SEED: Semantics Enhanced Encoder-Decoder Framework for Scene Text Recognition

Zhi Qiao[1,2] Yu Zhou[1*] Dongbao Yang[1] Yucan Zhou[1] Weiping Wang[1]
[1]Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[2]School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
{qiaozhi, zhouyu, yangdongbao, zhouyucan, wangweiping}@iie.ac.cn

Abstract

Scene text recognition is a hot research topic in computer vision. Recently, many recognition methods based on the encoder-decoder framework have been proposed, and they can handle scene texts of perspective distortion and curve shape. Nevertheless, they still face lots of challenges like image blur, uneven illumination, and incomplete characters. We argue that most encoder-decoder methods are based on local visual features without explicit global semantic information. In this work, we propose a semantics enhanced encoder-decoder framework to robustly recognize low-quality scene texts. The semantic information is used both in the encoder module for supervision and in the decoder module for initializing. In particular, the state-of-the-art ASTER method is integrated into the proposed framework as an exemplar. Extensive experiments demonstrate that the proposed framework is more robust for low-quality text images, and achieves state-of-the-art results on several benchmark datasets. The source code will be available.[†]

Figure 1. The comparison of our SEED with the existing encoder-decoder framework such as [45]. The first column shows the examples of some challenging scene text including image blur, occlusion, and background interference. The second column is the results of the existing encoder-decoder framework and the third column gives the predictions of our approach. It shows that our proposed method is more robust to the low-quality images.

1. Introduction

Scene text detection and recognition have attracted great attention in recent years owing to its various applications such as autonomous driving, road sign recognition, helping visual impaired and so on. Inspired by object detection [27, 40, 26, 58], scene text detection [24, 48, 60, 38, 6] achieved convincing performance. Despite the maturity of conventional text recognition in documents, scene text recognition is still a challenging task.

With the development of deep learning, recent works [16, 15, 43, 46, 22, 44, 45, 54, 7, 8, 2, 23, 25, 57, 52, 32, 53] on scene text recognition have shown promising results. However, existing methods are still facing various problems when dealing with image blur, background interference, occlusion and incomplete characters as shown in Fig. 1.

Recently, inspired by neural machine translation of the natural language processing field, the encoder-decoder framework with attention mechanism has been widely used in scene text recognition. For regular text recognition [22, 7, 10], the encoder is based on CNN with RNN and another RNN with attention mechanism is used as the decoder to predict character at each time step. For irregular text recognition, the rectification based methods [44, 45, 28, 57, 32, 53], the multi-direction encoding method [8] and the 2D-attention based methods [54, 23] are proposed. Rectification based methods first rectify the irregular images, then the following pipeline is as those of regular recognition. The multi-direction encoding method uses CNN with two LSTMs to encode four different directions. The

*The corresponding author
[†]https://github.com/Pay20Y/SEED

2D-attention based methods use 2D-attention mechanism to deal with irregular text which handles feature map from two dimensions directly.

The existing methods define the text recognition task as a sequence character classification task locally, but ignore the global information of the whole word. As a result, they may struggle to handle low-quality images such as image blur, occlusion and incomplete characters. However, people can deal with these low-quality cases well by considering the global information of the text.

To address this problem, we propose the **S**emantics **E**nhanced **E**ncoder-**D**ecoder framework (SEED), in which an additional semantic information is predicted acting as the global information. The semantic information is then used to initialize the decoder as illustrated in Fig. 2 (c). The semantic information has two main advantages, 1) it can be supervised by a word embedding in natural language processing field, 2) it can reduce the gap between the encoder focusing on the visual feature and the decoder focusing on the language information, since the text recognition can be regarded as a cross-modality task. Specifically, we get the word embedding from a pre-trained language model and compute a loss between the semantic information and the word embedding during training. By this way, the semantic information contains richer semantics, then the predicted semantic information is used to guide the decoding process. As a result, the decoding process can be limited in a semantic space, and the performance of recognition will be better. Some examples are shown in Fig. 1. As an example, in the fourth sub-image of Fig. 1, the last two characters "se" are recognized as "R" because of the occlusion, but it can be corrected in our framework with the global semantic information. In other words, the semantic information works as an "intuition", which is like a glimpse before people read a word carefully.

Predicting semantic information from images directly has already been studied before. [12] predicts semantic concepts directly from a word image with a CNN and a weighted ranking loss. [51] tries to embed image features into a word embedding space for text spotting. [21] proposes to learn embedding of the word images and the text labels in an end-to-end way. These works validate that semantic information is helpful to the text related tasks.

The main contributions are as follows:

1. We propose SEED for scene text recognition, which predicts additional global semantic information to guide the decoding process, and the predicted semantic information is supervised by the word embedding from a pre-trained language model.

2. We integrate the state-of-the-art ASTER method [45] to our framework as an exemplar.

3. Extensive experiments on several public scene text benchmarks demonstrate the proposed framework can obtain state-of-the-art performance, especially on the low-quality datasets ICDAR2015 and SVT-Perspective, and it is particularly more robust for incomplete characters.

The rest of this paper is organized as follows: Sec. 2 reviews the related works, Sec. 3 describes the proposed framework and the exemplar, Sec. 4 conducts profuse experiments and Sec. 5 concludes the work.

2. Related Work

2.1. Scene Text Recognition

Existing scene text recognition methods can be divided into two categories, namely traditional methods and deep learning based methods.

Traditional methods usually adopt a bottom-up approach which detects and classifies characters first and then groups them to a word or text line with heuristic rules, language models or lexicons. They design various hand-craft features then use these features to train a classifier such as SVM. For example, [34] uses a set of computationally expensive features like aspect ratio, hole area ratio, etc. [50, 49] use sliding windows with HOG descriptors, and [55, 3] use Hough voting with random forest classifier. Most traditional methods suffer from designing various hand-crafted features, and these features are limited for high-level representation.

With the development of deep learning, most methods use CNN to perform a top-down approach which recognizes word or text line directly. [16] treats a word as a class, then converts the recognition problem into the image classification problem. Recently, most works treat the recognition problem as the sequence prediction problem. Existing methods can be almost divided into two techniques namely Connectionist Temporal Classification (CTC) and attention mechanism. For CTC-based decoding, [15, 43, 46] propose to use CNN and RNN to encode the sequence features and use CTC for character alignment. For attention-based decoding, [22] proposes recursive CNN to capture longer contextual dependencies and uses an attention-based decoder for sequence generation. [7] introduces the problem of attention drift, and proposes focusing attention for better performance.

However, these works all assume that the text is horizontal, and can not handle the text of irregular shapes such as perspective distortion and curvature. To solve the problem of irregular text recognition, [44, 45] propose to rectify the text first based on Spatial Transformer Network [17] and then treat it as horizontal text. Furthermore, [57] gets better performance with iterative rectification and [53] rectifies with some geometric constraints. [32] rectifies text by predicting pixels offset. Instead of rectifying the whole text, [28] takes an approach of detecting and rectifying individual characters. In spite of rectification, [8] encodes the images in four directions and proposes a filter gate to fuse the features. [54] introduces an auxiliary dense character detection task and an

alignment loss into the 2D attenton based network. [23] proposes a tailored 2D attention based framework for irregular text recognition. Without encoder-decoder framework, [25] converts irregular text recognition into character segmentation with fully convolutional network [31]. [52] proposes a new loss function for more effective decoding.

2.2. Semantics in Scene Text

Many works try to bring semantics into the tasks of text recognition or text retrieval. [12] predicts semantic concepts directly from a word image with CNN. [36] proposes to generate contextualized lexicons for scene images with only visual information, and word-spotting task benefits a lot from the lexicons. [51, 21] learn to map the word images to a word embedding space and apply it into word spotting system. [18] tries to detect and recognize text in online images with the help of context information such as tags, comments, and titles. [42] introduces to use the language model and the semantic correlation between scene and text to re-rank the recognition results. [37] proposes to boost the performance of text spotting with the object information. [11] uses the text embedded in advertisement images to enhance the image classification. [59] proposes to use a pre-trained language model to correct the inaccurate recognition results with the text context in the image.

As discussed before, state-of-the-art recognition methods do not utilize the semantics of the text well. The related semantics works do not integrate the semantics into the recognition pipeline explicitly and effectively.

3. Method

In this section we describe the proposed method in detail. The general framework is shown in Fig. 2 (c), which consists of 4 major components: 1) The *encoder* including CNN backbone and RNN for extracting visual features; 2) The *semantic module* for predicting semantic information from the visual features; 3) The *pre-trained language model* for supervising the semantic information predicted by *semantic module*; 4) The *decoder* including RNN with attention mechanism for generating the recognition results. First we review the encoder-decoder framework in Sec. 3.1, and introduce the pre-trained language model detailedly in Sec. 3.2. In Sec. 3.3, we describe our proposed method. Specifically, we present the general framework in Sec. 3.3.1. After that, we show the details of the proposed method which integrate state-of-the-art method ASTER [45] into proposed framework in Sec. 3.3.2. Finally, the loss function and the training strategies are presented in Sec. 3.4.

3.1. Encoder-Decoder Framework

The Encoder-decoder framework is widely used in neural machine translation, speech recognition, text recognition and so on. [47] first introduces the structure of the framework

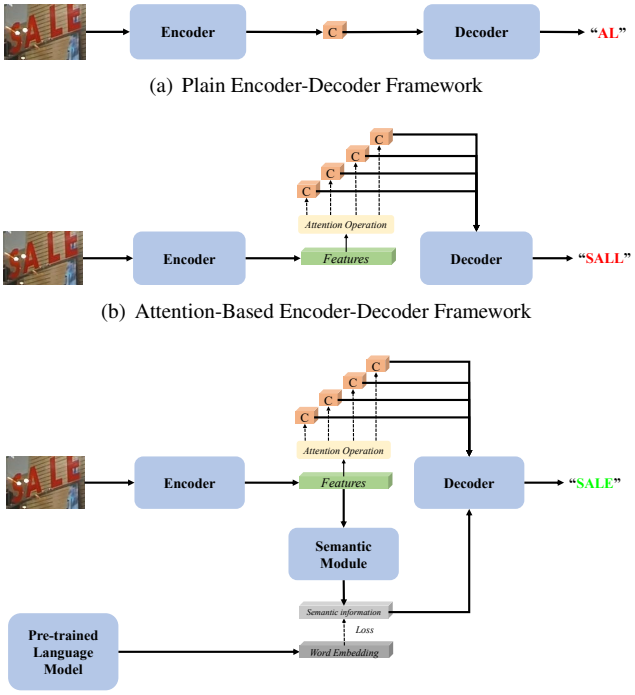

Figure 2. Comparison of three kinds of framework. "C" represents context information. The plain encoder-decoder framework gets incorrect results due to limited context representation. The attention-based encoder-decoder framework works better but still can not handle incomplete characters without global information. Our proposed encoder-decoder framework predicts the correct result with the help of global semantic information.

and applies it into neural machine translation. For simplicity, we call this framework plain encoder-decoder framework. As visualized in Fig. 2 (a), the encoder extracts rich features and generates a context vector C which contains global information of the inputs, then the decoder converts the context vector to target outputs. Source inputs and target outputs are different due to different tasks, as for text recognition, the inputs are images and target outputs are the texts in the images. The specific composition of encoder and decoder is not fixed, CNN and LSTM are all common choices.

Despite great effectiveness, the plain encoder-decoder framework has an obvious drawback, where the context information has limited ability to represent the whole inputs. Inspired by human visual attention, researchers introduce the attention mechanism into the encoder-decoder framework, which is defined as the attention-based encoder-decoder framework. As shown in Fig. 2 (b), attention mechanism attempts to build shortcuts between the context and the whole inputs. The decoder can select the appropriate context at each decoding step which is capable of resolving long-range dependency problems, and the alignment between encoder

and decoder is trained in a weakly supervised way.

For scene text recognition, the decoder only depends on the limited local visual features for decoding in both the plain encoder-decoder framework and the attention-based encoder-decoder framework, so it is difficult to deal with some low-quality images without global information. In our proposed framework, the encoder learns explicit global semantic information and uses it as guidance for the decoder. We use FastText [4] to generate word embedding as the supervision of the semantic information in that it can solve the problem of "out of vocabulary".

3.2. FastText Model

We choose FastText as our pre-trained language model, which is based on skip-gram. Let $T = \{w_{i-l}, \ldots, w_{i+l}\}$ be a sentence in a text corpus. l indicates the length of the sentence and is a hyper-parameter. In skip-gram, a word w_i is represented by a single embedding vector v_i and then input to a simple feed-forward neural network, which aims to predict the context represented as $C_i = \{w_{i-l}, \ldots, w_{i-1}, w_{i+1}, \ldots, w_{i+l}\}$. With training the feed-forward network, the embedding vector is simultaneously optimized, and the final embedding vector of a word is close to the words with similar semantics.

FastText additionally embeds subwords and uses them to generate final embedding of the word w_i. Given the hyper-paramters l_{min} and l_{max} denoting a minimum and a maximum length of the subwords. For example, let $l_{min} = 2$, $l_{max} = 4$ and the word be "where", the set of subwords is $\{$*wh*, *he*, *er*, *re*, *whe*, *her*, *ere*, *wher*, *here*$\}$. The word representation is obtained by the combination of the embedding vectors of all subwords and the word itself. Accordingly, FastText model can handle the problem of "out of vocabulary". There are some novel words or incomplete words in the benchmark datasets such as ICDAR2015 and SVT-Perspective, so FastText is suitable for our framework.

3.3. SEED

3.3.1 General Framework

Many scene text recognition methods are based on the encoder-decoder framework with attention. The decoder focuses on specific regions of visual features and outputs corresponding characters step by step. The framework works well in most scenarios except in low-quality images. In some low-quality images, texts may be blurred or occluded. To address these problems, utilizing global semantic information is an alternative. The proposed framework is shown in Fig. 2 (c). Different from the attention-based encoder-decoder framework, the proposed semantic module predicts extra semantic information. Further, we use the word embedding from a pre-trained language model as the supervision to improve the performance. After that, the semantic information is fed into the decoder along with the visual features.

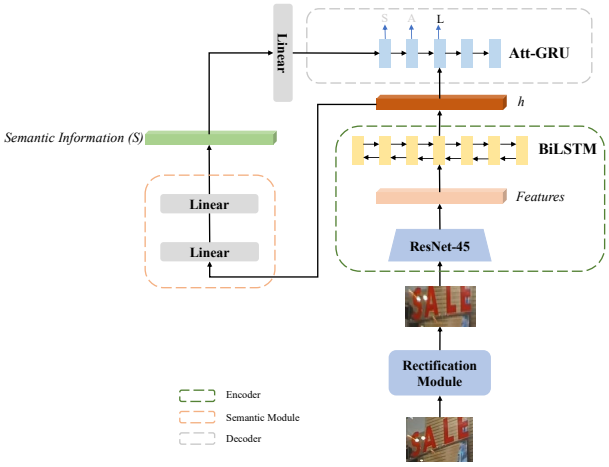

Figure 3. Details of our SE-ASTER. It consists of four main modules, rectification module, encoder, semantic module, and decoder. The semantic module predicts semantic information from the outputs of the encoder which is fed into decoder as the guidance.

In this way, our method is robust to low-quality images and can correct recognition mistakes.

3.3.2 Architecture of Semantics Enhanced ASTER

We use ASTER [45] as an exemplar for our proposed framework, and we call the proposed method **S**emantics **E**nhanced **ASTER** (SE-ASTER). The SE-ASTER is illustrated in Fig. 3. There are four modules: the rectification module is to rectify the irregular text images, the encoder is to extract rich visual features, the semantic module is to predict semantic information from the visual features, and the decoder transcribes the final recognition results.

First, the image is input to the rectification module to predict control points with a shallow CNN, then Thin-plate Splines [5] is applied to the image. In this way, the distorted text image will be rectified. This module is the same as [45], so we don't describe it in detail. Thereafter, the rectified image will be input to the encoder, and rich visual features can be generated. Specifically, the encoder consists of a 45-layer ResNet based CNN same as [45] and a 2-layer Bidirectional LSTM [13] (BiLSTM) network with 256 hidden units. The output of the encoder is a feature sequence $h = (h_1, \ldots, h_L)$ with the shape of $L \times C$, where L is the width of the last feature map in CNN, and C is the depth.

The feature sequence h has two functions, one is to predict the semantic information by the semantic module and the other is as the input of the decoder. For predicting semantic information, we first flatten the feature sequence into a one-dimensional feature vector I with dimension of K, where $K = L \times C$. The semantic information S is predicted

with two linear functions as following:

$$S = W_2\sigma(W_1 I + b_1) + b_2\,. \tag{1}$$

where W_1, W_2, b_1, b_2 are trainable weights in the linear function, σ is a ReLU activation function. We also evaluate predicting the semantic information with the final hidden state h_L of BiLSTM in the encoder, and it gets worse performance. It may originate from that predicting semantic information needs larger feature contexts and it is more proper to use the BiLSTM outputs. The semantic information will be supervised by the word embedding provided by the pre-trained FastText model. The loss function used here will be introduced in Sec. 3.4.

The decoder adopts the Bahdanau-Attention mechanism [1] which consists of a single layer attentional GRU [9] with 512 hidden units and 512 attention units. Different from [45] we use a single direction decoder here. In particular, the semantic information S is used to initialize the states of GRU after a linear function for transforming the dimension. Instead of using zero-state initializing, the decoding process will be guided with global semantics, so the decoder uses not only local visual information but also global semantic information to generate more accurate results.

3.4. Loss Function and Training Strategy

We add supervision at both the semantic module and the decoder module. SE-ASTER is trained end-to-end. The loss function is as follows:

$$L = L_{rec} + \lambda L_{sem}\,. \tag{2}$$

where L_{rec} is the standard cross-entropy loss of the predicted probabilities with respect to the ground-truth, and L_{sem} is the cosine embedding loss of the predicted semantic information with respect to the word embedding of the transcription label from the pre-trained FastText model. λ is hyper-parameters to balance the loss, and we set it to 1 here. Note that we just use a simple cosine based loss function here instead of contrastive loss for faster training speed.

$$L_{sem} = 1 - cos(S, em)\,. \tag{3}$$

where S is the predicted semantic information and em is the word embedding from pre-trained FastText model.

There are two training strategies. The first is initializing the state of the decoder with the word embedding from the pre-trained FastText model rather than the predicted semantic information. Another is to use the predicted semantic information directly. We evaluate these two strategies, and their performances are similar. We use the second training strategy which trains the model in a pure end-to-end way.

4. Experiments

In this section, we conduct extensive experiments to verify the effectiveness of our proposed method. First, we introduce the datasets used for training and evaluation, and the implementation details are described. Next, we perform ablation studies to analyze the performance of the different strategies. Finally, our method is compared with state-of-the-art methods on several benchmarks.

4.1. Datasets

IIIT5K-Words (IIIT5K) [33] contains 5000 images, most of which are regular samples. There are 3000 images for testing. Each sample in test set is associated with a 50-word lexicon and a 1k-word lexicon.

Street View Text (SVT) [49] consists of 647 cropped word images from 249 street view images. Most of word images are horizontal, but some of them are severely corrupted by noise, blur, and low resolution. A 50-word lexicon is provided for each image.

SVT-Perspective (SVTP) [39] contains 645 word images for evaluation. most images suffer in heavy perspective distortions which are difficult for recognition. Each image is associated with a 50-word lexicon.

ICDAR2013 (IC13) [20] consists of 1015 images for testing, most of which are regular text images. Some of them are under uneven illumination.

ICDAR2015 (IC15) [19] was collected without careful capture. Most of images are with various distortions and blurry which are challenging for most existing methods.

CUTE80 (CUTE) [41] consists of 288 word images only for evaluation. Most of them are curved but with high resolution, no lexicon is provided.

Synth90K [16] consists of 9 million synthetic images generated from a lexicon of 90K words. It has been widely used in text recognition task. We use it as one of our training datasets. It contains words from the testing set of the IC13 and SVT.

SynthText [14] is another synthetic dataset for text detection task. We crop the words with ground-truth word bounding boxes and use for training our model.

4.2. Implementation Details

The proposed SE-ASTER is implemented in PyTorch [35]. The pre-trained FastText model is the officially available model[1] trained on *Common Crawl*[2] and *Wikipedia*[3]. In total 97 symbols are recognized, including digits, upper-case and lower-case letters, 32 punctuation marks, end-of-sequence symbol, padding symbol, and unknown symbol.

[1] https://fasttext.cc/docs/en/crawl-vectors.html
[2] https://commoncrawl.org/
[3] https://www.wikipedia.org/

Methods \ Images	IC13-sr				IC15-sr			
								
ASTER	ier	for	irst	cooker	poon	echars	mest	xerr
ASTER+WES	herb	room	its	ooker	spoon	rechery	inest	verry
ASTER+INIT	here	look	first	looker	poon	keekers	inest	merri
SE-ASTER	here	room	first	cookery	spoon	kechers	finest	merry

Table 1. Visualization of the recognition results on the two shrink datasets. Red: wrong results; Green: correct results.

Methods	WES	INIT	IC13	SVTP	IC15
ASTER [45]			91.8	78.5	76.1
ASTER-r			90.9	79.1	78.4
ASTER	✓		90.8	79.2	77.0
ASTER		✓	91.1	78.1	76.1
ASTER	✓	✓	**92.8**	**81.4**	**80.0**

Table 2. Performance comparison between different strategies. WES represents word embedding supervision. INIT represents initializing the state of the GRU in the decoder. ASTER-r represents the model re-trained by ourselves.

The size of input images are resized to 64×256 without keeping ratio, and we adopt the ADADELTA [56] to minimize the objective function. Without any pre-training and data augmentation, our model is trained on SynthText and Synth90K for 6 epochs with the batch size of 512, the learning rate is set to 1.0 and is decayed to 0.1 and 0.01 at the 4th epoch and the 5th epoch. The model is trained on one NVIDIA M40 graphics card.

For evaluation, we resize the input images to the same size as for training. We use beam search for GRU decoding, which keeps the k candidates with the highest accumulative scores, where k is set to 5 in all our experiments.

4.3. Ablation Study

There are two steps about the semantic module, one is the word embedding supervision and the other is initializing decoder with the predicted semantic information. We evaluate these two steps separately by using the Synth90K and SynthText as training data consistently. The results are shown in Tab. 2. The model supervised with word embedding only does not improve the performance compared with the baselines. Using predicted holistic features from the encoder to initialize decoder improves the performance by almost 0.2% in ICDAR13, but gets worse performance on SVTP and IC15. It shows that learning global information in an implicit weakly supervised way still struggles with low-quality images. A combination of these two steps gets the best performance. The improvements of 1.9%, 2.3% and 1.6% are obtained on IC13, SVTP and IC15 respectively. Compared with ASTER without word embedding supervision, it improves the accuracy by 1.7% on IC13, 3.3% on SVTP and 3.9% on IC15, which verifies that the supervision with word embedding is quite important.

4.4. Performance with Inaccurate Bounding Boxes

Scene text recognition in real applications is always combined with the detection branch to achieve an end-to-end pipeline. However, the detection branch may not output ideal bounding boxes. If text recognition is robust to inaccurate detection results, the overall end-to-end performance can be more satisfactory. Limited by the receptive field of CNNs, the most frequent inaccurate detection is incomplete characters. We conduct experiments to show our method is robust with this situation. Here we also use SE-ASTER as an exemplar. Note that the SE-ASTER is only trained on Synth90K and SynthText without any data augmentation such as random cropping. We first generate two shrink datasets IC13-sr and IC15-sr based on IC13 and IC15 respectively as follows.

We randomly remove the original word images up to 15% in the left, right, top and bottom directions simultaneously. All of the cropped images still have an intersection over union with the original ones larger or equal than $(1-0.15\times 2)^2 = 0.49$. According to the evaluation protocol of detection, these cropped images are all positive localizations because the IoU is above the standard threshold of 0.5. Some examples are shown in Tab. 1.

Methods	IC13 IC13-sr	GAP	IC15 IC15-sr	GAP
ASTER	90.9 71.4	-19.5	78.4 65.6	-12.8
ASTER+WES	90.8 71.9	-18.9	77.0 62.8	-14.2
ASTER+INIT	91.1 74.6	-16.5	76.1 63.1	-13.0
SE-ASTER	92.8 77.4	**-15.4**	80.0 70.0	**-10.0**

Table 3. Results on the shrink datasets, GAP indicates the decline between two datasets.

The quantitative results are illustrated in Tab. 3. The performances of the ASTER baseline drop 19.5% and 12.8% on the IC13-sr dataset and the IC15-sr dataset respectively, which reveal that the ASTER baseline suffers a lot from the incomplete characters. However, with the supervision of

Input Images	SAR SE-SAR	ASTER SE-ASTER
	baf bar	batf bar
	ale sale	ale sale
	orchano orchard	orghand orchard
	nex mex	me mex
	martis martin	martia martin
	xccessorize accessorize	recessorize accessorize
	hilfiger hilfger	hilfgger hilfger

Figure 4. Examples of low-quality images and recognition results in four methods. Red characters are the wrong results, and green ones are the correct.

word embedding, the model still struggles with the shrink images. Using the holistic information from encoder as the guidance of the decoder gets better results with 16.5% and 13.0% decline. SE-ASTER gets the best results, which shows that our model is more robust with incomplete characters. Some visualizing examples are illustrated in Tab. 1.

4.5. Generalization of Proposed Framework

To verify the generalization of SEED, we integrate another state-of-the-art recognition method SAR [23]. SAR is a 2D-attention based recognition method without rectification on input images, and it already adopts an LSTM to generate a holistic feature. However as we mentioned before, the holistic feature may be not effective in a weakly supervised training strategy, so we make some modifications and call our new model **S**emantics **E**nhanced **SAR** (SE-SAR).

In SE-SAR, we replace the max-pooling along the vertical axis with a shallow CNN. The output of the shallow CNN is a feature map with the height of 1, then the feature map is fed into a 2-layer LSTM to extract context information. Two linear functions are applied to the output of LSTM to predict the semantic information. Except for the 2D-attention decoder in SAR, we apply another decoder to the output of the LSTM and supervise with the transcription labels. In this way, the output of LSTM contains richer information and helps predict semantic information. Finally, the semantic information is used to initialize the LSTM of the decoder. The model is trained on Synth90K and SynthText for 2 epochs with the batch size of 128.

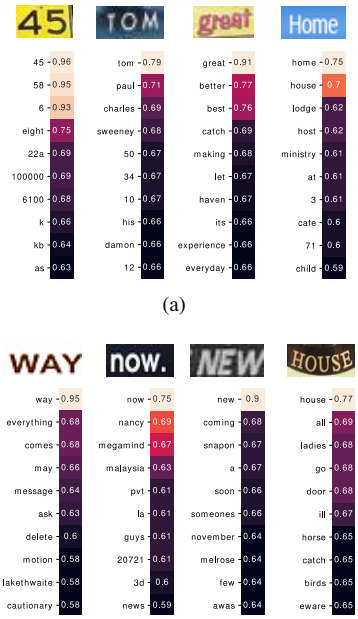

Figure 5. Visualization of cosine similarity of the predicted semantic information from the image w.r.t the word embedding of the words from lexicons. Larger value means more similar semantics.

Methods	IC13	IC15	SVT	SVTP
SAR [23]	**91.0**	69.2	84.5	76.4
SE-SAR	90.9	**73.4**	**85.8**	**78.7**

Table 4. Recognition performance on SAR and SE-SAR.

We conduct some experiments on IC13, IC15, SVT, and SVTP to show the effectiveness of the SE-SAR. The results are demonstrated in Tab. 4. Compared with the baseline, our SE-SAR improves 4.2%, 1.3% and 2.3% on IC15, SVT, and SVTP respectively. SE-SAR is only comparable with SAR in that low-quality images are scarce in IC13.

4.6. Qualitative Results and Visualization

We visualize low-quality images including blur or occlusion. Some examples are shown in Fig. 4. As can be seen, our proposed methods SE-ASTER and SE-SAR are robust with low-quality images. We explain that semantic information will provide an effective global feature to decoder, which is robust to the interference in the images.

We also perform experiments on IIIT5K to visualize the validity of the predicted semantic information. As illustrated in Fig. 5, we compute the cosine similarity between the predicted semantic information and the word embedding of each word from lexicons (50 words for each image). In

Methods	IIIT5K	SVT	IC13	IC15	SVTP	CUTE
Shi *et al.* [43]	81.2	82.7	89.6	-	-	-
Shi *et al.* [44]	81.9	81.9	88.6	-	71.8	59.2
Lee *et al.* [22]	78.4	80.7	90.0	-	-	-
Yang *et al.* [54]*	-	-	-	-	75.8	69.3
Cheng *et al.* [7]*	87.4	85.9	93.3	70.6	-	-
Cheng *et al.* [8]	87.0	82.8	-	68.2	73.0	76.8
Liu *et al.* [28]*	92.0	85.5	91.1	74.2	78.9	-
Bai *et al.* [2]*	88.3	87.5	**94.4**	73.9	-	-
Liu *et al.* [30]*	87.0	-	92.9	-	-	-
Liu *et al.* [29]	89.4	87.1	94.0	-	73.9	62.5
Liao *et al.* [25]*	91.9	86.4	91.5	-	-	79.9
Zhan *et al.* [57]	93.3	**90.2**	91.3	76.9	79.6	83.3
Xie *et al.* [52]	-	-	-	68.9	70.1	82.6
Li *et al.* [23]	91.5	84.5	91.0	69.2	76.4	83.3
Luo *et al.* [32]	91.2	88.3	92.4	74.7	76.1	77.4
Yang *et al.* [53]*	**94.4**	88.9	93.9	78.7	80.8	**87.5**
ASTER [45]	93.4	89.5	91.8	76.1	78.5	79.5
ASTER baseline reproduced	93.5	87.2	90.9	78.4	79.1	82.3
SE-ASTER (Ours)	93.8	89.6	92.8	**80.0**	**81.4**	83.6

Table 5. Lexicon-free performance on public benchmarks. **Bold** represents the best performance. Underline represents the second best result. * indicates using both word-level and character-level annotations to train model.

Fig. 5 (a), the predicted semantic information is very related to the words which have similar semantics. For example, "home", "house", and "lodge' all have the meaning of residence. "Tom", "Paul" and "Charles" are all common names. The second row illustrates the robustness of the predicted semantic information. For example, "house" and "horse" have a similar spelling and are of the edit distance of 1, but their semantics are quite different as shown in Fig. 5 (b). With the help of global semantic information, the model can distinguish them easily.

4.7. Comparison with State-of-the-art

We also compare our methods with previous state-of-the-art methods on several benchmarks. The results are shown in Tab. 5. Compared with other methods, we achieve 2 best results and 3 second best results out of 6 in the lexicon-free scenario with only word-level annotations.

Our proposed method works effectively on some low-quality datasets such as IC15 and SVTP compared with other methods. Especially, SE-ASTER improve 3.9% on IC15 (from 76.1% to 80.0%) and 2.9% on SVTP (from 78.5% to 81.4%) compared with ASTER [45]. It also outperforms state-of-the-art method ScRN [53] 0.6% on SVTP and 1.3% on IC15, although our method is based on a weaker backbone and without character-level annotations.

SE-ASTER also gets superior or comparable results on several high-quality datasets. Compared with ASTER [45] we get 0.4% and 4.1% improvements on IIIT5K and CUTE respectively. On SVT and IC13, our method gets accuracies of 89.6% and 92.8%, which are slightly worse than ESIR [57] and [2] by 0.6% and 1.6%. Note that our framework is very flexible and can be integrated with most existing methods, and we believe that if we replace a stronger baseline model better results can be achieved.

5. Conclusion and Future Works

In this work, we propose the semantics enhanced encoder-decoder framework for scene text recognition. Our framework predicts an additional global semantic information supervised by the word embedding from a pre-trained language model. Using the predicted semantic information as the decoder initialization, the recognition accuracy can be improved especially for low-quality images. By integrating the state-of-the-art method ASTER into our framework, we can achieve superior results on several standard benchmark datasets. In the future, we will extend our framework to an end-to-end text spotting system. In this way, more semantic information can be utilized.

Acknowledgment

This work is supported by the National Key R&D Program of China (2017YFB1002400) and the Strategic Priority Research Program of Chinese Academy of Sciences (XDC02000000). In addition, we sincerely thank Mingkun Yang for his help.

References

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and

translate. *CoRR*, abs/1409.0473, 2014.

[2] Fan Bai, Zhanzhan Cheng, Yi Niu, Shiliang Pu, and Shuigeng Zhou. Edit probability for scene text recognition. In *CVPR*, pages 1508–1516, 2018.

[3] Xiang Bai, Cong Yao, and Wenyu Liu. Strokelets: A learned multi-scale mid-level representation for scene text recognition. *TIP*, 25(6):2789–2802, 2016.

[4] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *TACL*, pages 135–146, 2017.

[5] Fred L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *TPAMI*, 11(6):567–585, 1989.

[6] Yudi Chen, Yu Zhou, Dongbao Yang, and Weiping Wang. Constrained relation network for character detection in scene images. In *PRICAI*, pages 137–149, 2019.

[7] Zhanzhan Cheng, Fan Bai, Yunlu Xu, Gang Zheng, Shiliang Pu, and Shuigeng Zhou. Focusing attention: Towards accurate text recognition in natural images. In *ICCV*, pages 5076–5084, 2017.

[8] Zhanzhan Cheng, Yangliu Xu, Fan Bai, Yi Niu, Shiliang Pu, and Shuigeng Zhou. AON: Towards arbitrarily-oriented text recognition. In *CVPR*, pages 5571–5579, 2018.

[9] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *EMNLP*, pages 1724–1734, 2014.

[10] Suman K Ghosh, Ernest Valveny, and Andrew D Bagdanov. Visual attention models for scene text recognition. In *ICDAR*, pages 943–948, 2017.

[11] Suman K Ghosh, Ernest Valveny, et al. Don't only feel read: Using scene text to understand advertisements. *CoRR*, abs/1806.08279, 2018.

[12] Albert Gordo, Jon Almazán, Naila Murray, and Florent Perronin. LEWIS: Latent embeddings for word images and their semantics. In *ICCV*, pages 1242–1250, 2015.

[13] Alex Graves, Marcus Liwicki, Santiago Fernández, Roman Bertolami, Horst Bunke, and Jürgen Schmidhuber. A novel connectionist system for unconstrained handwriting recognition. *TPAMI*, 31(5):855–868, 2008.

[14] Ankush Gupta, Andrea Vedaldi, and Andrew Zisserman. Synthetic data for text localisation in natural images. In *CVPR*, pages 2315–2324, 2016.

[15] Pan He, Weilin Huang, Yu Qiao, Change Loy Chen, and Xiaoou Tang. Reading scene text in deep convolutional sequences. In *AAAI*, pages 3501–3508, 2016.

[16] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Reading text in the wild with convolutional neural networks. *IJCV*, 116(1):1–20, 2016.

[17] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *NeurIPS*, pages 2017–2025, 2015.

[18] Chulmoo Kang, Gunhee Kim, and Suk I Yoo. Detection and recognition of text embedded in online images via neural context models. In *AAAI*, pages 4103–4110, 2017.

[19] Dimosthenis Karatzas, Lluis Gomez-Bigorda, Anguelos Nicolaou, Suman Ghosh, Andrew Bagdanov, Masakazu Iwamura, Jiri Matas, Lukas Neumann, Vijay Ramaseshan Chandrasekhar, Shijian Lu, et al. ICDAR 2015 competition on robust reading. In *ICDAR*, pages 1156–1160, 2015.

[20] Dimosthenis Karatzas, Faisal Shafait, Seiichi Uchida, Masakazu Iwamura, Lluis Gomez i Bigorda, Sergi Robles Mestre, Joan Mas, David Fernandez Mota, Jon Almazan Almazan, and Lluis Pere De Las Heras. ICDAR 2013 robust reading competition. In *ICDAR*, pages 1484–1493, 2013.

[21] Praveen Krishnan, Kartik Dutta, and CV Jawahar. Word spotting and recognition using deep embedding. In *DAS*, pages 1–6, 2018.

[22] Chen-Yu Lee and Simon Osindero. Recursive recurrent nets with attention modeling for ocr in the wild. In *CVPR*, pages 2231–2239, 2016.

[23] Hui Li, Peng Wang, Chunhua Shen, and Guyu Zhang. Show, attend and read: A simple and strong baseline for irregular text recognition. In *AAAI*, pages 8610–8617, 2019.

[24] Minghui Liao, Baoguang Shi, Xiang Bai, Xinggang Wang, and Wenyu Liu. TextBoxes: A fast text detector with a single deep neural network. In *AAAI*, pages 4161–4167, 2017.

[25] Minghui Liao, Jian Zhang, Zhaoyi Wan, Fengming Xie, Jiajun Liang, Pengyuan Lyu, Cong Yao, and Xiang Bai. Scene text recognition from two-dimensional perspective. In *AAAI*, pages 8714–8721, 2019.

[26] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, pages 2980–2988, 2017.

[27] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. SSD: Single shot multibox detector. In *ECCV*, pages 21–37, 2016.

[28] Wei Liu, Chaofeng Chen, and Kwan-Yee K Wong. Char-Net: A character-aware neural network for distorted scene text recognition. In *AAAI*, pages 7154–7162, 2018.

[29] Yang Liu, Zhaowen Wang, Hailin Jin, and Ian Wassell. Synthetically supervised feature learning for scene text recognition. In *ECCV*, pages 435–451, 2018.

[30] Zichuan Liu, Yixing Li, Fengbo Ren, Wang Ling Goh, and Hao Yu. SqueezedText: A real-time scene text recognition by binary convolutional encoder-decoder network. In *AAAI*, pages 7194–7201, 2018.

[31] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440, 2015.

[32] Canjie Luo, Lianwen Jin, and Zenghui Sun. MORAN: A multi-object rectified attention network for scene text recognition. *PR*, 90:109–118, 2019.

[33] Anand Mishra, Karteek Alahari, and CV Jawahar. Scene text recognition using higher order language priors. In *BMVC*, pages 1–11, 2012.

[34] Lukáš Neumann and Jiří Matas. Real-time scene text localization and recognition. In *CVPR*, pages 3538–3545, 2012.

[35] Adam Paszke, Sam Gross, Francisco Massa, et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, pages 8026–8037. 2019.

[36] Yash Patel, Lluis Gomez, Marçal Rusinol, and Dimosthenis Karatzas. Dynamic lexicon generation for natural scene images. In *ECCV*, pages 395–410, 2016.

[37] Shitala Prasad and Adams Wai Kin Kong. Using object information for spotting text. In *ECCV*, pages 540–557, 2018.

[38] Xugong Qin, Yu Zhou, Dongbao Yang, and Weiping Wang. Curved text detection in natural scene images with semi-and weakly-supervised learning. In *ICDAR*, pages 559–564, 2019.

[39] Trung Quy Phan, Palaiahnakote Shivakumara, Shangxuan Tian, and Chew Lim Tan. Recognizing text with perspective distortion in natural scenes. In *ICCV*, pages 569–576, 2013.

[40] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NeurIPS*, pages 91–99, 2015.

[41] Anhar Risnumawan, Palaiahankote Shivakumara, Chee Seng Chan, and Chew Lim Tan. A robust arbitrary text detection system for natural scene images. *ESA*, 41(18):8027–8048, 2014.

[42] Ahmed Sabir, Francesc Moreno-Noguer, and Lluís Padró. Visual re-ranking with natural language understanding for text spotting. In *ACCV*, pages 68–82, 2018.

[43] Baoguang Shi, Xiang Bai, and Cong Yao. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *TPAMI*, 39(11):2298–2304, 2017.

[44] Baoguang Shi, Xinggang Wang, Pengyuan Lyu, Cong Yao, and Xiang Bai. Robust scene text recognition with automatic rectification. In *CVPR*, pages 4168–4176, 2016.

[45] Baoguang Shi, Mingkun Yang, Xinggang Wang, Pengyuan Lyu, Cong Yao, and Xiang Bai. ASTER: An attentional scene text recognizer with flexible rectification. *TPAMI*, 41(9):2035–2048, 2018.

[46] Bolan Su and Shijian Lu. Accurate recognition of words in scenes without character segmentation using recurrent neural network. *PR*, 63:397–405, 2017.

[47] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *NeurIPS*, pages 3104–3112, 2014.

[48] Zhi Tian, Weilin Huang, Tong He, Pan He, and Yu Qiao. Detecting text in natural image with connectionist text proposal network. In *ECCV*, pages 56–72, 2016.

[49] Kai Wang, Boris Babenko, and Serge Belongie. End-to-end scene text recognition. In *ICCV*, pages 1457–1464, 2011.

[50] Kai Wang and Serge Belongie. Word spotting in the wild. In *ECCV*, pages 591–604, 2010.

[51] Tomas Wilkinson and Anders Brun. Semantic and verbatim word spotting using deep neural networks. In *ICFHR*, pages 307–312, 2016.

[52] Zecheng Xie, Yaoxiong Huang, Yuanzhi Zhu, Lianwen Jin, Yuliang Liu, and Lele Xie. Aggregation cross-entropy for sequence recognition. In *CVPR*, pages 6538–6547, 2019.

[53] Mingkun Yang, Yushuo Guan, Minghui Liao, Xin He, Kaigui Bian, Song Bai, Cong Yao, and Xiang Bai. Symmetry-constrained rectification network for scene text recognition. In *ICCV*, pages 9147–9156, 2019.

[54] Xiao Yang, Dafang He, Zihan Zhou, Daniel Kifer, and C Lee Giles. Learning to read irregular text with attention mechanisms. In *IJCAI*, pages 3280–3286, 2017.

[55] Cong Yao, Xiang Bai, Baoguang Shi, and Wenyu Liu. Strokelets: A learned multi-scale representation for scene text recognition. In *CVPR*, pages 4042–4049, 2014.

[56] Matthew D Zeiler. Adadelta: An adaptive learning rate method. *CoRR preprint abs:1212.5701*, 2012.

[57] Fangneng Zhan and Shijian Lu. ESIR: End-to-end scene text recognition via iterative image rectification. In *CVPR*, pages 2059–2068, 2019.

[58] Xiaosong Zhang, Fang Wan, Chang Liu, Rongrong Ji, and Qixiang Ye. FreeAnchor: Learning to match anchors for visual object detection. In *NeurIPS*, pages 147–155, 2019.

[59] Yi Zheng, Qitong Wang, and Margrit Betke. Deep neural network for semantic-based text recognition in images. *CoRR*, abs/1908.01403, 2019.

[60] Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, and Jiajun Liang. EAST: An efficient and accurate scene text detector. In *CVPR*, pages 5551–5560, 2017.