RobustScanner: Dynamically Enhancing Positional Clues for Robust Text Recognition

Xiaoyu Yue[1][0000−0001−7023−0264], Zhanghui Kuang[1*][0000−0002−9102−5152], Chenhao Lin[2], Hongbin Sun[1][0000−0001−9183−4462], and Wayne Zhang[1][0000−0002−8415−1062]

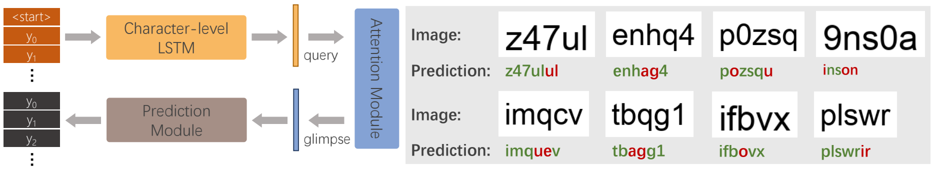
[1] SenseTime Research
{yuexiaoyu, kuangzhanghui, sunhongbin, wayne.zhang}@sensetime.com
[2] School of Cyber Science and Engineering, Xi'an Jiaotong University
linchenhao@xjtu.edu.cn

Abstract. The attention-based encoder-decoder framework has recently achieved impressive results for scene text recognition, and many variants have emerged with improvements in recognition quality. However, it performs poorly on contextless texts (*e.g.*, random character sequences) which is unacceptable in most of real application scenarios. In this paper, we first deeply investigate the decoding process of the decoder. We empirically find that a representative character-level sequence decoder utilizes not only context information but also positional information. Contextual information, which the existing approaches heavily rely on, causes the problem of attention drift. To suppress such side-effect, we propose a novel position enhancement branch, and dynamically fuse its outputs with those of the decoder attention module for scene text recognition. Specifically, it contains a position aware module to enable the encoder to output feature vectors encoding their own spatial positions, and an attention module to estimate glimpses using the positional clue (*i.e.*, the current decoding time step) only. The dynamic fusion is conducted for more robust feature via an element-wise gate mechanism. Theoretically, our proposed method, dubbed *RobustScanner*, decodes individual characters with dynamic ratio between context and positional clues, and utilizes more positional ones when the decoding sequences with scarce context, and thus is robust and practical. Empirically, it has achieved new state-of-the-art results on popular regular and irregular text recognition benchmarks while without much performance drop on contextless benchmarks, validating its robustness in both contextual and contextless application scenarios.

1 Introduction

Scene text recognition is crucial for visual understanding and reasoning in many application scenarios [44, 5]. Despite great progress recently, it remains a challenging task because the potential irregularity and diversity of text shapes and

[*] Zhanghui Kuang is the corresponding author.



(a) Decoding procedure

(b) Failure examples

Fig. 1. Illustration of the decoding procedure of encoder-decoder with attention framework and its failure cases.

layouts in the wild, which can be curved, oriented or distorted, make the misalignment between the output character sequence and the two-dimensional input image [29, 57, 50, 25, 42]. The prevalent approaches for scene text recognition are inspired by machine translation [46, 30, 2] and image caption [54] following the encoder-decoder framework with varied attention mechanisms [39, 50, 25, 56, 42].

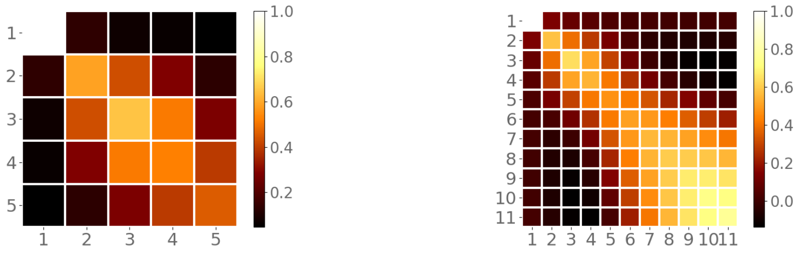
Typically, the encoder-decoder with attention framework [25, 56] consists of one encoder, and one decoder. The decoder contains one character-level LSTM, one attention module, and one prediction module. During decoding, at each step, the LSTM takes the previously predicted character and the hidden state as inputs, and outputs one *query* feature vector, which is fed into the attention module to estimate attention map and compute one *glimpse* feature vector. The glimpse vector is finally classified into one character category or the <EOS> token in the prediction module as illustrated in Figure 1 (a). In spite of the appealing effectiveness of the encoder-decoder with attention framework on academic benchmarks, it performs poorly on contextless text sequence images, which hinders it from being widely used in real application scenarios. To demonstrate it, we synthesize one benchmark of random character sequence images, dubbed *RandText*, to evaluate existing encoder-decoder with attention based methods. Surprisingly, the official released model of the state-of-the-art method SAR [25] obtains a low accuracy of 59.6% on it. Figure 1 (b) shows some failure samples, all of which are without background clutter, low resolutions, varied illumination or distortions, and thus considered as easy cases by human.

To explore the underlying reasons of the failure recognition, in this work, we deeply investigate the decoding process of encoder-decoder with attention based methods. By computing the averaged cosine similarity between the *query* feature vectors of the i^{th} and j^{th} time steps on ICDAR 2013 [20] test set, we observed the high averaged similarity between those of the same time step (as shown in Figure 2) although their corresponding characters are different. The observation suggests that the *query* vectors encode not only context information but also the positional information. We also observed that the averaged similarity between the *query* vectors of neighborhood time steps increases as the time step increases, which suggests that the positional information is drowned with the introduction of others including context information at latter time steps. It can easily lead to alignment drift and misrecognition of latter characters of contextless sequences, which coincides with most of the failure samples in Figure 1 (b).

To mitigate the above misrecognition, in this paper, we propose RobustScanner for text recognition via dynamically enhancing the positional clues of the decoder. Specifically, besides the conventional decoder, it consists of one position enhancement branch and one dynamic fusion module. The former is tailored for enhancing the conventional decoder in terms of positional encoding capability via estimating the *glimpses* with positional clues (*i.e.*, the current decoding time step) only. The latter is designed to dynamically fuse the *glimpses* of the position enhancement branch, and those of the conventional decoder via one element-wise gate mechanism. In such a way, the fusion ratio between positional and context information at each time step can be dynamically and adaptively adjusted according to the their own importance.

Our **contributions** can be summarized as follows:

- We investigate the intrinsic mechanism of the decoding procedure of the encoder-decoder with attention framework for the first time. We find out that the *query* feature vectors of the LSTM encodes not only context but also positional information, and the context information dominates the *query* at latter decoding time steps, which can lead to misrecognition on contextless text images.
- We propose RobustScanner to mitigate the issue of misrecognition in contextless scenarios via introducing a novel position enhancement branch and a dynamic fusion module.
- We extensively evaluate our proposed RobustScanner, which achieves new state-of-the-art performances on popular regular and irregular text recognition benchmarks while without much performance drop on contextless *RandText*, validating its robustness in both context and contextless application scenarios.



(a) $l = 5$

(b) $l = 11$

Fig. 2. Visualization of the averaged similarity matrix. The x-axis and the y-axis indicate the position index in sequences, while the color indicate the averaged similarity. The block at (i, j) indicates $S_l(i, j)$. (a) visualizes the averaged similarity matrix when $l = 5$ while (b) $l = 11$.

2 Related Work

Most of traditional methods for scene text recognition [48, 49, 33, 23, 34] adopt the bottom-up approach in which individual character is first detected by sliding window, and then integrated for taking the dependence with its neighbors into consideration. These methods might fail to detect small characters, and are easily disturbed by background clutter, illumination, and low image quality, *e.g.*, blurring, noise, *etc*. Later, top-down methods [12, 40, 15, 11, 24, 41, 3, 42, 57, 8, 7, 25, 50, 35, 1, 55, 45, 16] were proposed, in which text sequences are end-to-end predicted without the single character detection. Recently, approaches which target at challenging arbitrary-shaped text recognition become dominant. These approaches can be roughly categorized into rectification-based, segmentation-based and encoder-decoder with attention-based.

Rectification-based approaches. They attempt to rectify irregular images to regular ones before recognition. STN [18] was first introduced into text recognition network by Shi *et al* [41], which was extended by considering more flexible Transformation Thin-Plate-Spline (TPS) in [42, 55] and conducting rectification iteratively in [57]. Rather than rectifying the whole text region, Liu *et al* [27] detected individual characters first and then rectified them separately. The shortness of this strategy is that, if some important information missed during rectification process, it cannot be compensated in the latter part of the network, which may be fatal at times.

Segmentation-based approaches. They segment each character individually to avoid the issue of irregular layout. Liao *et al* [26] employed fully convolutional network to detect and recognize individual characters followed by character-combination in post-processing. However, they need character-level annotations which are unavailable in most public datasets. Xing *et al* [53] alleviated this issue by an iterative character detection, which can transform the ability of character detection learned from synthetic data to real-world images. All the segmentation-based approaches cannot trivially extend to the text recognition with huge character dictionary, such as Chinese recognition, as they maintain one probability heat map for each character.

Encoder-decoder with attention-based approaches. Most of existing state-of-the-art methods for irregular text recognition follow the encoder-decoder framework with attention mechanisms [41, 56, 7, 27, 25, 56, 35, 1, 45, 50]. The encoder-decoder with attention framework was first proposed for NLP tasks such as machine translation [46, 30, 2]. Later, it was introduced into scene text recognition to align the character in the output sequence with local image regions in [41]. Since then, many variants [7, 25, 56] were emerged with improvements in recognition quality. Cheng *et al* [7] introduced a focus network to suppress the attention drift problem. SAR [25] employed a tailored 2D attention mechanism to recognize irregular texts and achieved impressive results. Recently, [50, 39] introduced Transformer to replace RNN structure to capture long distance context. Wang *et al* [51] proposed Decoupled Attention Network (DAN) to mitigate the alignment drift problem. Our approach falls into the encoder-decoder with attention framework. Similar to DAN [51], we also target at suppressing the misrecognition caused by alignment drift. Instead of decoupling attention from historical decoding results as done in [51], we propose the position enhancement branch and dynamically fusion module to adaptively adjust the ratio of positional and context clues during decoding. Besides, we deeply investigate the intrinsic mechanism of the decoding procedure in the decoder for the first time.

Our approach is related to the positional encoding model [47, 9, 38]. We do not target at enumerating all possible architectures and finding the optimal one of positional encoding model. In contrast, we focus on enhancing positional clues of the attention-based decoder.

3 Methodology

In this section, we first review the encoder-decoder with attention-based scene text recognition, and then dissect the decoder and analyze what information the *query* feature vectors contain. Finally, we introduce our proposed approach RobustScanner.

3.1 Background

As [25, 56], one representative encoder-decoder with attention-based scene text recognition approach consists of one encoder and one decoder. The decoder has one LSTM-based sequence module, one attention module and one prediction module. Given one input image, the encoder extracts one feature map $\mathbf{F} \in \mathbb{R}^{H \times W \times C}$. During decoding, at time step t, the LSTM-based sequence model first generates one hidden feature vector $\mathbf{h_t}$. Formally,

$$\mathbf{h_t} = \text{LSTM}(\mathbf{x_t}, \mathbf{h_{t-1}}), \mathbf{x_t} = \begin{cases} \mathbf{y_{t-1}} & \text{if } t > 1 \\ \text{<start>} & \text{if } t = 1 \end{cases}, \tag{1}$$

where <start> is a special start token, and $\mathbf{y_{t-1}}$ is the output of decoding process at time step $t-1$. $\mathbf{h_t}$ is considered as the *query* feature vector of the attention module to compute one attention map as follows:

$$\alpha_{ij}^{t} = \text{softmax}({\mathbf{h_t}}^{T} \mathbf{f_{i,j}}), \tag{2}$$

where ${\mathbf{h_t}}^{T}$ indicates the transpose of the vector $\mathbf{h_t}$, and $\mathbf{f_{i,j}}$ represents the feature vector at the position (i, j) of $\mathbf{F}$. The *glimpse* vector $\mathbf{g_t}$ is then computed by the weighted aggregation of the convolutional feature map $\mathbf{F}$ as follows:

$$\mathbf{g_t} = \sum_{ij} \alpha_{ij}^{t} \mathbf{f_{i,j}}. \tag{3}$$

Finally, the glimpse vector $\mathbf{g_t}$ is classified into one character or the <EOS> token. Formally,

$$\mathbf{y_t} = \text{softmax}(\mathbf{W} \mathbf{g_t} + b), \tag{4}$$

where $\mathbf{W}$ and b indicate the linear transformation and the bias of the classifier respectively.

3.2 Decoder Dissection

From Equation (1), (2), (3), and (4), we conclude that given the feature map $\mathbf{F}$ and learned network parameters, the recognized character $\mathbf{y_t}$ depends on the *query* vector $\mathbf{h_t}$ only. What information does the *query* vector $\mathbf{h_t}$ encode so that its corresponding attention weight α^t can highlight and the classifier can correctly recognize the t^{th} character on the input image? Obviously, the *query* vector contains context information since it depends on previously-predicted characters from Equation (1).

We observe that the *query* vector $\mathbf{h_1}$ keeps unchanged for different text sequences, and does not encode any context information. However, the first character can still be correctly recognized. This implies that the *query* vector $\mathbf{h_t}$ (including $\mathbf{h_1}$) contains the positional information. *i.e.*, the character index in one sequence.

To verify the above conjecture, we analyze the similarity of the *query* for different text sequences at the same time step. We conduct experiments on test set of ICDAR 2013 [20]. Let $\mathcal{I}$ denote the set of all text sequences, and $\widehat{\mathcal{I}_l}$ be the subset of $\mathcal{I}$ with all text sequences of length l. We compute the average cosine similarity $S_l(i,j)$ between the *query* vectors at position i and j over $\widehat{\mathcal{I}_l}$ as follows:

$$S_l(i,j) = \frac{\sum_{m \neq n}^{\left|\widehat{\mathcal{I}_l}\right|} \cos(\mathbf{h_i^{m,l}}, \mathbf{h_j^{n,l}})}{\left|\widehat{\mathcal{I}_l}\right| (\left|\widehat{\mathcal{I}_l}\right| - 1)}, \tag{5}$$

where $\left|\widehat{\mathcal{I}_l}\right|$ denotes the cardinality of $\mathcal{I}_l$, and $\mathbf{h_i^m}$ denotes the i^{th} *query* vector for the m^{th} sequence in $\mathcal{I}_l$.

Figure 2 visualizes the averaged similarity matrixes when $l = 5$ and $l = 11$. We have two observations:

- Averaged similarities between the *query* vectors of the same position are obviously higher than those of different positions.
- As the time step increases, the contrast between the averaged similarities between the *query* vectors of the same position, and those of neighbor positions become smaller.

Note that the i^{th} character of sequences in $\mathcal{I}_l$ varies from one sequence to another. High value of $S_l(i,i)$ suggests queries can be well separated into groups corresponding to their steps using kernel methods. We further conduct the linear regression between queries ($\mathbf{h_t}$) and positions (t) via fitting $t = \mathbf{W_r}\mathbf{h_t} + b_r$ on the data with $l = 5$ (90% for training, 10% for test). We report the classic R-squared R^2 which indicates the proportion of the variance of the dependent variable that is predictable. We get $R^2 = 0.994$ and $R^2 = 0.956$ on training and test set. High R^2 scores prove queries can reliably predict positions. Thus, queries encode them. The second observation suggests that the positional information is drowned with the introduction of others including the context information.

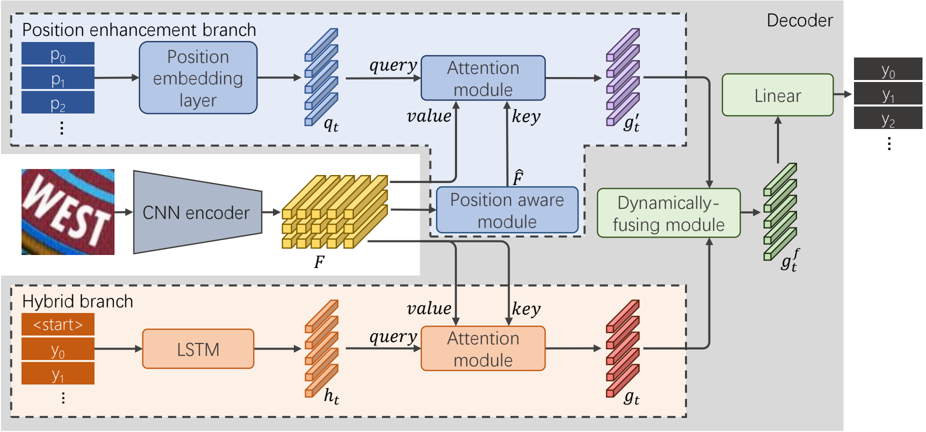


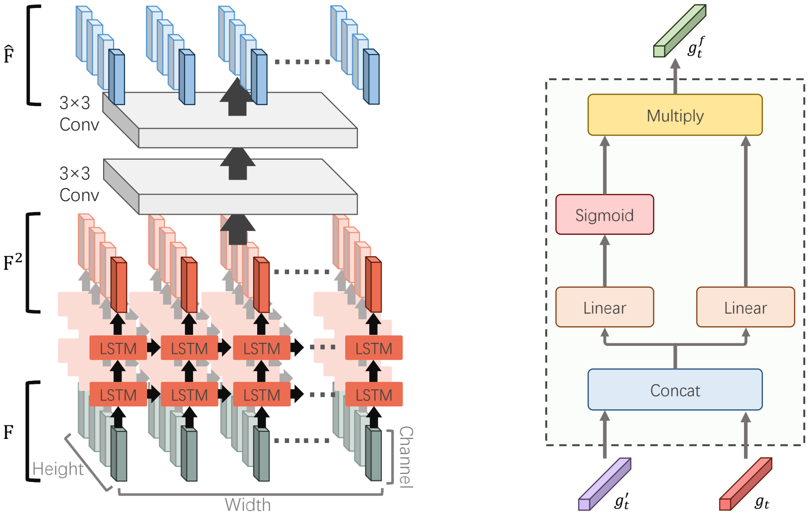
Fig. 3. Architecture of RobustScanner. Given an input image, we first extract its feature map $\mathbf{F}$ by the CNN encoder. $\mathbf{F}$ is then fed into the hybrid branch, and the position enhancement branch, obtaining the *glimpses* $\mathbf{g_t}$ and $\mathbf{g'_t}$, which are dynamically fused by the dynamically-fusing module before predicting the t^{th} character.

3.3 RobustScanner

Overview. As discussed in Section 3.2, the *query* vectors in the encoder-decoder with attention based framework contain the hybrid information of context and position. The positional clues become weaker while the contextual ones become stronger as the time step increases during decoding, which may lead to alignment drift and misrecognition, especially on contextless text images. To enhance the positional information, we propose one position enhancement branch and dynamically fuse its outputs with those of the conventional decoder. As shown in Figure 3, our RobustScanner consists of one encoder and one decoder. In the encoder, we adapt one 31-layer ResNet [14] as backbone as done in [25]. The decoder consists of one hybrid branch, one position enhancement branch, one dynamically-fusing module, and one prediction module.

Hybrid Branch. The hybrid branch consists of one two-layer LSTM with 128 hidden state size and one attention module. The LSTM takes the previously-predicted character and its hidden variable as input and generates the *query* vector $\mathbf{h_t}$. Then the *query* vector is fed into the attention module to estimate *glimpse* vector $\mathbf{g_t}$ for the character prediction during decoding (see Equation (1) and (3)). It utilizes both contextual and positional information simultaneously.

Position Enhancement Branch. The positional information becomes weak while the contextual information becomes strong in the latter time steps during decoding. It would lead to the alignment drift and serious misrecongition on contextless text images where context cannot be used to reliably predict characters. Our position enhancement branch is designed to mitigate this problem. It consists one position embedding layer, one position aware module and one attention module.



(a) Position aware module (b) Dynamically-fusing module

Fig. 4. Architecture of the position aware module and that of the dynamically-fusing module. (a) the position aware module inputs $\mathbf{F}$, then obtains $\mathbf{F^2}$ after two LSTM layers, and finally outputs $\widehat{\mathbf{F}}$ after two 3×3 convolutional layers. (b)the dynamically-fusing module takes $\mathbf{g_t}$ and $\mathbf{g_t'}$ as inputs and output one adaptively fused glimpse $\mathbf{g_t^f}$.

The position embedding layer encodes the decoding time step. It inputs one hot vectors and outputs one specific embedding vector $\mathbf{q_t}$ for each time step t. Note that the embedding vector of each time step keeps unchanged across different decoding sequences.

Position Aware Module. Although the position embedding layer encodes the character index in sequences, precisely locating the t^{th} character on the encoder output feature map $\mathbf{F}$, which requires the global information, is still challenging. However, the width of input image might be beyond the receptive field size of the CNN encoder. Moreover, counting the indexes of characters in images needs to understand the shapes and directions of the text sequences. To this end, we propose a position aware module to capture the global and high-level information so that the encoder output feature map is position-aware. Figure 4 (a) shows the detailed architecture of the proposed position aware module. We employ two-layer LSTM with 128 hidden state size for each row of the feature map $\mathbf{F}$ to capture the global context. For all rows, the LSTMs share parameters to overcome overfitting and reduce parameter amount. We then employ two 3×3 convolutional layers with one ReLU between them to generate the *key* vectors $\widehat{\mathbf{F}}$ of its following attention module. Formally,

$$\mathbf{f^1_{i,j}} = \mathrm{LSTM}(\mathbf{f_{i,j}}, \mathbf{f^1_{i,j-1}}), \tag{6}$$

$$\mathbf{f^2_{i,j}} = \mathrm{LSTM}(\mathbf{f^1_{i,j}}, \mathbf{f^2_{i,j-1}}), \tag{7}$$

$$\widehat{\mathbf{F}} = f(\mathbf{F^2}), \tag{8}$$

where $\mathbf{f^1_{i,j}}$ and $\mathbf{f^2_{i,j}}$ are the first and the second LSTM hidden variables respectively. $\mathbf{F^2}$ is the tensor consisting of $\mathbf{f^2_{i,j}}$, and $f(\cdot)$ is the stack of 3×3 *conv*, ReLU, and 3×3 *conv* operations.

We feed the output vectors $\mathbf{q_t}$ of the embedding layer into the attention module as *key* vectors to estimate attention maps, which are used to compute the final *glimpse* vectors $\mathbf{g'_t}$ as done in Equation (2) and (3). Different from conventional methods, we use the position aware feature map $\widehat{\mathbf{F}}$ instead of $\mathbf{F}$ when computing the attention maps.

Note that position aware module injects the global context so that it has the capability to output position aware features. The query vectors in the position enhancement branch explicitly encode the character order indexes (positions). The end-to-end training enforces the position aware module to output feature maps correlated with positions so that the attention module in the branch can correctly output the feature glimpse at each decoding step (position) with the position aware module output feature maps and position-specific query vector as inputs (see Figure 3). Otherwise, texts cannot be correctly decoded.

Dynamically-Fusing Module. As shown in Figure 4 (b), we propose to dynamically fuse the hybrid branch output $\mathbf{g_t}$ and the position enhancement branch output $\mathbf{g'_t}$ at each time step t. We design one gate mechanism to predict one attention weight for each dimension of their concatenation, which is used to enhance or suppress their corresponding feature. Formally,

$$\mathbf{w_t} = \text{sigmoid}(\mathbf{W_a}[\mathbf{g_t};\mathbf{g'_t}]), \tag{9}$$

$$\mathbf{g^f_t} = \mathbf{w_t} \odot (\mathbf{W_p}[\mathbf{g_t};\mathbf{g'_t}]), \tag{10}$$

where $\mathbf{W_a}$ and $\mathbf{W_p}$ are two learned linear transformations. $\odot$ indicates the element-wise multiplication operation. $\mathbf{g^f_t}$ is the final output of our dynamically-fusion module, which is used to predict the character via the prediction module.

4 Experiments

4.1 Datasets

For fair comparison with previous state-of-the-art approaches, we follow their settings to train the proposed network by using two public available synthetic datasets, *i.e.* MJSynth [17] and SynthText [13]. We conduct extensively experiments on 6 standard benchmarks including 3 regular text datasets (IIIT 5K-words [32], Street View Text [48], ICDAR 2013 [20]), 3 irregular text datasets (ICDAR 2015 [19], Street View Text Perspective [36], CUTE 80 [37]).

IIIT 5K-words (IIIT5K) [32] is a large dataset containing 5000 word patches cropped from natural scene images collected from Google image search, in which 2000 images are used for training and 3000 for test. Text instances in these patches are regular with the horizontal layout.

Street View Text (SVT) [48] consists of 647 word patches cropped from Google Street View for testing. Most images are horizontal, but severely corrupted by noise and blur, or with low-resolution.

ICDAR 2013 (IC 2013) [20] has 848 cropped word patches for training and 1095 for test. For fair comparison with other reported results, we discard images that contain non-alphanumeric characters, which results in 1015 test patches. Words in this dataset are mostly regular.

ICDAR 2015 (IC 2015) [19] contains word patches cropped from incidental scene images captured under arbitrary angles. Hence most word patches in this dataset are irregular (oriented, perspective or curved). It contains 4468 patches for training and 2077 for test.

Street View Text Perspective (SVTP) [36] consists of 639 word patches, which are cropped from side view snapshots in Google Street View and encounter severe perspective distortions. All patches are used for test.

CUTE 80 [37] contains 288 cropped high resolution images for test and many of them are curved irregular text images.

RandText contains 500 test images. We synthesize them by pasting black random character sequences on white background images. The character set contains small and capital letters, and numbers. Some examples can be found in Figure 1 (b). We will publicly release RandText to facilitate the future research.

4.2 Implementation Details

The proposed framework is implemented by using PyTorch. All experiments are conducted on servers with 4 NVIDIA Titan X GPUs with 12 GB memory.

Our RobustScanner is trained from scratch using Adam optimizer [21] with the base learning rate 1×10^{-3}. The whole training process contains 5 epochs, while the learning rate decreases to 1×10^{-4} at the 3^{rd} epoch and 1×10^{-5} at the 4^{th} epoch. The batch size at training phase is set to 128. For both training and test, heights of all image patches are set to 48, while widths are proportionally scaled with heights, but no longer than 160 and no smaller than 48 pixels. Our models recognizes 91 token classes, including 10 digits, 52 case sensitive letters, 28 punctuation characters and an <EOS> token.

To reduce the computation time of two attention module, the dimension of the hybrid query vector h_t and the position embedding q_t is set to 128. Before feeding F to the hybrid attention module and the position aware module, a 1×1 *conv* layer is added to reduce the channel dimension to 128. The maximum number of position embeddings is set to 36.

In the test stage, for images with height larger than width, we will rotate the image by 90 degree clockwise and anticlockwise respectively, and recognize them together with the original image. For the recognition results of each 3-sibling image group, the top-score one will be chosen as the final recognition result.

Table 1. Comparison with state-of-the-art methods. "MJ", "ST" and "R" are the training data of MJSynth [17], SynthText [13] and training splits of real datasets, respectively.

Method	Training Data	Regular Text			Irregular Text		
		IIIT5K	SVT	IC 2013	IC 2015	SVTP	CUTE 80
Shi *et al* [42]	MJ + ST	93.4	**93.6**	91.8	76.1	78.5	79.5
Zhan and Lu [57]	MJ + ST	93.3	90.2	91.3	76.9	79.6	83.3
Gao *et al* [10]	MJ + ST	94.0	88.6	93.2	**77.1**	80.6	88.5
Bai *et al* [3]	MJ + ST	88.3	87.5	94.4	73.9	-	-
Luo *et al* [29]	MJ + ST	91.2	88.3	92.4	68.8	76.1	77.4
Wang *et al* [50]	MJ + ST	93.3	88.1	91.3	74.0	80.2	85.1
Lyu *et al* [31]	MJ + ST	94.0	90.1	92.7	76.3	82.3	86.8
Xie *et al* [52]	MJ + ST	82.3	82.6	89.7	68.9	70.1	82.6
DAN [51]	MJ + ST	94.3	89.2	93.9	74.5	80.0	84.4
Bartz *et al* [4]	MJ + ST	94.6	89.2	93.1	74.2	**83.1**	89.6
Bleeker *et al* [6]	MJ + ST	94.7	89.0	93.4	75.7	80.6	82.5
Long *et al* [28]	MJ + ST	93.7	88.9	92.4	76.6	78.8	86.8
Baek *et al* [1]	MJ + ST	87.9	87.5	92.3	71.8	79.2	74.0
RobustScanner	MJ + ST	**95.3**	88.1	**94.8**	**77.1**	79.5	**90.3**
SAR [25]	MJ + ST + R	95.0	**91.2**	94.0	78.8	**86.4**	89.6
RobustScanner	MJ + ST + R	**95.4**	89.3	**94.1**	**79.2**	82.9	**92.4**

4.3 Comparison with State-of-the-art Approaches

We compare our approach with previous state-of-the-art methods in Table 1. It has been shown that our proposed RobustScanner achieves best results on four datasets including IIIT5K, ICDAR 2013, ICDAR 2015 and CUTE 80 when only synthetic training datasets are used. To demonstrate the potential of our proposed method, we further train it on MJSynth, SynthText, and training sets of real datasets in Section 4.1 following the training setting of SAR [25]. We have observed that RobustScanner outperforms its competitor SAR on four out of six benchmarks. Especially, it obtains the accuracy of 92.4%, and outperforms SAR with impressive margins on the challenging irregular text dataset CUTE 80. It can be noticed that our method performs worse than SAR [25] on two street view text datasets, we attribute the performance gap to that SVT and SVTP consist of contextual words and suffer from low resolution, blur and distortion, where context is more important than positions when recognizing texts.

Table 2. Comparison on RandText. All methods are trained on the training data of MJSynth [17] and SynthText [13] for fair comparison.

Method	SAR [25]	DAN [51]	Wang *et al* [50]	RobustScanner
Accuracy	59.6	76.4	78.8	**81.2**

This paper targets at mitigating the misrecongition issue on contextless text images of the encoder-decoder with attention based framework. Therefore, we evaluate our proposed RobustScanner on RandText dataset. Table 2 compared it with representative encoder-decoder with attention based approaches. Again, our proposed RobustScanner obviously outperforms its counterparts including LSTM based methods [25] and [51] and the Transformer based method [50]. Especially, DAN [51] decouples the historical decoded characters from the attention estimation to suppress the alignment drift problem. However, it still performs worse than our RobustScanner by absolute 4.8%. The effectiveness of RobustScanner on both context and contextless benchmarks validate its robustness in real application scenarios.

4.4 Ablation Study

The effectiveness of each branch. To show the effectiveness of each branch, we evaluate our RobustScanner without the position enhancement branch or the hybrid branch in Table 3. In these experiments, the *glimpse* vectors are fed directly into the classifier to decode the character at each time step without dynamic fusion. We have two observations. First, RobustScanner greatly outperforms its counterpart without the hybrid branch on all academical regular and irregular benchmarks, and performs slightly worse on RandText. Due to the collection bias, all academical benchmarks have abundant context. It suggests that the context clues introduced via the hybrid branch can boost the recognition performances on context text images, and causes side effect on contextless text images. Second, the performance of RobustScanner drops drastically when without applying the position enhancement branch on both context and contextless text images. Particularly, RobustScanner without the position enhancement branch obtains a low accuracy of 46.8% on RandText, which is lower than it with both branches by absolutely 34.4%. We conclude that the capacity of encoding positional clues in the decoder with attention should be enhanced, especially when recognizing contextless text images.

Table 3. Evaluation of the effectiveness of each branch in RobustScanner. HB and PEB indicate the hybrid branch and the position enhancement branch respectively.

	IIIT5K	SVT	IC 2013	IC 2015	SVTP	CUTE 80	RandText
w/o HB	92.5	84.4	91.4	70.7	74.4	83.3	**83.4**
w/o PEB	91.0	**88.3**	91.7	72.0	75.4	84.0	46.8
RobustScanner	**95.3**	88.1	**94.8**	**77.1**	**79.5**	**90.3**	81.2

The effectiveness of the position aware module. Enhancing position encoding is nontrivial since it requires the global context of the input image and the comprehension of the shapes and directions of text sequences. Our proposed position aware module is the core component of the position enhancement branch. Table 4 evaluates its effect. if we take out the position aware module, our RobustScanner decreases from 92.5%, 84.4%, 91.4%, 70.7%, 74.4%, 83.3% and 83.4% to 88.7%, 84.1%, 89.0%, 61.8%, 62.3%, 76.7% and 68.0% on IIIT5K, SVT, ICDAR 2013, ICDAR 2015, SVTP, CUTE 80 and RandText respectively. We also compare our position aware module with sine and cosine positional encoding [47]. We compute the 128-dimensional sine and cosine positional encoding via $pe(pos, 2i) = \sin(pos/1000^{2i/128})$ and $pe(pos, 2i+1) = \cos(pos/1000^{2i/128})$ with pos being the horizontal coordinate of the current encoder output feature vector in the spatial space, and $2i$ and $2i+1$ being the encoding indexes. We pad the sine and cosine positional encoding with $\mathbf{F}$ and obtain $\widehat{\mathbf{F}}$, which is fed into the attention module as *key* vectors. As shown in Table 4, sine and cosine positional encoding leads to obvious performance drop. Arguably, it is because since and cosine positional encoding contain absolute coordinate information only while relative character indexes in sequences are required in the position enhancement branch.

Table 4. Evaluation of the effectiveness of the position aware module, and other position encoding method. PAM indicates the position aware module. sine&cosine indicates padding sine and cosine positional encoding with the encoder output feature maps.

	IIIT5K	SVT	IC 2013	IC 2015	SVTP	CUTE 80	RandText
w/o PAM	88.7	84.1	89.0	61.8	62.3	76.7	68.0
sine&cosine	86.1	83.2	87.8	57.5	56.1	74.0	49.0
RobustScanner	**92.5**	**84.4**	**91.4**	**70.7**	**74.4**	**83.3**	**83.4**

The effectiveness of the dynamically-fusing module. Table 5 compares our proposed dynamically-fusing module with other baseline fusion methods. Namely, element-wise addition and concatenation. They element-wise addition or concatenate the *glimpse* vectors $\mathbf{g_t}$ and $\mathbf{g'_t}$ from the hybrid branch and the position enhancement branch before feeding them into the classifier at the t^{th} time step during decoding. Different from dynamically-fusing, they are static and keep unchanged across different time steps and different decoding sequences. Our proposed dynamically-fusing module achieves best results on all 6 academical benchmarks and the second best result on RandText. Therefore, it is more robust than its static competitors, and applicable in more general application scenarios including those with or without context.

Performance on license plate recognition. To verify the effectiveness of our RobustScanner on contextless data, we conduct experiments on the license plate recognition task. We crop 97 license plate images from Cars dataset [22] as the test set by using the character-level training annotations introduced in [43]. Noted that all methods are trained on the training data of MJSynth [17] and SynthText [13]. Therefore, the training and test data are very different in terms of background, font, and layout. All methods get relatively low accuracy on it. However, our RobustScanner outperforms other encoder-decoder with attention

Table 5. Comparison with different fusion methods.

Method	IIIT5K	SVT	IC 2013	IC 2015	SVTP	CUTE 80	RandText
Element-wise addition	94.8	87.9	94.3	75.5	78.6	89.2	**82.4**
Concatenation	95	**88.1**	94.2	75.2	78.6	88.2	80.4
Dynamically-fusing	**95.3**	**88.1**	**94.8**	**77.1**	**79.5**	**90.3**	81.2

Table 6. Comparison on the license plate dataset.

Method	SAR [25]	DAN [51]	Wang *et al* [50]	RobustScanner
Accuracy	29.9	51.1	46.4	**55.7**

based methods with impressive margins, as shown in Table 6. Figure 5 shows some results of our method and SAR [25] on the license plate test set.

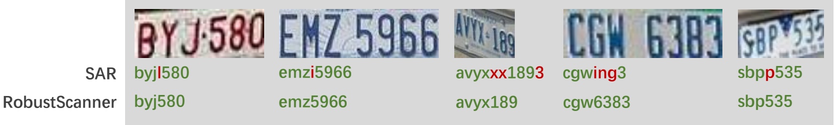


Fig. 5. Samples of recognition results of our RobustScanner and SAR [25] on the license plate dataset.

5 Conclusions

In this paper, we target at mitigating the misrecognition problem of the encoder-decoder with attention framework on contextless text images. We have investigated the decoding procedure and have found that the *query* vectors of attention in the decoder contains both positional and context information, and positional clues become weak while contextual ones become strong as the decoding time step increases. Motivated by this finding, we have proposed RobustScanner for scene text recognition, which contains one hybrid branch and one position enhancement branch, and dynamically fuses the two branches at each time step during decoding. Moreover, we have proposed one novel position aware module, which can strengthen the positional encoding capacity of the position enhancement branch. We have extensively evaluated our proposed RobustScanner on both academical benchmarks and our synthesized RandText. The experimental results show that our approach has achieved new state-of-the-art results on popular regular and irregular text benchmarks while without much performance drop on contextless benchmarks. It has been validated that our proposed RobustScanner is robust in both context and contextless application scenarios.

References

1. Baek, J., Kim, G., Lee, J., Park, S., Han, D., Yun, S., Oh, S.J., Lee, H.: What is wrong with scene text recognition model comparisons? dataset and model analysis. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 4715–4723 (2019)
2. Bahdanau, D., Cho, K., Bengio, Y.: Neural Machine Translaton by Jointly Learning to Align And Translate. In: ICLR. pp. 1–15 (2015)
3. Bai, F., Cheng, Z., Niu, Y., Pu, S., Zhou, S.: Edit Probability for Scene Text Recognition. In: CVPR. pp. 1508–1516 (2018)
4. Bartz, C., Bethge, J., Yang, H., Meinel, C.: Kiss: Keeping it simple for scene text recognition. arXiv preprint arXiv:1911.08400 (2019)
5. Biten, A.F., Tito, R., Mafla, A., Gomez, L., Rusinol, M., Valveny, E., Jawahar, C., Karatzas, D.: Scene text visual question answering. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 4291–4301 (2019)
6. Bleeker, M., de Rijke, M.: Bidirectional scene text recognition with a single decoder. arXiv preprint arXiv:1912.03656 (2019)
7. Cheng, Z., Bai, F., Xu, Y., Zheng, G., Pu, S., Zhou, S.: Focusing Attention: Towards Accurate Text Recognition in Natural Images. In: ICCV. pp. 5076–5084 (2017)
8. Cheng, Z., Xu, Y., Bai, F., Niu, Y., Pu, S., Zhou, S.: AON: Towards Arbitrarily-Oriented Text Recognition. In: CVPR. pp. 5571–5579 (2018)
9. Dai, Z., Yang, Z., Yang, Y., Cohen, W.W., Carbonell, J., Le, Q.V., Salakhutdinov, R.: Transformer-xl: Attentive language models beyond a fixed-length context. arXiv preprint arXiv:1901.02860 (2019)
10. Gao, Y., Chen, Y., Wang, J., Lei, Z., Zhang, X.Y., Lu, H.: Recurrent calibration network for irregular text recognition. arXiv preprint arXiv:1812.07145 (2018)
11. Gao, Y., Chen, Y., Wang, J., Lu, H.: Reading Scene Text with Attention Convolutional Sequence Modeling. arXiv preprint arXiv:1709.04303 (2017)
12. Graves, A., Fernández, S., Gomez, F., Schmidhuber, J.: Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In: Proceedings of the 23rd International Conference on Machine Learning. pp. 369–376. ACM (2006)
13. Gupta, A., Vedaldi, A., Zisserman, A.: Synthetic data for text localisation in natural images. In: CVPR (2016)
14. He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition. In: CVPR. pp. 770–778 (2016)
15. He, P., Huang, W., Qiao, Y., Loy, C.C., Tang, X.: Reading scene text in deep convolutional sequences. In: AAAI (2016)
16. He, W., Yin, F., Zhang, X.y., Liu, C.l.: TextDragon : An End-to-End Framework for Arbitrary Shaped Text Spotting. In: ICCV. pp. 9076–9085 (2019)
17. Jaderberg, M., Simonyan, K., Vedaldi, A., Zisserman, A.: Synthetic data and artificial neural networks for natural scene text recognition. arXiv preprint arXiv:1406.2227 (2014)
18. Jaderberg, M., Simonyan, K., Zisserman, A., Others: Spatial transformer networks. In: NIPS. pp. 2017–2025 (2015)
19. Karatzas, D., Gomez-Bigorda, L., Nicolaou, A., Ghosh, S., Bagdanov, A., Iwamura, M., Matas, J., Neumann, L., Chandrasekhar, V.R., Lu, S., Others: ICDAR 2015 Competition on Robust Reading. In: ICDAR. pp. 1156–1160. IEEE (2015)
20. Karatzas, D., Shafait, F., Uchida, S., Iwamura, M., Bigorda, L.G.I., Mestre, S.R., Mas, J., Mota, D.F., Almazan, J.A., De Las Heras, L.P.: ICDAR 2013 robust reading competition. ICDAR pp. 1484–1493 (2013)

21. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
22. Krause, J., Stark, M., Deng, J., Fei-Fei, L.: 3d object representations for fine-grained categorization. In: 4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13). Sydney, Australia (2013)
23. Lee, C.Y., Bhardwaj, A., Di, W., Jagadeesh, V., Piramuthu, R.: Region-based discriminative feature pooling for scene text recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4050–4057 (2014)
24. Lee, C.Y., Osindero, S.: Recursive Recurrent Nets with Attention Modeling for OCR in The Wild. In: CVPR. pp. 2231–2239 (2016)
25. Li, H., Wang, P., Shen, C., Zhang, G.: Show, Attend and Read: A Simple and Strong Baseline for Irregular Text Recognition. AAAI (2019)
26. Liao, M., Zhang, J., Wan, Z., Xie, F., Liang, J., Lyu, P., Yao, C., Bai, X.: Scene text recognition from two-dimensional perspective. arXiv preprint arXiv:1809.06508 (2018)
27. Liu, W., Chen, C., Wong, K.Y.K.: Char-Net: A Character-Aware Neural Network for Distorted Scene Text Recognition. In: AAAI (2018)
28. Long, S., Guan, Y., Bian, K., Yao, C.: A new perspective for flexible feature gathering in scene text recognition via character anchor pooling. arXiv preprint arXiv:2002.03509 (2020)
29. Luo, C., Jin, L., Sun, Z.: MORAN: A Multi-Object Rectified Attention Network for Scene Text Recognition. Pattern Recognition **90**, 109–118 (2019)
30. Luong, M.T., Pham, H., D.Manning, C.: Effective Approaches to Attention-based Neural Machine Translation. In: EMNLP (2015)
31. Lyu, P., Yang, Z., Leng, X., Wu, X., Li, R., Shen, X.: 2d attentional irregular scene text recognizer. arXiv preprint arXiv:1906.05708 (2019)
32. Mishra, A., Alahari, K., Jawahar, C.V.: Scene text recognition using higher order language priors. In: BMVC-British Machine Vision Conference. BMVA (2012)
33. Mishra, A., Alahari, K., Jawahar, C.V.: Top-down and bottom-up cues for scene text recognition. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 2687–2694 (2012)
34. Mishra, A., Alahari, K., Jawahar, C.V.: Enhancing Energy Minimization Framework for Scene Text Recognition with Top-down Cues. Computer Vision and Image Understanding **145**, 30–42 (2016)
35. Qin, S., Bissacco, A., Raptis, M., Fujii, Y., Xiao, Y.: Towards Unconstrained End-to-End Text Spotting. In: ICCV (2019)
36. Quy Phan, T., Shivakumara, P., Tian, S., Lim Tan, C.: Recognizing text with perspective distortion in natural scenes. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 569–576 (2013)
37. Risnumawan, A., Shivakumara, P., Chan, C.S., Tan, C.L.: A robust arbitrary text detection system for natural scene images. Expert Systems with Applications **41**(18), 8027–8048 (2014)
38. Shaw, P., Uszkoreit, J., Vaswani, A.: Self-Attention with Relative Position Representations. arXiv preprint arXiv:1803.02155 (2018)
39. Sheng, F., Chen, Z., Xu, B.: NRTR: A No-Recurrence Sequence-to-Sequence Model for Scene Text Recognition. arXiv preprint (2017)
40. Shi, B., Bai, X., Yao, C.: An End-to-End Trainable Neural Network for Image-Based Sequence Recognition and Its Application to Scene Text Recognition. PAMI **39**(11), 2298–2304 (2016)

41. Shi, B., Wang, X., Lyu, P., Yao, C., Bai, X.: Robust Scene Text Recognition with Automatic Rectification. In: CVPR. pp. 4168–4176 (2016)
42. Shi, B., Yang, M., Wang, X., Lyu, P., Yao, C., Bai, X.: ASTER : An Attentional Scene Text Recognizer with Flexible Rectification. PAMI (2018)
43. Silva, S.M., Jung, C.R.: License plate detection and recognition in unconstrained scenarios. In: 2018 European Conference on Computer Vision (ECCV) (2018)
44. Singh, A., Natarajan, V., Shah, M., Jiang, Y., Chen, X., Batra, D., Parikh, D., Rohrbach, M.: Towards vqa models that can read. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8317–8326 (2019)
45. Sun, Y., Liu, J., Liu, W., Han, J., Ding, E., Liu, J.: Chinese Street View Text : Large-scale Chinese Text Reading with Partially Supervised Learning. In: ICCV. pp. 9086–9095 (2019)
46. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to Sequence Learning with Neural Networks. In: NIPS (2014)
47. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems. pp. 5998–6008 (2017)
48. Wang, K., Babenko, B., Belongie, S.: End-to-end scene text recognition. In: 2011 International Conference on Computer Vision. pp. 1457–1464 (2011)
49. Wang, K., Belongie, S.: Word spotting in the wild. In: European Conference on Computer Vision. pp. 591–604. Springer (2010)
50. Wang, P., Yang, L., Li, H., Deng, Y., Shen, C., Zhang, Y.: A Simple and Robust Convolutional-Attention Network for Irregular Text Recognition. In: aXiv preprint (2019)
51. Wang, T., Zhu, Y., Jin, L., Luo, C., Chen, X., Wu, Y., Wang, Q., Cai, M.: Decoupled attention network for text recognition. In: AAAI Conference on Artificial Intelligence (2020)
52. Xie, Z., Huang, Y., Zhu, Y., Jin, L., Liu, Y., Xie, L.: Aggregation cross-entropy for sequence recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6538–6547 (2019)
53. Xing, L., Tian, Z., Huang, W., Scott, M.R.: Convolutional Character Networks. In: ICCV. pp. 9126–9136 (2019)
54. Xu, K., Courville, A., Zemel, R.S., Bengio, Y.: Show , Attend and Tell : Neural Image Caption Generation with Visual Attention. In: ICML (2015)
55. Yang, M., Guan, Y., Liao, M., He, X., Bian, K., Bai, S., Yao, C., Bai, X.: Symmetry-constrained rectification network for scene text recognition. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 9147–9156 (2019)
56. Yang, X., He, D., Zhou, Z., Kifer, D., Giles, C.L.: Learning to Read Irregular Text with Attention Mechanisms. In: IJCAI (2017)
57. Zhan, F., Lu, S.: Esir: End-to-end scene text recognition via iterative image rectification. arXiv preprint arXiv:1812.05824 (2018)