A. Vedaldi, and A. Zisserman, “Deep features for text spotting,” in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 512–528.

[46] B. Su and S. Lu, “Accurate scene text recognition based on recurrent neural network,” in *Proc. Asian Conf. Comput. Vis.*, 2014, pp. 35–48.

[47] A. Gordo, "Supervised mid-level features for word image representation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 2956–2964.

[48] X. Yang, D. He, Z. Zhou, D. Kifer, and C. L. Giles, "Learning to read irregular text with attention mechanisms," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 3280–3286.

[49] W. Liu, C. Chen, K.-Y. Wong, Z. Su, and J. Han, "STAR-net: A SpaTial attention residue network for scene text recognition," in *Proc. Brit. Mach. Vis. Conf.*, 2016, p. 7.

[50] S. K. Ghosh, E. Valveny, and A. D. Bagdanov, "Visual attention models for scene text recognition," in *Proc. 14th IAPR Int. Conf. Document Anal. Recognit. (ICDAR)*, Nov. 2017, pp. 943–948.

[51] W. Liu, C. Chen, and K. Wong, "Char-net: A character-aware neural network for distorted scene text recognition," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 1–8.

[52] J. Wang and X. Hu, "Gated recurrent convolution neural network for ocr," in *Proc. Neural Inf. Process. Syst.*, 2017, pp. 334–343.

[53] Z. Cheng, X. Liu, F. Bai, Y. Niu, S. Pu, and S. Zhou, "Arbitrarily-oriented text recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 5571–5579.

[54] C. Yang *et al.*, "AdaDNNs: Adaptive ensemble of deep neural networks for scene text recognition," 2017, *arXiv:1710.03425*. [Online]. Available: http://arxiv.org/abs/1710.03425

Yunze Gao received the B.S. degree from the University of Electronic Science and Technology of China in 2015 and the Ph.D. degree in pattern recognition and intelligence systems from the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, in 2020. Her current research interests include pattern recognition and machine learning, image and video processing, and scene text recognition.

Yingying Chen received the B.S. degree from the Communication University of China in 2013 and the Ph.D. degree from the University of Chinese Academy of Sciences in 2018. She is currently an Assistant Professor of pattern recognition and intelligence systems with the National Laboratory of Pattern Recognition, Chinese Academy of Sciences. Her current research interests include pattern recognition and machine learning, image and video processing, and intelligent video surveillance.

Jinqiao Wang (Member, IEEE) received the B.E. degree from the Hebei University of Technology, China, in 2001, the M.S. degree from Tianjin University, China, in 2004, and the Ph.D. degree in pattern recognition and intelligence systems from the National Laboratory of Pattern Recognition, Chinese Academy of Sciences, in 2008. He is currently a Professor with the Chinese Academy of Sciences. His research interests include pattern recognition and machine learning, image and video processing, mobile multimedia, and intelligent video surveillance.

Hanqing Lu (Senior Member, IEEE) received the B.E. and M.E. degrees from the Harbin Institute of Technology in 1982 and 1985, respectively, and the Ph.D. degree from the Huazhong University of Sciences and Technology in 1992. He is currently a Deputy Director of the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences. His research interests include image and video analysis, medical image processing, and object recognition.