Parallel and Robust Text Rectifier for Scene Text Recognition

Bingcong Li[1†]
bingcong_li@163.com
Xin Tang[1†]
tangxint@gmail.com
Jun Wang[2]
deeplearning.pku@qq.com
Liang Diao[1]
diaoliang91@gmail.com
Rui Fang[1]
rui_fang_science@163.com
Guotong Xie[2]
xieguotong_cmt@163.com
Weifu Chen[3*]
waifook.chan@gmail.com

[1] Visual Computing Group, Ping An Property & Casualty Insurance Company, Shenzhen, China
[2] Ping An Technology (Shenzhen) Co. Ltd.
[3] School of Information and Telecommunication Engineering, Guangzhou Maritime University, Guangzhou, China

Abstract

Scene text recognition (STR) is to recognize text appearing in images. Current state-of-the-art STR methods usually adopt a multi-stage framework which uses a rectifier to iteratively rectify errors from previous stage. However, the rectifiers of those models are not proficient in addressing the misalignment problem. To alleviate this problem, we proposed a novel network named Parallel and Robust Text Rectifier (PRTR), which consists of a bi-directional position attention initial decoder and a sequence of stacked Robust Visual Semantic Rectifiers (RVSRs). In essence, PRTR is creatively designed as a coarse-to-fine architecture that exploits a sequence of rectifiers for repeatedly refining the prediction in a stage-wise manner. RVSR is a core component in the proposed model which comprises two key modules, Dual-Path Semantic Alignment (DPSA) module and Visual-Linguistic Alignment (VLA). DPSA can rectify the linguistic misalignment issues via the global semantic features that are derived from the recognized characters as a whole, while VLA re-aligns the linguistic features with visual features by an attention model to avoid the overfitting of linguistic features. All parts of PRTR are non-autoregressive (**parallel**), and its RVSR re-aligns its output according to the linguistic features and the visual features, so it is **robust** to the mis-aligned error. Extensive experiments on mainstream benchmarks demonstrate that the proposed model can alleviate the misalignment problem to a large extent and outperformed state-of-the-art models.

† : Equal contribution; *: Corresponding author.

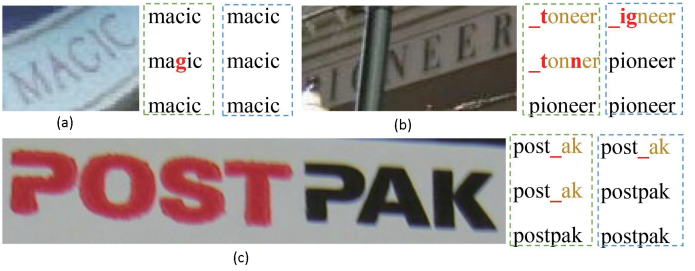

Figure 1: Three examples with intermediate predictions. Texts in green dash boxes and blue dash boxes represent the results of ABINet [7] and PRTR (ours) respectively. In each box, the first row is the initial prediction, the second row is the prediction of the rectifier, and the last one is the ground-truth. "_" is a placeholder, characters in red represent aligned errors, and characters in yellow correspond to misaligned errors.

1 Introduction

Scene text recognition (STR) which aims to recognize text from natural images, is an important task in computer vision (CV) and natural language processing (NLP), and benefits for many downstream tasks, such as automatic driving, travel translator etc. Currently, mainstream models for STR are based on attention mechanism [11, 12, 22, 25, 26, 28], which mainly focus on how to guide the visual features to attend the corresponding regions of characters. However, it is challenging for accurately identifying each character from scene images, due to uncontrolled issues like illumination, distortion or obstacle.

There exist a bunch of methods [1, 2, 17, 27] to model the linguistic information to enrich the global context for character recognition. For example, [27] proposed a semantic reasoning network to capture the global semantic context via a transformer-based module. Fang *et al*. [7] proposed a language model named ABINet to estimate the probability distribution of characters for progressively correcting the prediction. Similarly, [1] proposed a multi-stage prediction paradigm with joint visual-semantic reasoning (JVSR) to make full use of visual and linguistic clues for progressively refining recognized letters. We divide these methods into two categories according to whether the prediction sequence is refined: One-stage STR methods [10, 11, 12, 13, 20, 21, 22, 28] and Multi-stage STR methods[1, 2, 27]. One-stage STR methods directly predict the final text, while multi-stage STR methods contain a rectifier module to refine previous predictions.

Although most of previous frameworks are effective in capturing the global semantic context with a rectifier to refine the last prediction, they still easily fail to achieve high Recognition accuracy for STR tasks due to following limitations in the rectifying process: (i) those model are prone to recognize test characters as existing words with their language models, for example, as shown in Fig. 1(a), ABINet turns the correct prediction of the initial decoder into "magic"; (ii) it is not effective enough for those models to solve the misaligned problem which we focus on in this work, as shown in Fig. 1(b) and (c). Here "*aligned error*" is defined as the error that if the selection of operations contains only substitution during the calculation of the minimum edit distance of the incorrect prediction and the ground-truth, and other prediction errors except aligned errors are defined as "*misaligned error*". For example, assume the ground-truth is "aeroplane" and the prediction is "aenoplane", then the error is an aligned error, while if the prediction is "aeoplane", it is a misaligned error. As

	Error rectified (%)	
Error type	aligned	misaligned
ABINet	44.5	3.6
PRTR (Ours)	44.2	17.4

Table 1: Quantitative analysis of ABINet[2] and our method on the performance of the rectifier. For each method, we categorize the errors of the initial decoder on the test datasets into misaligned and aligned errors, and summaries the ratio of errors corrected by the rectifier.

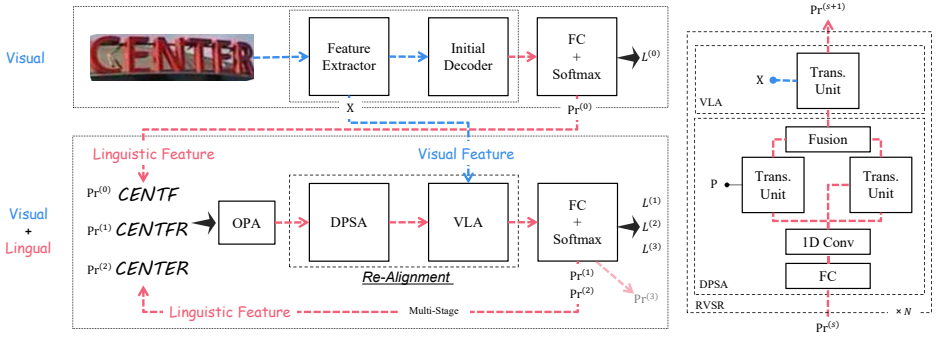

Figure 2: Overall architecture of PRTR. OPA, DPSA and VLA denote online probability augmentation, dual-path semantic alignment and visual-linguistic alignment respectively. The right part demonstrates the structure of DPSA and VLA.

listed in Tab. 1, ABINet is able to rectify 44.5% of the aligned errors from the initial decoder, but can only rectify 3.6% of the misaligned errors. More efficient and robust models are required to promote recognition accuracy for misaligned errors.

To mitigate the misalignment issues we propose a novel framework called Parallel and Robust Text Rectifier (PRTR) which mainly consists of an initial encoder and a sequence of Robust Visual-Semantic Rectifiers (RVSRs). The novelty of PRTR to solve this misalignment problem is to rectify the prediction error in a stage-by-stage manner, which is implemented by imposing a sequence of RVSRs on the output of the initial encoder. Since RVSR comprises two alignment modules, namely dual-path semantic alignment (DPSA) module and a visual-linguistic alignment (VLA) module, which aims to align semantic information with linguistic features, and linguistic features with visual features respectively, it can obviously alleviate misalignment errors. We also propose an online probability augment (OPA) module to strengthen the error correction ability of RVSR. Hence, our RVSR is robust to error propagation derived from the primitive prediction of the initial decoder. We summarize the major contributions of this work as: (1) We propose a multi-stage parallel and robust text rectifier framework (PRTR) to solve the misalignment error; (2) We explain why the misalignment error can be alleviated by PRTR in a stage-by-stage rectifying manner; (3) Extensive experiments conducted on seven benchmarks demonstrate that our PRTR achieved outstanding performance comparing with popular state-of-the-art models.

[2]Test with the model provided officially on https://github.com/FangShancheng/ABINet

2 Methodology

Motivation: Given an input image $\mathbf{I} \in \mathbb{R}^{H\times W\times 3}$, we aim at predicting the character sequence $Y = \{y_1, y_2, \cdots, y_T\}$ in the image, where T is the length of the text. For convenience, let $p(Y|\mathbf{I})$ denote the conditional distribution over $\mathbf{I}$. Motivated by the observation in physiology that humans progressively improve prediction confidence by iteratively correcting the recognition results [2], instead of predicting Y directly from $\mathbf{I}$, we sequentially predict $S(S \geq 1)$ character sequences $\{Y^{(s)}\}_{s=0}^{S}$ to progressively approximate the ground-truth sequence Y^* . Assume that the prediction $Y^{(s)}$ just depends on $Y^{(s-1)}$ and $\mathbf{I}$, the conditional likelihood $p(Y|\mathbf{I})$ to incorporate the intermediate predictions can be formulated as:

$$\begin{aligned} p(Y|\mathbf{I}) &= p(Y^{(S)}|\mathbf{I}) = p(Y^{(S)}|\mathbf{I}) \cdot 1 = p(Y^{(S)}|\mathbf{I}) p(|Y^{(0)}, \cdots, Y^{(S-1)}|Y^{(S)}, \mathbf{I}) \\ &= p(Y^{(0)}, \cdots, Y^{(S-1)}, Y^{(S)}|\mathbf{I}) = p(Y^{(S)}|Y^{(S-1)}, \mathbf{I}) \cdots p(Y^{(1)}|Y^{(0)}, \mathbf{I}) p(Y^{(0)}|\mathbf{I}) \end{aligned} \tag{1}$$

In order to achieve above goals, our framework is designed to consist of three parts: (i) *a visual feature extraction network* that extracts a context-rich visual feature map, (ii) *an initial decoder* that predicts the character sequence $Y^{(0)}$ based on the visual feature map, and (iii) *a robust visual-semantic rectifier (RVSR) module* that generates the character sequence $Y^{(s)}$ by progressively rectifying the last prediction $Y^{(s-1)}$. Since we rectify the prediction in a stage-by-stage manner, several RVSRs will be stacked to enhance linguistic information in the prediction sequence.

2.1 Visual Feature Extraction

We adopt ResNet31 [4] with multi-aspect global context attention [14] as the CNN-based feature extractor ($\mathcal{C}$). To enrich the feature context, we use several transformer encoder layers [23] ($\mathcal{T}$) to model global spatial dependencies like in [2, 27]. Then, the visual features $\mathbf{X}$ can be obtained as:

$$\mathbf{X} = \mathcal{R}(\mathcal{T}(\mathcal{C}(\mathbf{I}))) \in \mathbb{R}^{N\times D}, \tag{2}$$

where $\mathcal{R}$ represents the reshape operator, D denotes the dimension of the features, $N = \frac{H}{8} \times \frac{W}{4}$, H and W are the height and the width of the image $\mathbf{I}$.

2.2 Initial Decoder

The distribution $p(Y^{(0)}|\mathbf{I})$ denoting the initial decoder in Eq. (1) can be written as $p(Y^{(0)}|\mathbf{I}) = \prod_{t=1}^{T} p(y_t|\mathbf{X})$. Each character output y_t is predicted by a conditional distribution over the visual features $\mathbf{X}$. The initial decoder is entirely visually based. As in [2], position attention is used to obtain the aligned visual features for all time-steps, based on the query paradigm:

$$\mathbf{V} = \text{softmax}(\frac{\mathbf{Q}\mathcal{U}(\mathbf{X})^T}{\sqrt{D}})\mathbf{X}, \tag{3}$$

where $\mathcal{U}(\cdot)$ is a mini U-Net, $Q \in \mathbb{R}^{T\times D}$ is positional encodings denoting the character orders, $V \in \mathbb{R}^{T\times D}$ is the aligned visual features, and T is the length of character sequence. Given the aligned features, the output probability over characters at time step t is computed by $\mathbf{Pr}_t^{(0)} = \text{Softmax}(\mathbf{W}^{(0)}\mathbf{v}_t + \mathbf{b}^{(0)}) \in \mathbb{R}^C$, where $\mathbf{W}^{(0)} \in \mathbb{R}^{C\times D}$ and $\mathbf{b}^{(0)} \in \mathbb{R}^D$ are trainable parameters.

2.3 Robust Visual-Semantic Rectifier

In the initial decoder, we decode the character sequence in a non-autoregressive way, which lacks the ability of modeling contextual information. Therefore, there inevitably exist errors in $Y^{(0)}$ which we expect can be rectified by language models. We propose a robust visual-semantic rectifier (RVSR) to progressively refine erroneous predictions via joint visual-semantic alignment in a stage-by-stage manner. Suppose there are S stages of RVSR, at Stage 1, the input of RVSR is the prediction of the inital decoder $\mathbf{Pr}_t^{(0)}$ and the output of RVSR is $\mathbf{Pr}_t^{(1)}$. At Stage s, the refined prediction $Y^{(s)}$ is modeled as follows:

$$p(Y^{(s)}|Y^{(s-1)},\mathbf{I}) = \prod_{t=1}^{T} p(y_t^{(s)}|y_1^{(s-1)},y_2^{(s-1)},\cdots,y_T^{(s-1)},\mathbf{I}) = \prod_{t=1}^{T} p(y_t^{(s)}|\mathcal{E}(y_1^{(s-1)}),\cdots,\mathcal{E}(y_T^{(s-1)}),\mathbf{X})$$
$$= \prod_{t=1}^{T} p(y_t^{(s)}|\mathcal{E}(\mathcal{A}(\mathbf{Pr}_1^{(s-1)})),\cdots,\mathcal{E}(\mathcal{A}(\mathbf{Pr}_T^{(s-1)})),\mathbf{X}) \approx \prod_{t=1}^{T} p(y_t^{(s)}|\mathcal{F}(\mathbf{Pr}_1^{(s-1)}),\cdots,\mathcal{F}(\mathbf{Pr}_T^{(s-1)}),\mathbf{X}),$$

where $\mathcal{A}$ is non-differentiable *argmax* operator, $\mathcal{E}$ denotes the character embedding and $\mathcal{F}$ represents the fully connected layer. RVSR uses $\mathbf{Pr}_t^{(s-1)}$ which contain richer semantic information to approximate one-hot encoding $y_t^{(s-1)}$. RVSR first uses a fully connected layer to convert probability vectors into feature vectors $\mathbf{e}_t^{(s-1)} = \mathcal{F}(\mathbf{Pr}_1^{(s-1)})$. Since $\{y_t^{(s-1)}\}_{t=1}^T$ may contain errors, it is inevitable to propagate the errors to the linguistic features $\{\mathbf{e}_t^{(s-1)}\}_{t=1}^T$. RVSR employs a dual-path semantic alignment (DPSA) module and a visual-lingual alignment (VLA) module to refine the prediction with linguistic knowledge among the words and visual information of the images.

Dual-path semantic alignment (DPSA). The structure of a DPSA module is shown in the right block of Fig. 2. To model the semantic information among the linguistic feature sequence, DPSA projects $\mathbf{Pr}_t^{(s-1)}$ to a feature space with a fully-connected layer and obtains $\{\mathbf{e}_t^{(s-1)}\}_{t=1}^T$. As stated above, $\mathbf{Pr}_t^{(s-1)}$ might contain noise. In computer vision, blurring is commonly used to preprocess noise reduction. Inspired by this, DPSA applies 1D convolution (with parameters $\mathbf{w}$ and kernel size k) to $\{\mathbf{e}_t^{(s-1)}\}_{t=1}^T$ and acquires $\{\mathbf{e}_t^{(s)}\}_t^T$ as following:

$$\mathbf{e}_t^{(s)} = \sum_{i=1}^{k} \mathbf{w}_i \cdot \mathbf{e}_{t-\lfloor \frac{k}{2} \rfloor + i}^{(s-1)}. \tag{4}$$

To enhance the linguistic knowledge, DPSA introduces a semantic-enhancing path in which a vanilla Transformer unit is employed, which is defined as:

$$\hat{\mathbf{e}}_t^{(s)} = \alpha_t v_t + \sum_{j \neq t} \alpha_j v_j, \tag{5}$$

where $\alpha_j = \exp(q_t^T k_j)/\sum_{i=1}^T \exp(q_t^T k_i)$ and $q_t = \mathbf{W}_q \mathbf{e}_t^{(s)}$, $k_t = \mathbf{W}_k \mathbf{e}_t^{(s)}$,$v_t = \mathbf{W}_v \mathbf{e}_t^{(s)}$ are the query, key and value ($t \in \{1,2,...,T\}$), separately. To capture the diversity of relationships, multi-head mechanism is integrated in Eq. (5).

However, if some characters are missed or superfluous in $\{y_t^{(s-1)}\}_{t=1}^T$, most linguistic feature vectors in the sequence could be misaligned. To handle this problem, DPSA additionally introduces a semantic-aligning path to alleviate the misalignment propagated from the prediction of Stage $s-1$. Like in [2], the semantic-aligning path contains a Transformer unit which take T trainable global tokens $\{\mathbf{p}_t\}_{t=1}^T$ as query, and $\{\mathbf{e}_t^{(s)}\}_{t=1}^T$ as key and value,

where $\{\mathbf{p}_t\}_{t=1}^{T}$ are independent on $\{\mathbf{e}_t^{(s)}\}_{t=1}^{T}$. Ideally, the semantic-aligning path is designed to align the output with the ground-truth sequence and acquire a novel linguistic feature sequence $\widetilde{\mathbf{E}}_t^{(s)} \in \mathbb{R}^{T\times D}$ as follows:

$$\widetilde{\mathbf{E}}_t^{(s)} = \text{FFN}(\text{MultiHead}(\mathbf{P}, \mathbf{E}^{(s)}, \mathbf{E}^{(s)})) \tag{6}$$

where $\mathbf{P}$ and $\mathbf{E}^{(s)}$ are the matrix form of $\{\mathbf{p}_t\}_{t=1}^{T}$ and $\{\mathbf{e}_t^{(s)}\}_{t=1}^{T}$. $\text{MultiHead}(query, key, value)$ and $\text{FFN}(x)$ are defined the same as in the standard Transformer encoder [23].

Combining the outputs of these two paths, we have,

$$\begin{aligned} \mathbf{e}_\mathbf{t}^{(\mathbf{s})} &= \mathbf{G} \odot \hat{\mathbf{e}}_t^{(s)} + (1-\mathbf{G}) \odot \widetilde{\mathbf{e}}_t^{(s)} \\ \mathbf{G} &= \sigma(\mathbf{W}_g \, \text{Concat}(\hat{\mathbf{e}}_t^{(s)}, \widetilde{\mathbf{e}}_t^{(s)})) \end{aligned}, \tag{7}$$

where $\mathbf{W}_g \in \mathbb{R}^{2D\times D}$ and $\mathbf{G} \in \mathbb{R}^{D}$. Consequently, $\mathbf{e}_t^{(s)}$ has been enhanced and realigned.

Visual-linguistic alignment (VLA) is to align the linguistic features with the visual features, as depicted in Fig. 2. Previous researches show that fusing the aligned linguistic features learned by the initial decoder and the rectifiers can give better results [2, 27]. However, the fusion inevitably propagates misalignment to the final result. Aim to handle the error propagation problem, VLA applies an attention model on the visual feature $\mathbf{X}$, in which each linguistic feature $\mathbf{e}_\mathbf{t}^{(\mathbf{s})}$ focuses on a specific region of $\mathbf{X}$ and is combined with other selected visual features of interest,

$$\mathbf{e}_t^{(s)} = \text{FFN}(\text{MultiHead}(\mathbf{e}_t^{(s)}, \mathbf{X}, \mathbf{X})), \tag{8}$$

where $\text{MultiHead}, \text{FFN}$ are the same as Eq. (6).

In this work, L RVSR blocks are stacked to update the linguistic features $\mathbf{E}^{(s)} \in \mathbb{R}^{T\times D}$. Finally, each linguistic feature is transferred into probabilities over the character set and obtained a novel prediction $\mathbf{Pr}_t^{(s)} = \text{Softmax}(\mathbf{W}^{(s)}\mathbf{e}_t^{(s)} + \mathbf{b}^{(s)})$, where $\mathbf{W}^{(s)} \in \mathbb{R}^{C\times D}$ and $\mathbf{b}^{(s)} \in \mathbb{R}^{D}$ are parameters.

Online Probability Augmentation (OPA). In this work, four random data augmentation strategies (including random removal, repetition, mixture and replacement) are adopted to enhance the generalization and robustness of RVSR. OPA simulates possible types of errors in the prediction, and pushes RVSR to correct them. We use an example to demonstrate how these strategies are implemented. Let $\mathbf{Pr}^{(0)} \in \mathbb{R}^{T\times C}$ represent the probabilities over the character set for all T time steps and $\mathbf{Pr}_t^{(0)}$ correspond to the probabilities at the time step t. Denote $\mathbf{o}$ as the one-hot encoding of padding symbols. We build a probability memory queue $\mathbf{M} \in \mathbb{R}^{K\times C}$ by randomly sampling probability vectors $\mathbf{Pr}^{(0)}$ in each mini-batch. In each subsequent iteration, we randomly select a row of $\mathbf{Pr}^{(0)}$ (for instant $\mathbf{Pr}_t^{(0)}$) and one of the four augmentation strategies to be performed. If "removal" is sampled, we remove $\mathbf{Pr}_t^{(0)}$, and $\mathbf{o}$ is added into the last row of $\mathbf{Pr}^{(0)}$. If random "repetition" is selected, we insert another copy of $\mathbf{Pr}_t^{(0)}$ after itself and remove the last row of $\mathbf{Pr}^{(0)}$. "Random replacement" is that we randomly choose a sample from $\mathbf{M}$ to replace $\mathbf{Pr}_t^{(0)}$. Like mixup [29], "random mixture" randomly chooses a sample from $\mathbf{M}$ and mixes it with $\mathbf{Pr}_t^{(0)}$.

Arch.		Data		IIIT	SVT	IC 03	IC 13	IC 15	SVTP	CUTE	Avg
GCB	TPS	Aug.	SA								
				95.1	87.3	94.2	94.1	78.4	80.9	92.0	89.4
✓				95.8	88.9	94.9	94.9	80.7	84.5	90.3	90.7
✓	✓			95.5	89.8	95.5	95.0	83.0	85.4	92.0	91.3
✓	✓	✓		96.2	92.0	95.8	95.0	84.2	86.9	93.1	92.2
✓	✓	✓	✓	96.3	92.3	95.2	94.8	84.7	87.0	93.8	92.3

Table 2: The effectiveness of GCB, TPS, data augmentation and SA on the baseline model.

2.4 Training Objective

PRTR is trained in an end-to-end manner by penalizing the difference between the prediction and ground-truth at each stage. The objective function is given by:

$$L = \lambda_0 L^{(0)} + \sum_{s=1}^{S} \lambda_s L^{(s)} = -\lambda_0 \sum_{t=1}^{T} y_t^* \log p(\mathbf{Pr}_t^{(0)}) - \sum_{s=1}^{S} \lambda_s \sum_{t=1}^{T} y_t^* \log p(\mathbf{Pr}_t^{(s)} | \mathbf{Pr}_t^{(s-1)}) \quad (9)$$

where $\{y_1^*, y_2^*, \cdots, y_T^*\}$ is the ground-truth label, λ_0, λ_s and S are hyper-parameters.

3 Experiments

3.1 Datasets and Implementation Details

IIIT 5K-words (IIIT) [16] contains 3,000 cropped images for test. **Street View Text** (SVT) [24] has 647 testing images. **ICDAR 2003** (IC 03) [15] contains 867 cropped images for testing after discarding images that contained non-alphanumeric characters or had fewer than three characters [24]. **ICDAR 2013** (IC 13) [8] contains 1,095 testing images. Following [28], We select 1,015 images as a test set. **ICDAR 2015** (IC 15) [9] is captured by Google Glass without careful focusing. We use 1,811 images for evaluation without some extremely distorted images like in [2, 27]. **Street View Text Perspective** (SVTP) [18] contains 639 testing images. **CUTE** [19] contains 288 testing images.

The proposed PRTR was implemented using the PyTorch framework on 4 Tesla V100 GPUs with 16GB memory. The codes will be released soon. As [5, 11, 17], PRTR was trained from scratch only on synthetic datasets, including MJSynth (MJ) [6], SynthText (ST) [3] and SynthAdd (SA) [11], with a sampling ratio of 0.4 : 0.4 : 0.2. Images were resized to 48×160 (keeping the original ratio). 39 symbols were recognized in total, which are 0-9, a-z, end-of-sequence symbols and unknown symbols. We used ADAM optimizer with the batch size of 272. In the first 50 steps, we used warming-up strategy. Then the learning rate was set to $1e-4$ for training and decreased it to $1e-5$ at 0.8M. The training was stopped at 1.2M. We fixed the random seed for all experiments. The model dimension D was set to 512 throughout. There were 3 stacked transformer units in visual feature extraction. For each transformer unit, the number of heads was 8 and the number of hidden units were 512. RVSR consisted of 3 blocks, thus $L = 3$. The hyper-parameters S, λ_0, λ_1, λ_2, λ_3 were set to 3, 0.2, 0.2, 0.2, 0.8, respectively.

3.2 Ablation study

Baseline. For comparison, we build a baseline model. Like PRTR, the backbone of the baseline model was also ResNet31 with a 3-layer transformer encoder stacked on the top

DPSA	VLA	OPA	IIIT	SVT	IC03	IC13	IC15	SVTP	CUTE	Avg
			96.3	92.3	95.2	94.8	84.7	87.0	93.8	92.3
✓			95.6	94.0	96.8	96.2	85.6	90.4	94.1	93.0
✓		✓	96.2	94.6	97.0	95.9	84.9	90.2	94.8	93.1
✓	✓		96.8	93.5	96.0	96.0	85.0	89.5	95.5	93.1
✓	✓	✓	97.0	94.4	96.4	95.8	86.1	89.8	96.5	93.6

Table 3: Ablation study of RVSR. We design a serial of experiments to validate the impact of each component.

Conv.	IIIT	SVT	IC03	IC13	IC15	SVTP	CUTE	Avg
✗	96.0	92.6	96.5	95.3	84.8	88.7	94.4	92.6
✓	95.6	94.0	96.8	96.2	85.6	90.4	94.1	93.0

Table 4: The effect of the 1D convolution in DPSA. Conv denotes whether employ 1D convolution.

S	IIIT	SVT	IC03	IC13	IC15	SVTP	CUTE	Avg
1	96.6	93.5	96.0	95.4	84.8	88.5	95.8	92.9
2	96.6	93.2	96.3	96.2	86.3	89.6	95.5	93.4
3	97.0	94.4	96.4	95.8	86.1	89.8	96.5	93.6

Table 5: The performance of PRTR with different numbers of stages (S). Increasing the number of stages has positive impacts on the recognition performance.

and a multi-aspect global context attention block (GCB) [3] applied to the feature maps at intermediate three stages. We used Thin Plate Spline (TPS) [2, 22] transformation to normalize input images. Data augmentation including perspective distortion, blur, Gaussian noise and color jitter were used to train the baseline. To check whether synthesized datasets were useful for training the models, we also tested the performance on the aforementioned SA dataset [11]. Tab. 2 shows the effectiveness of these four components (GCB, TPS, data augmentation and SA) and the evaluation results of the baseline model. The last column is the average accuracy on the seven test datasets. GCB boosted the accuracy about 1.3% on average. TPS, data augmentation and SA also increased the accuracy by 0.6%, 0.9%, and 0.1% respectively. These tricks were also used in PRTR.

Comparing with the baseline, *PRTR introduces additional robust visual-semantic rectifiers (RVSRs) to refine the predictions in a stage-by-stage manner, where each RVSR block contains a dual-path semantic alignment (DPSA) module and a visual-linguistic alignment (VLA) module*. We also present an online probability augmentation (OPA) to augment the intermediate predictions. We performed several experiments to validate the effectiveness of these components in following.

The effectiveness of DPSA. The proposed DPSA captures the semantic information in previous prediction and realign the linguistic features with its dual-path structure. Comparing the first row and the second row in Tab. 3, using DPSA alone improved the baseline model by 0.7% in average accuracy. Especially, DPSA boosted the accuracy by 1.7%, 1.6%, 3.4% on SVT, IC03 and SVTP respectively. We also found an interesting conclusion that, by removing the 1D convolution from DPSA, the average accuracy was decreased by 0.4%, as shown in Tab. 4. We reckon that the 1D convolution acts like a blurring filter and reduces the impact of input noises.

The effectiveness of OPA. To boost the correction capability of the rectifier, ABINet and JVSR both pre-train themselves with extra corpus by language model training topology. However, this strategy is not end-to-end, which makes the training procedure complex.

Moreover, in the fine-tuning phase, we found that the output of the initial decoder achieved higher than 98% training accuracy. This indicates that the rectifier may lack of erroneous cases to learn, which limits the rectifying ability of the rectifiers on test instances, which spurs us to propose OPA. According to the second and third rows of Tab. 3, it can be seen that OPA further improves the performance of using DPSA alone by 0.1%.

The effectiveness of VLA. VLA is used to refine the linguistic features by aligning it with the visual features and avoid the over-confidence on the linguistic knowledge, such as the failure case of "MACIC" in Fig. 1. Unlike ABINet, VLA does not integrate the misaligned visual features of the initial decoder which may contain potential misaligned errors. As shown in the second and fourth rows in Tab. 3, VLA slightly improved the average performance by 0.1%. Especially, VLA increased the accuracy by 1.4% on CUTE, which consists of many high-resolution images. We believe that VLA achieved high performance on CUTE due to its ability to repeatedly align the linguistic features with the high-quality visual features.

Interaction between these components. From the experiments above, using OPA and VLA alone with DPSA could only improve the average accuracy by small margins. However, when they were used together, the accuracy was boosted about 0.5%, as shown in the fifth row of Tab. 3. A reasonable explanation is that, by adding VLA alone, we increase the model capacity, but there are not enough erroneous cases to learn. OPA enriches the erroneous cases, and thus combining these two modules together can improve the performance largely.

Progressive rectification. We conducted a serial of experiments to explore the impact of the number of stages (S). Experiment results are shown in Tab. 5, where S is set to be 1, 2, and 3. Compared to the baseline model, An RVSR block can boost the average accuracy about 0.6%. When we increase the number of RVSR blocks, PRTR performs better. With 3 RSVR blocks, PRTR improve the average accuracy by 1.3% and achieves 97% accuracy on IIIT, 96.5% accuracy on CUTE.

3.3 Comparisons with State-of-the-Arts

The comparison of our proposed PRTR with state-of-the-art methods is illustrated in Tab. 6. PRTR exhibited superior performance on most of the datasets. Especially, PRTR achieved 97% on IIIT while the result of ABINet-LV is 96.2%. PRTR improved 4.8% on CUTE (from 91.7% to 96.5%). Compared to our baseline, PRTR boosted the accuracy on CUTE by 2.7% (from 93.8% to 96.5%). Especially, Table 3 shows that VLA improved the accuracy on CUTE by 1.7%. CUTE contains images with high quality but extreme curvy text. In our setting, PRTR revisits the visual features in 9 VLA layers, and thus can make good use of the high quality visual features. For fair comparison with other methods, we also conducted experiment without SynthAdd dataset. Results show that PRTR without SA also achieves impressive performance.

4 Conclusion

In this paper, we propose a parallel and robust scene text recognition framework (PRTR) to tackle the misalignment issues of conventional multi-stage STR methods, which is equipped with stacked RVSR blocks for rectifying primitive predictions from visual-semantic perspective. The proposed model can effectively reduce the semantic alignment error and visual-linguistic error, and hence outperform other state-of-the-art models on seven popular bench-

Method	Regular test dataset				Irregular test dataset		
	IIIT	SVT	IC03	IC13	IC15	SVTP	CUTE
ASTER [20]	93.4	89.5	94.5	91.8	76.1	78.5	79.5
SAR [19]	91.5	84.5	_	91.0	69.2	76.4	83.3
DAN [23]	94.3	89.2	95.0	93.9	74.5	80.0	84.4
SRN [22]	94.8	91.5	_	95.5	82.7	85.1	87.8
SCATTER [18]	93.7	92.7	96.3	93.9	82.2	86.9	87.5
RobustScanner [28]	95.3	88.1	_	94.8	77.1	79.5	90.3
GTC [6]	95.5	92.9	95.2	94.3	82.5	86.2	92.3
JVSR [8]	95.2	92.2	_	95.5	84.0	85.7	89.7
PREN [21]	95.6	94.0	95.8	96.4	83.0	87.6	91.7
ABINet-SV [5]	95.4	93.2	_	96.8	84.0	87.0	88.9
ABINet-LV [5]	96.2	93.5	_	**97.4**	86.0	89.3	89.2
PRTR (OURS)	**97.0**	94.4	**96.4**	95.8	**86.1**	**89.8**	**96.5**
PRTR WO\SA (OURS)	96.9	**94.6**	96.1	95.5	85.7	88.8	96.2

Table 6: Performance Comparison on seven benchmarks. **Bold** and underline represents the best and the second best performance. The **PRTR** row shows the results of PRTR trained only MJ and ST. The **PRTR** WO\SA row shows the results removing the usage of SynthAdd.

mark datasets. In the future, we will further generalize PRTR on low-resolution or impaired text images.

References

[1] Ayan Kumar Bhunia, Aneeshan Sain, Amandeep Kumar, Shuvozit Ghose, Pinaki Nath Chowdhury, and Yi-Zhe Song. Joint visual semantic reasoning: Multi-stage decoder for text recognition. *arXiv preprint arXiv:2107.12090*, 2021.

[2] Shancheng Fang, Hongtao Xie, Yuxin Wang, Zhendong Mao, and Yongdong Zhang. Read like humans: Autonomous, bidirectional and iterative language modeling for scene text recognition. *arXiv preprint arXiv:2103.06495*, 2021.

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

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[5] Wenyang Hu, Xiaocong Cai, Jun Hou, Shuai Yi, and Zhiping Lin. Gtc: Guided training of ctc towards efficient and accurate scene text recognition. In *AAAI*, pages 11005–11012, 2020.

[6] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Synthetic data and artificial neural networks for natural scene text recognition. *arXiv preprint arXiv:1406.2227*, 2014.

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

[8] 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. IEEE, 2013.

[9] 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. IEEE, 2015.

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

[11] Hui Li, Peng Wang, Chunhua Shen, and Guyu Zhang. Show, attend and read: A simple and strong baseline for irregular text recognition. In *AAAI*, volume 33, pages 8610–8617, 2019.

[12] Ron Litman, Oron Anschel, Shahar Tsiper, Roee Litman, Shai Mazor, and R Manmatha. Scatter: selective context attentional scene text recognizer. In *CVPR*, pages 11962–11972, 2020.

[13] Ning Lu, Wenwen Yu, Xianbiao Qi, Yihao Chen, Ping Gong, and Rong Xiao. Master: Multi-aspect non-local network for scene text recognition. *arXiv preprint arXiv:1910.02562*, 2019.

[14] Ning Lu, Wenwen Yu, Xianbiao Qi, Yihao Chen, Ping Gong, Rong Xiao, and Xiang Bai. Master: Multi-aspect non-local network for scene text recognition. *Pattern Recognition*, 117:107980, 2021.

[15] Simon M Lucas, Alex Panaretos, Luis Sosa, Anthony Tang, Shirley Wong, and Robert Young. Icdar 2003 robust reading competitions. In *ICDAR*, pages 682–687. Citeseer, 2003.

[16] Anand Mishra, Karteek Alahari, and CV Jawahar. Top-down and bottom-up cues for scene text recognition. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2687–2694. IEEE, 2012.

[17] Zhi Qiao, Yu Zhou, Dongbao Yang, Yucan Zhou, and Weiping Wang. Seed: Semantics enhanced encoder-decoder framework for scene text recognition. In *CVPR*, pages 13528–13537, 2020.

[18] 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.

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

[20] Fenfen Sheng, Zhineng Chen, and Bo Xu. Nrtr: A no-recurrence sequence-to-sequence model for scene text recognition. In *ICDAR*, pages 781–786. IEEE, 2019.

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

[22] Baoguang Shi, Mingkun Yang, Xinggang Wang, Pengyuan Lyu, Cong Yao, and Xiang Bai. Aster: An attentional scene text recognizer with flexible rectification. *IEEE transactions on pattern analysis and machine intelligence*, 41(9):2035–2048, 2018.

[23] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, pages 5998–6008, 2017.

[24] Kai Wang, Boris Babenko, and Serge Belongie. End-to-end scene text recognition. In *2011 International Conference on Computer Vision*, pages 1457–1464. IEEE, 2011.

[25] Tianwei Wang, Yuanzhi Zhu, Lianwen Jin, Canjie Luo, Xiaoxue Chen, Yaqiang Wu, Qianying Wang, and Mingxiang Cai. Decoupled attention network for text recognition. In *AAAI*, pages 12216–12224, 2020.

[26] Ruijie Yan, Liangrui Peng, Shanyu Xiao, and Gang Yao. Primitive representation learning for scene text recognition. In *CVPR*, pages 284–293, 2021.

[27] Deli Yu, Xuan Li, Chengquan Zhang, Tao Liu, Junyu Han, Jingtuo Liu, and Errui Ding. Towards accurate scene text recognition with semantic reasoning networks. In *CVPR*, pages 12113–12122, 2020.

[28] Xiaoyu Yue, Zhanghui Kuang, Chenhao Lin, Hongbin Sun, and Wayne Zhang. Robustscanner: Dynamically enhancing positional clues for robust text recognition. In *ECCV*, pages 135–151. Springer, 2020.

[29] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.