

SVTRv2: CTC Beats Encoder-Decoder Models in Scene Text Recognition

Yongkun Du¹, Zhineng Chen^{1*}, Hongtao Xie², Caiyan Jia³, Yu-Gang Jiang¹

¹School of Computer Science, Fudan University, China

²University of Science and Technology of China, China

³School of Computer and Information Technology, Beijing Jiaotong University, China

ykdu23@m.fudan.edu.cn, {zhincheng, ygj}@fudan.edu.cn, htxie@ustc.edu.cn, cyjia@bjtu.edu.cn

Abstract

Connectionist temporal classification (CTC)-based scene text recognition (STR) methods, e.g., SVTR, are widely employed in OCR applications, mainly due to their simple architecture, which only contains a visual model and a CTC-aligned linear classifier, and therefore fast inference. However, they generally have worse accuracy than encoder-decoder-based methods (EDTRs), particularly in challenging scenarios. In this paper, we propose SVTRv2, a CTC model that beats leading EDTRs in both accuracy and inference speed. SVTRv2 introduces novel upgrades to handle text irregularity and utilize linguistic context, which endows it with the capability to deal with challenging and diverse text instances. First, a multi-size resizing (MSR) strategy is proposed to adaptively resize the text and maintain its readability. Meanwhile, we introduce a feature rearrangement module (FRM) to ensure that visual features accommodate the alignment requirement of CTC well, thus alleviating the alignment puzzle. Second, we propose a semantic guidance module (SGM). It integrates linguistic context into the visual model, allowing it to leverage language information for improved accuracy. Moreover, SGM can be omitted at the inference stage and would not increase the inference cost. We evaluate SVTRv2 in both standard and recent challenging benchmarks, where SVTRv2 is fairly compared with 24 mainstream STR models across multiple scenarios, including different types of text irregularity, languages, and long text. The results indicate that SVTRv2 surpasses all the EDTRs across the scenarios in terms of accuracy and speed. Code is available at <https://github.com/Topdu/OpenOCR>.

1. Introduction

As a task of extracting text from natural images, scene text recognition (STR) has garnered considerable interest over

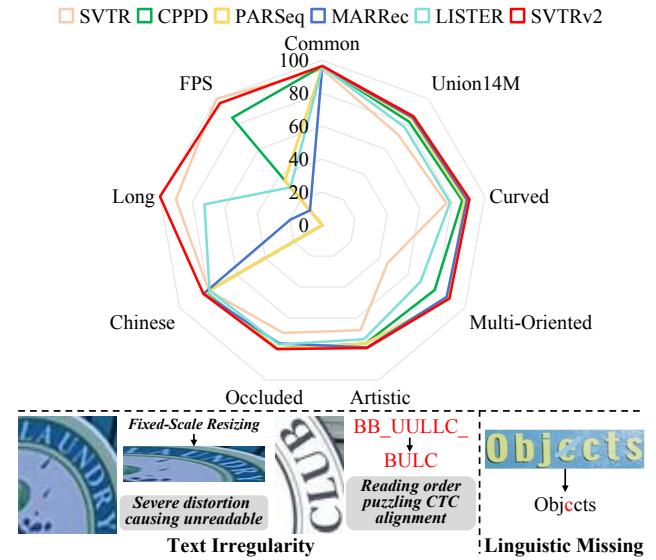


Figure 1. **Top:** comparison with previous methods [4, 8, 11, 12, 25] best in a single scenario, where long text recognition accuracy (Long) and FPS are normalized. Our SVTRv2 achieves the new state of the arts in every scenario except for FPS. Nevertheless, SVTRv2 is still the fastest compared to all the EDTRs. **Bottom:** challenges caused by text irregularity and linguistic missing.

decades. Unlike text from scanned documents, scene text often exists within complex natural scenarios, posing challenges such as background noise, text distortions, irregular layouts, artistic fonts [7], etc. To tackle these challenges, a variety of STR methods have been developed and they can be roughly divided into two categories, i.e., connectionist temporal classification (CTC)-based methods and encoder-decoder-based methods (EDTRs).

Typically, CTC-based methods [11, 23, 28, 39] employ a single visual model to extract image features and then apply a CTC-aligned linear classifier [16] to predict recognition results. This straightforward architecture provides advantages such as fast inference, which makes them especially popular in OCR applications. However, these mod-

*Corresponding Author

els struggle to handle text irregularity, i.e., text distortions, varying layouts, etc. As a consequence, advanced attention-based decoders are introduced as alternatives to the CTC classifier, leading to a series of EDTRs [4, 8, 9, 12, 15, 17, 18, 29, 32, 34, 36, 38, 40, 46–49, 51, 52, 54, 55, 57–59, 62, 63]. These attention-based decoders show appealing performance in integrating multi-modal cues, including visual [12, 48, 52, 58], linguistic [15, 36, 38, 55], and positional [8, 57, 62] ones, which are largely missed in current CTC models. This integration enables EDTRs to perform more effectively in complex scenarios. As depicted in the top of Fig. 1, compared to SVTR [11], a leading CTC model that is adopted by famous commercial OCR engines [28], EDTRs achieve superior results in English and Chinese benchmarks [6, 25], covering challenging scenarios such as curved, multi-oriented, occluded, and artistic text.

Nevertheless, EDTRs are commonly built upon complex architectures, thus sacrificing the inference speed (FPS), as shown in the top of Fig. 1. In addition, besides slower inference speed, EDTRs do not handle long text well. Even LISTER [8], an EDTR dedicated to long text recognition, performs worse than SVTR [11]. Since fast response and recognizing long text are both important for many applications, the OCR community has to face the dilemma that no model excels in accuracy, speed and versatility. When selecting either CTC-based models or EDTRs, users have to accept that the model is inferior in some aspects.

The inferior accuracy of CTC models can be attributed to two primary factors. First, these models struggle with irregular text, as CTC alignment presumes that the text appears in a near canonical left-to-right order [2, 7], which is not always true, particularly in complex scenarios. Second, CTC models seldom encode linguistic information, which is typically accomplished by the decoder of EDTRs. While recent advancements deal with the two issues by employing text rectification [32, 40, 61], developing 2D CTC [44], utilizing masked image modeling [48, 58], etc., the accuracy gap between CTC and EDTRs remains significant, indicating that novel solutions still need to be investigated.

In this paper, we aim to build more powerful CTC models by better handling text irregularity and integrating linguistic context. For the former, we address this challenge by first extracting discriminative features and then better aligning them. First, existing methods uniformly resize text images with various shapes to a fixed size before feeding into the visual model. We question the rationality of this resizing, which easily causes severe distortion of the text, making it unreadable to humans, as shown in the bottom-left of Fig. 1. To this end, we propose a multi-size resizing (MSR) strategy to adaptively resize text images according to their aspect ratios, thus minimizing text distortion and ensuring the discrimination of the extracted visual features. Second, irregular text may be rotated significantly, and the

character arrangement does not align with the reading order of the text, causing the puzzle for CTC alignment, as shown in the bottom-centre example in Fig. 1. To solve this, we introduce a feature rearrangement module (FRM) which rearranges visual features with a horizontal rearrangement, and then identifying and prioritizing relevant vertical features. FRM maps 2D visual features into a sequence aligned with the text’ reading order, thus effectively alleviating the alignment puzzle. Consequently, CTC models integrating MSR and FRM can recognize irregular text effectively, without using rectification modules or attention-based decoders.

As for the latter, the mistakenly recognized example shown in the bottom-right of Fig. 1 clearly highlights the necessity of integrating linguistic information. Since CTC models directly classify visual features, we have to endow the visual model with linguistic context modeling capability, which is less discussed previously. To this end, inspired by guided training of CTC (GTC) [23, 28] and string matching-based recognition [13], we propose a semantic guidance module (SGM), which devises a new scheme that solely leverages surrounding string context to recognize target characters during training. This approach effectively guides the visual model to learn to perceive the linguistic context without rely on decoders. During inference, SGM can be omitted and would not increase the inference cost.

With these contributions, we develop SVTRv2, a novel CTC-based method whose recognition ability has been largely enhanced, while still maintaining a simple architecture and fast inference. To thoroughly validate SVTRv2, we conducted extensive ablation and comparative experiments on benchmarks including standard regular and irregular text [2], occluded scene text [48], the recent Union14M-L benchmarks [25], long text [13], and Chinese [6] text. The results demonstrate that SVTRv2 consistently outperforms all the compared EDTRs across the evaluated scenarios in terms of accuracy and speed, highlighting its effectiveness and broad applicability.

In addition, recent advances [4, 25, 37] revealed the importance of large-scale real-world datasets in improving STR model performance. However, many STR methods primarily derived from synthetic data [19, 24], which fail to fully represent real-world complexities and lead to performance limitations, particularly on challenging scenarios. Notably, we observe that the existing real-word datasets [4, 25, 37] suffer from data leakage, and the results reported in [25] should be updated. As a result, we introduce *U14M-Filter*, a rigorously filtered version of the real-world training dataset Union14M-L [25]. We systematically reproduced and retrained 24 mainstream STR methods from scratch based on *U14M-Filter*. These methods are also thoroughly evaluated across various STR benchmarks. Their accuracy, model size, and inference time constitute a comprehensive and reliable new benchmark for future reference.

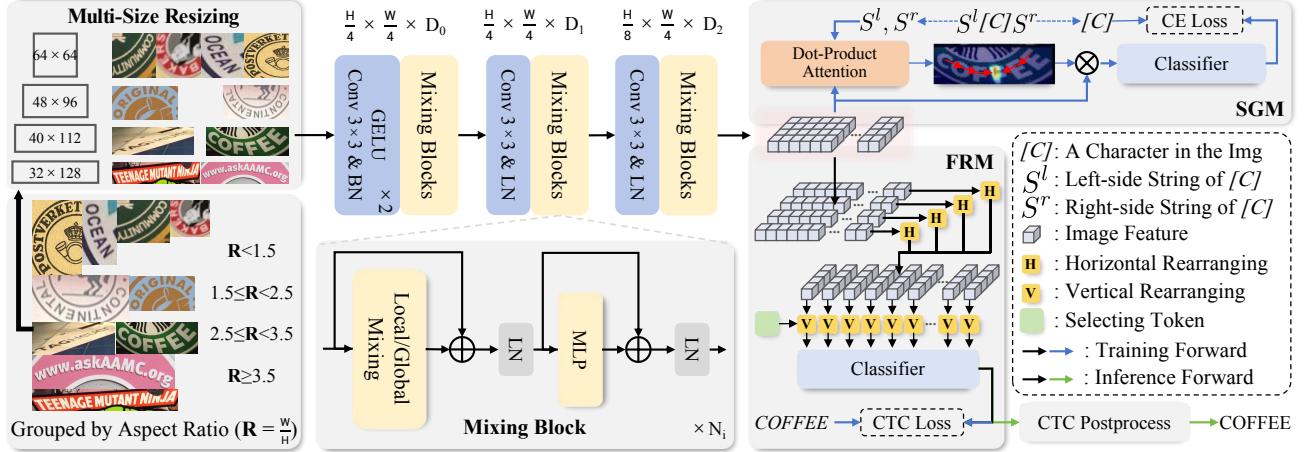


Figure 2. An illustrative overview of SVTRv2. The text is first resized according to multi-size resizing (MSR), then experiences feature extraction. During training both the semantic guidance module (SGM) and feature rearrangement module (FRM) are employed, which are responsible for linguistic context modeling and CTC-oriented feature rearrangement, respectively. Only FRM is retained during inference.

2. Related Work

Irregular text recognition [1, 25, 35] has posed a significant challenge in STR due to the diverse variation of text instances, where CTC-based methods [11, 23, 28, 39] are often less effective. To address this, some methods [11, 36, 40, 54, 59, 61, 62] incorporate rectification modules [32, 40, 61] that aim to transform irregular text into more regular format. Alternatively, more methods utilize attention-based decoders [29, 38, 47], which employ the attention mechanism to dynamically localize characters regardless of text layout, and thus less affected. However, these methods generally have tailored training hyper-parameters. For instance, the rectification modules [32, 40, 61] typically specify a fixed output image size (e.g. 32×128), which is not always a suitable choice. While attention-based decoders [29, 38, 47] generally set the maximum recognition length to 25 characters, thus longer text cannot be correctly recognized, as shown in Fig. 5.

Linguistic Context Modeling. There are several ways of modeling linguistic context. One major branch is auto-regressive (AR)-based STR methods [14, 25, 29, 38, 40, 47, 51, 52, 54, 57, 62, 63], which utilize previously decoded characters to model contextual cues. However, their inference speed is slow due to the character-by-character decoding nature. Some other methods [4, 15, 34, 55] integrate external language models to model the linguistic context and correct the recognition results. While effective, the linguistic context is purely text-based, making it challenging to adapt them to CTC models. There also are some studies [36, 48, 58] to modeling linguistic context with visual information only by using masked image modeling-based pretraining [3, 21]. However, they still depend on attention-based decoders to unleash the linguistic information, not in-

tegrating linguistic cues into the visual model, thus limiting their effectiveness in enhancing CTC models.

3. Methods

Fig. 2 illustrates the overview of SVTRv2. A text image is first resized by MSR to the closest aspect ratio, forming the input $X \in \mathbb{R}^{3 \times H \times W}$. X then experiences three consecutive feature extraction stages, yielding visual features $F \in \mathbb{R}^{\frac{H}{8} \times \frac{W}{4} \times D_2}$. During training, F is fed into both SGM and FRM. SGM guides SVTRv2 to model linguistic context, while FRM rearranges F into the character feature sequence $\tilde{F} \in \mathbb{R}^{\frac{W}{4} \times D_2}$, which is synchronized with the text reading order and aligns with the label sequence well. During inference, the SGM is discarded for efficiency.

3.1. Multi-Scale Resizing

Previous works typically resize irregular text images to a fixed size, such as 32×128, which, however, may cause undesired text distortion and severely affect the quality of extracted visual features. To address this issue, we propose a simple yet effective multi-size resizing (MSR) strategy that resizes text shapes based on the aspect ratio ($R = \frac{W}{H}$). Specifically, we define four specific sizes: [64, 64], [48, 96], [40, 112], and [32, $\lfloor R \rfloor \times 32$], respectively corresponding to aspect ratio: $R < 1.5$ (R_1), $1.5 \leq R < 2.5$ (R_2), $2.5 \leq R < 3.5$ (R_3), and $R \geq 3.5$ (R_4). Therefore, MSR allows text instances adaptively resized under the principles of roughly maintaining their aspect ratios, such that significant text distortion caused by resizing is almost eliminated. As a result, the quality of extracted visual features is guaranteed.

3.2. Visual Feature Extraction

Motivated by SVTR [11], the network architecture of SVTRv2 comprises three stages, with stage_i containing N_i mixing blocks, as illustrated in Fig. 2. To extract discriminative visual features, we devise two types of mixing blocks: local and global. Local mixing is implemented through two consecutive grouped convolutions, which are expected to capture local character features, such as edges, textures, and strokes. Meanwhile, global mixing is realized by the multi-head self-attention (MHSA) mechanism [42]. This mechanism performs global contextual modeling on features, thereby enhancing the model’s comprehension of inter-character relationships and the overall text image. Both the number of groups in the grouped convolution and the number of heads in MHSA are set to $\frac{D_i}{32}$. Similar to SVTR, by adjusting hyper-parameters N_i and D_i , we can derive three variants of SVTRv2 with different capacities, i.e., Tiny, Small, and Base, which are detailed in Sec. 7 of *Supplementary*.

3.3. Feature Rearranging Module

We propose a feature rearrangement module (FRM) to tackle the CTC alignment puzzle arising from rotated text. It rearranges the 2D features $\mathbf{F} \in \mathbb{R}^{(\frac{H}{8} \times \frac{W}{4}) \times D_2}$ into a feature sequence $\tilde{\mathbf{F}} \in \mathbb{R}^{\frac{W}{4} \times D_2}$ synchronized with the reading order of the text image. We regard this process as mapping the relevant features from $\mathbf{F}_{i,j} \in \mathbb{R}^{1 \times D_2}$ to $\tilde{\mathbf{F}}_m \in \mathbb{R}^{1 \times D_2}$, where $i \in \{1, 2, \dots, \frac{H}{8}\}$ and $j, m \in \{1, 2, \dots, \frac{W}{4}\}$. This rearrangement can be formalized by a matrix $\mathbf{M} \in \mathbb{R}^{\frac{W}{4} \times (\frac{H}{8} \times \frac{W}{4})}$, whereby $\tilde{\mathbf{F}}$ can be derived from $\mathbf{M} \times \mathbf{F}$.

Grounding in the fact that the degree of text curve and rotation can be decomposed into offset components in both horizontal and vertical directions, we propose to learn \mathbf{M} by using two distinct steps: horizontal rearrangement and vertical rearrangement. As described in Eq. 1, the horizontal one operates on $\mathbf{F}_i \in \mathbb{R}^{\frac{W}{4} \times D_2}$, each row in the feature map, to learn a horizontal rearrangement matrix $\mathbf{M}_i^h \in \mathbb{R}^{\frac{W}{4} \times \frac{W}{4}}$, where element $\mathbf{M}_{i,j,m}^h$ represents the probability that the horizontal rearranged feature $\mathbf{F}_{i,j}^h$ corresponds to the original feature $\mathbf{F}_{i,m}$. Based on the learned \mathbf{M}_i^h , the features in each row are rearranged on the horizontal direction. Subsequently, through a residual and MLP processing, we obtain $\mathbf{F}^h \in \mathbb{R}^{(\frac{H}{8} \times \frac{W}{4}) \times D_2}$.

$$\mathbf{M}_i^h = \sigma \left(\mathbf{F}_i W_i^q \left(\mathbf{F}_i W_i^k \right)^t \right) \quad (1)$$

$$\mathbf{F}_i^{h'} = \text{LN}(\mathbf{M}_i^h \mathbf{F}_i W_i^v + \mathbf{F}_i), \mathbf{F}_i^h = \text{LN}(\text{MLP}(\mathbf{F}_i^{h'}) + \mathbf{F}_i^{h'})$$

where $W_i^q, W_i^k, W_i^v \in \mathbb{R}^{D_2 \times D_2}$ are learnable weights, σ is the Softmax function, and $\mathbf{F}^h = \{\mathbf{F}_1^h, \mathbf{F}_2^h, \dots, \mathbf{F}_{\frac{H}{8}}^h\}$.

In the following vertical rearrangement, we introduce a selecting token, denoted as $\mathbf{T}^s \in \mathbb{R}^{1 \times D_2}$, which simultaneously attends to each column of features $\mathbf{F}_{:,j}^h \in \mathbb{R}^{\frac{H}{8} \times D_2}$

within \mathbf{F}^h to learn a vertical rearrangement matrix $\mathbf{M}_j^v \in \mathbb{R}^{1 \times \frac{H}{8}}$, as detailed in Eq. 2. The element $\mathbf{M}_{j,i}^v$ represents the probability that the vertical rearranged feature $\mathbf{F}_j^v \in \mathbb{R}^{1 \times D_2}$ corresponds to $\mathbf{F}_{i,j}^h$. Moreover, all column features share a single selecting token, rather than assigning a unique selecting token to each column feature at different locations. This scheme allows the model to generalize to longer text sequences, even when the number of column features exceeds the number seen during training, thereby facilitating the effective recognition of long text.

$$\mathbf{M}_j^v = \sigma \left(\mathbf{T}^s \left(\mathbf{F}_{:,j}^h W_j^k \right)^t \right), \mathbf{F}_j^v = \mathbf{M}_j^v \mathbf{F}_{:,j}^h W_j^v \quad (2)$$

where $W_j^q, W_j^k, W_j^v \in \mathbb{R}^{D_2 \times D_2}$ are learnable weights.

We denote $\mathbf{F}^v = \{\mathbf{F}_1^v, \mathbf{F}_2^v, \dots, \mathbf{F}_{\frac{W}{4}}^v\} \in \mathbb{R}^{\frac{W}{4} \times D_2}$ as the rearranged feature sequence $\tilde{\mathbf{F}}$. Then, the predicted character sequence $\tilde{\mathbf{Y}}_{ctc} \in \mathbb{R}^{\frac{W}{4} \times N_c}$ is obtained after $\tilde{\mathbf{F}}$ passes through the classifier $\tilde{\mathbf{Y}}_{ctc} = \tilde{\mathbf{F}} W^{ctc}$, and further aligned with the label sequence \mathbf{Y} using the CTC rule, where $W^{ctc} \in \mathbb{R}^{D_2 \times N_c}$ is the learnable weight of the classifier. Furthermore, to intuitively clarify the role of FRM, we rewrite the mapping relation between $\tilde{\mathbf{F}}$ and the original feature \mathbf{F} by the following procedure:

$$\begin{aligned} \mathbf{F}_{i,j}^h &= \mathbf{M}_{i,j}^h \times \mathbf{F}_i = \sum_{w=1}^{\frac{W}{4}} \mathbf{M}_{i,j,w}^h \mathbf{F}_{i,w} \\ \mathbf{F}_j^v &= \mathbf{M}_j^v \times \mathbf{F}_j^h = \sum_{i=1}^{\frac{H}{8}} \mathbf{M}_{i,j}^v \mathbf{F}_{i,j}^h \\ &= \sum_{i=1}^{\frac{H}{8}} \mathbf{M}_{i,j}^v \sum_{w=1}^{\frac{W}{4}} \mathbf{M}_{i,j,w}^h \mathbf{F}_{i,w} \\ &= \mathbf{M}_j \times \mathbf{F} \\ \mathbf{M}_j &= \mathbf{M}_j^v \odot \{\mathbf{M}_{1,j}^h, \mathbf{M}_{2,j}^h, \dots, \mathbf{M}_{\frac{H}{8},j}^h\} \in \mathbb{R}^{1 \times (\frac{H}{8} \times \frac{W}{4})} \\ \mathbf{M} &= \{\mathbf{M}_1, \mathbf{M}_2, \dots, \mathbf{M}_{\frac{W}{4}}\} \in \mathbb{R}^{\frac{W}{4} \times (\frac{H}{8} \times \frac{W}{4})} \\ \tilde{\mathbf{F}} &= \mathbf{F}^v = \mathbf{M} \times \mathbf{F} \in \mathbb{R}^{\frac{W}{4} \times D_2} \end{aligned} \quad (4)$$

As seen in Eq. 4, the visual features \mathbf{F} , which fully describe the text to be recognized but may be irregularly organized, are mapped to a more canonical $\tilde{\mathbf{F}}$. Different from existing text rectification models [32, 40], which apply geometric transformations at image space and often fail to correct severely irregular text, our FRM is carried out at feature space. It learns a condensed feature sequence $\tilde{\mathbf{F}}$ by mapping and rearranging the relevant cues from \mathbf{F} to proper positions in $\tilde{\mathbf{F}}$. By employing a structured way that the features are first rearranged horizontally and then vertically, FRM can accommodate text instances with significant irregularities such as large orientation variations, and obtaining features better aligned with the CTC classification.

3.4. Semantic Guidance Module

CTC models directly classify visual features to obtain recognition results. This scheme inherently requires that

the linguistic context must be incorporated into visual features, so that the CTC could benefit from. In light of this, we propose a semantic guidance module (SGM) as follows.

For each character c_i in a text image with character labels $\mathcal{Y} = \{c_1, c_2, \dots, c_L\}$, we define its contextual information as the surrounding left string $S_i^l = \{c_{i-l_s}, \dots, c_{i-1}\}$ and right string $S_i^r = \{c_{i+1}, \dots, c_{i+l_s}\}$, where l_s denotes the context window length. The SGM's role is to guide the visual model to integrate context from both S_i^l and S_i^r into visual features.

We describe the process using the left string S_i^l , with the same process applied symmetrically to the right string S_i^r . Firstly, the characters in S_i^l are mapped to string embeddings $E_i^l \in \mathbb{R}^{l_s \times D_2}$. Subsequently, these embeddings are encoded to create a hidden representation $Q_i^l \in \mathbb{R}^{1 \times D_2}$, representing the context of the left-side string S_i^l . The attention map A_i^l is computed by applying a dot product between the hidden representation Q_i^l and the visual features F , transformed by learned weight matrices W^q and W^k . The detailed formulation is as follows:

$$\begin{aligned} Q_i^l &= \text{LN} \left(\sigma \left(\mathbf{T}^l W^q (E_i^l W^k)^t \right) E_i^l W^v + \mathbf{T}^l \right) \quad (5) \\ A_i^l &= \sigma \left(Q_i^l W^q (F W^k)^t \right), \quad F_i^l = A_i^l F W^v \end{aligned}$$

where $\mathbf{T}^l \in \mathbb{R}^{1 \times D_2}$ represents a pre-defined token that encodes the left-side string. The attention map A_i^l is used to weight the visual features F , producing the feature $F_i^l \in \mathbb{R}^{1 \times D_2}$ corresponding to character c_i . After processing through the classifier $\tilde{Y}_i^l = F_i^l W^{sgm}$, the predicted class probabilities $\tilde{Y}_i^l \in \mathbb{R}^{1 \times N_c}$ for c_i is obtained to calculate the cross-entropy loss with the label c_i , where $W^{sgm} \in \mathbb{R}^{D_2 \times N_c}$ is learnable weights of the classifier.

The weight of the attention map A_i^l records the relevance of Q_i^l to the visual features F , and moreover, Q_i^l represents the context of string S_i^l . So only when the visual model incorporates the context from S_i^l into the visual features of the target character c_i , the attention map A_i^l can maximize the relevance between Q_i^l and visual features of the character c_i , thereby accurately highlighting the corresponding position of character c_i , as shown in Fig. 3. A similar process applies to the right-side string S_i^r , where the corresponding attention map A_i^r and visual feature F_i^r contribute to the prediction \tilde{Y}_i^r . By leveraging the above scheme during training, SGM effectively guides the visual model in integrating the linguistic context into visual features. Consequently, even when SGM is not used during inference, the linguistic context can still be maintained alongside the visual features, and enhancing the accuracy of CTC models. In contrast, previous methods, such as VisionLAN [48] and LPV [58], despite modeling linguistic context using visual features, still rely on attention-based decoders to unleash linguistic information, a process that is incompatible with CTC models.

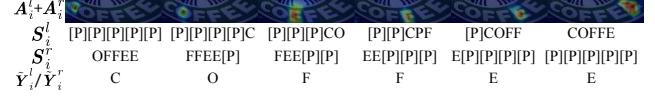


Figure 3. Visualization of attention maps when recognizing the target character by string matching on both sides, where l_i is set to 5. [P] denotes the padding symbol.

3.5. Optimization Objective

During training, The optimization objective is to minimize the loss \mathcal{L} , which comprises \mathcal{L}_{ctc} and \mathcal{L}_{sgm} , as listed below:

$$\begin{aligned} \mathcal{L}_{ctc} &= \text{CTCLoss}(\tilde{Y}_{ctc}, \mathcal{Y}) \quad (6) \\ \mathcal{L}_{sgm} &= \frac{1}{2L} \sum_{i=1}^L (\text{ce}(\tilde{Y}_i^l, \mathbf{c}_i) + \text{ce}(\tilde{Y}_i^r, \mathbf{c}_i)) \\ \mathcal{L} &= \lambda_1 \mathcal{L}_{ctc} + \lambda_2 \mathcal{L}_{sgm} \end{aligned}$$

where ce represents the cross-entropy loss, λ_1 and λ_2 are weighting parameters setting to 0.1 and 1, respectively.

4. Experiments

4.1. Datasets and Implementation Details

We evaluate SVTRv2 across multiple benchmarks covering diverse scenarios. They are: 1) six common regular and irregular benchmarks (*Com*), including ICDAR 2013 (*IC13*) [27], Street View Text (*SVT*) [45], IIIT5K-Words (*IIIT5K*) [33], ICDAR 2015 (*IC15*) [26], Street View Text-Perspective (*SVTP*) [35] and *CUTE80* [1]. For IC13 and IC15, we use the versions with 857 and 1811 images, respectively; 2) the test set of the recent Union14M-L benchmark (*U14M*) [25], which includes seven challenging subsets: *Curve*, *Multi-Oriented (MO)*, *Artistic*, *Contextless (Cless)*, *Salient*, *Multi-Words (MW)* and *General*; 3) occluded scene text dataset (*OST*) [48], which is categorized into two subsets based on the degree of occlusion: weak occlusion (*OST*_w) and heavy occlusion (*OST*_h); 4) long text benchmark (*LTB*) [13], which includes 3376 samples of text length from 25 to 35; 5) the test set of BCTR [6], a Chinese text recognition benchmark with four subsets: *Scene*, *Web*, *Document (Doc)* and *Hand-Writing (HW)*.

For English recognition, we train models on real-world datasets, from which the models exhibit stronger recognition capability [4, 25, 37]. There are three large-scale real-world training sets, i.e., the *Real* dataset [4], *REBU-Syn* [37], and the training set of *Union14M-L (U14M-Train)* [25]. However, they all overlap with *U14M* (detailed in Sec. 8 in *Supplementary*) across the seven subsets, leading to data leakage, which makes them unsuitable for training models. To resolve this, we introduce a filtered version of *Union14M-L* training set, termed as *U14M-Filter*, by filtering these overlapping instances. This new dataset is used to train SVTRv2 and 24 mainstream methods we reproduced.

For Chinese recognition, we train models on the training set of *BCTR* [6]. Unlike previous methods that train separately for each subset, we trained the model on an integrated dataset and then evaluated it on the four subsets.

We use AdamW optimizer [30] with a weight decay of 0.05 for training. The LR is set to 6.5×10^{-4} and batchsize is set to 1024. One cycle LR scheduler [43] with 1.5/4.5 epochs linear warm-up is used in all the 20/100 epochs, where a/b means a for English and b for Chinese. For English model, we take SVTRv2 without SGM as the pre-trained and fine-tuned SVTRv2 with SGM with the same above settings. Word accuracy is used as the evaluation metric. Data augmentation like rotation, perspective distortion, motion blur and gaussian noise, are randomly performed and the maximum text length is set to 25 during training. The size of the character set N_c is set to 94 for English and 6624 [28] for Chinese. In experiments, SVTRv2 means SVTRv2-B unless specified. All models are trained with mixed-precision on 4 RTX 4090 GPUs.

4.2. Ablation Study

Effectiveness of MSR. We group the *Curve* and *MO* text in *U14M* based on the aspect ratio \mathbf{R}_i . As shown in Tab. 1, the majority of irregular texts fall within \mathbf{R}_1 and \mathbf{R}_2 , where they are particularly prone to distortion when resized to a fixed size (see *Fixed* $_{32 \times 128}$ in Fig. 4). In contrast, MSR demonstrates significant improvements of 15.3% in \mathbf{R}_1 and 5.2% in \mathbf{R}_2 compared to *Fixed* $_{32 \times 128}$. Meanwhile, a large fixed-size *Fixed* $_{64 \times 256}$, although improving the accuracy compared to the baseline, still performs worse than our MSR by clear margins. The results strongly confirm our hypothesis that undesired resizing would hurt the recognition. Our MSR effectively mitigates this issue, providing better visual features thus enhancing the recognition accuracy.

Effectiveness of FRM. We ablate the two rearrangement sub-modules (Horizontal (H) rearranging and Vertical (V) rearranging). As shown in Tab. 1, compared to without FRM (w/o FRM), they individually improve accuracy by 2.03% and 0.71% on *MO*, and they together result in a 2.46% gain. Additionally, we explore using a Transformer block (+ TF₁) to learn the rearrangement matrix holistically, whose effectiveness is less obvious. The most probable reason is that this scheme does not well distinguish between vertical and horizontal orientations. In contrast, FRM performs feature rearrangement in both directions, making it highly sensitive to text irregularity, and thus facilitating accurate CTC alignment. As shown in the left five cases in Fig. 4, FRM successfully recognizes reverse instances, providing strong evidence of FRM’s effectiveness.

Effectiveness of SGM. As illustrated in Tab. 2, SGM achieves 0.41% and 2.28% increase on *Com* and *U14M*, respectively, while gains a 5.11% improvement on *OST*. Since *OST* frequently suffers from missing a portion of

		\mathbf{R}_1	\mathbf{R}_2	\mathbf{R}_3	\mathbf{R}_4	<i>Curve</i>	<i>MO</i>	<i>Com</i>	<i>U14M</i>
SVTRv2 (+MSR+FRM)		87.4	88.3	86.1	87.5	88.17	86.19	96.16	83.86
SVTRv2 (w/o both)		70.5	81.5	82.8	84.4	82.89	65.59	95.28	77.78
vs.	<i>Fixed</i> $_{32 \times 128}$	72.1	83.1	84.1	85.6	83.18	68.71	95.56	78.87
MSR	<i>Padding</i> $_{32 \times W}$	52.1	71.3	82.3	87.4	71.06	51.57	94.70	71.82
(+FRM)	<i>Fixed</i> $_{64 \times 256}$	76.6	81.6	81.9	80.2	85.70	67.49	95.07	79.03
vs.	w/o FRM	85.7	86.3	86.0	85.5	87.35	83.73	95.44	82.22
FRM	+ H rearranging	87.0	87.1	86.3	85.5	88.05	85.76	95.98	82.94
(+MSR)	+ V rearranging	85.0	87.6	88.5	85.5	88.01	84.44	95.66	82.70
	+ TF ₁	86.4	86.3	87.5	86.1	87.51	85.50	95.60	82.49
<hr/>									
-									
ResNet+TF ₃									
FocalNet-B									
ConvNeXtV2									
ViT-S									
SVTR-B									
<hr/>									
+FRM									
ResNet+TF ₃									
FocalNet-B									
ConvNeXtV2									
ViT-S									
SVTR-B									
<hr/>									
+MSR									
ResNet+TF ₃									
FocalNet-B									
ConvNeXtV2									
<hr/>									
-/+SGM									
<i>OST</i> _w <i>OST</i> _h Avg									
<i>OST</i> _w [*] <i>OST</i> _h [*] Avg									
<i>Com*</i> <i>U14M*</i>									

Table 1. Ablations on MSR and FRM (top) and assessing MSR, FRM, and SGM across visual models (lower). * means with SGM.

	Method	<i>OST</i> _w	<i>OST</i> _h	Avg	<i>Com</i>	<i>U14M</i>
Linguistic context modeling	w/o SGM	82.86	66.97	74.92	96.16	83.86
	SGM	86.26	73.80	80.03	96.57	86.14
	GTC [23]	83.07	68.32	75.70	96.01	84.33
	ABINet [15]	83.07	67.54	75.31	96.25	84.17
	VisionLAN [48]	83.25	68.97	76.11	96.39	84.01
	PARSEq [4]	83.85	69.24	76.55	96.21	84.72
<hr/>						
MAERec [25]						
83.21						
69.69						
76.45						
96.47						
84.69						

Table 2. Comparison of the proposed SGM with other language models in linguistic context modeling on *OST*.

characters, this notable gain implies that the linguistic context has been successfully established. For comparison, we also employ GTC [23] and four popular language decoders [4, 15, 25, 48] to substitute for our SGM. However, there is no much difference between the gains obtained from *OST* and the other two datasets (*Com* and *U14M*). This suggests that SGM offers a distinct advantage in integrating linguistic context into visual features, and significantly improving the recognition accuracy of CTC models. The five cases on the right side of Fig. 4 showcase that SGM facilitates SVTRv2 to accurately decipher occluded characters, achieving comparable results with PARSeq [4], which

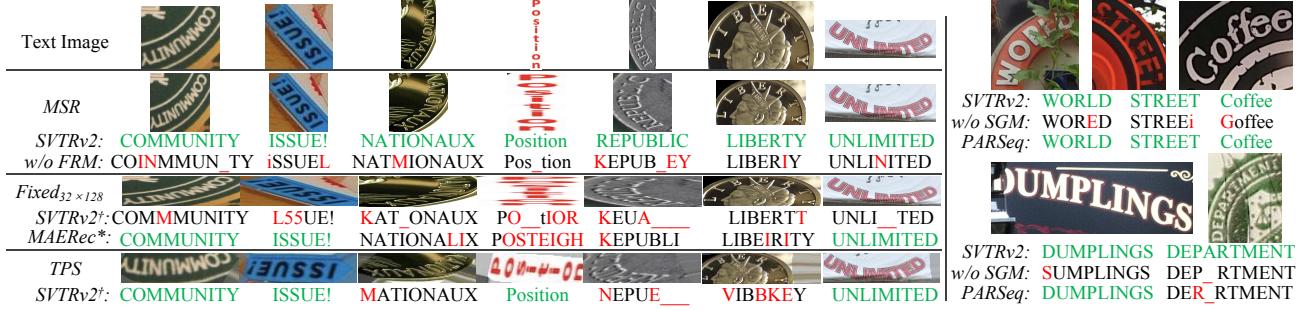


Figure 4. Qualitative comparison of SVTRv2 with previous methods on irregular and occluded text. \dagger means that SVTRv2 utilizes the fixed-scale ($Fixed_{32 \times 128}$) or rectification module (TPS) as the resize strategy. $MAERec^*$ means that $SVTRv2^\dagger$ integrates with the attention-based decoder from the previous best model, i.e. MAERec [25], such a decoder is widely employed in [5, 31, 38, 51, 52, 54, 56]. Green, red, and blue denotes correctly, wrongly and missed recognition, respectively.

IIT5k	SVT	ICDAR2013	ICDAR2015	SVTP	CUTE80	Curve	Multi-Oriented	Artistic	Contextless	Salient	Multi-Words	General										
Method	Venue	Encoder	Common Benchmarks				Avg	Union14M Benchmarks				Avg	LTB	OST	Size	FPS						
ASTER [40]	TPAMI19	ResNet+LSTM	96.1	93.0	94.9	86.1	87.9	92.0	91.70	70.9	82.2	56.7	62.9	73.9	58.5	76.3	68.75	0.1	61.9	19.0	67.1	
NRTR [38]	ICDAR19	Stem+TF ₆	98.1	96.8	97.8	88.9	93.3	94.4	94.89	67.9	42.4	66.5	73.6	66.4	77.2	78.3	67.46	0.0	74.8	44.3	17.3	
MORAN [32]	PR19	ResNet+LSTM	96.7	91.7	94.6	84.6	85.7	90.3	90.61	51.2	15.5	51.3	61.2	43.2	64.1	69.3	50.82	0.1	57.9	17.4	59.5	
SAR [29]	AAAI19	ResNet+LSTM	98.1	93.8	96.7	86.0	87.9	95.5	93.01	70.5	51.8	63.7	73.9	64.0	79.1	75.5	68.36	0.0	60.6	57.5	15.8	
DAN [47]	AAAI20	ResNet+FPN	97.5	94.7	96.5	87.1	89.1	94.4	93.24	74.9	63.3	63.4	70.6	70.2	71.1	76.8	70.05	0.0	61.8	27.7	99.0	
SRN [55]	CVPR20	ResNet+FPN	97.2	96.3	97.5	87.9	90.9	96.9	94.45	78.1	63.2	66.3	65.3	71.4	58.3	76.5	68.43	0.0	64.6	51.7	67.1	
SEED [36]	CVPR20	ResNet+LSTM	96.5	93.2	94.2	87.5	88.7	93.4	92.24	69.1	80.9	56.9	63.9	73.4	61.3	76.5	68.87	0.1	62.6	24.0	65.4	
AutoSTR [59]	ECCV20	NAS+LSTM	96.8	92.4	95.7	86.6	88.2	93.4	92.19	72.1	81.7	56.7	64.8	75.4	64.0	75.9	70.09	0.1	61.5	6.0	82.6	
RoScanner [57]	ECCV20	ResNet	98.5	95.8	97.7	88.2	90.1	97.6	94.65	79.4	68.1	70.5	79.6	71.6	82.5	80.8	76.08	0.0	68.6	48.0	64.1	
ABINet [15]	CVPR21	ResNet+TF ₃	98.5	98.1	97.7	90.1	94.1	96.5	95.83	80.4	69.0	71.7	74.7	77.6	76.8	79.8	75.72	0.0	75.0	36.9	73.0	
VisionLAN [48]	ICCV21	ResNet+TF ₃	98.2	95.8	97.1	88.6	91.2	96.2	94.50	79.6	71.4	67.9	73.7	76.1	73.9	79.1	74.53	0.0	66.4	32.9	93.5	
PARSeq [4]	ECCV22	ViT-S	98.9	98.1	98.4	90.1	94.3	98.6	96.40	87.6	88.8	76.5	83.4	84.4	84.3	84.9	84.26	0.0	79.9	23.8	52.6	
MATRN [34]	ECCV22	ResNet+TF ₃	98.8	98.3	97.9	90.3	95.2	97.2	96.29	82.2	73.0	73.4	76.9	79.4	77.4	81.0	77.62	0.0	77.8	44.3	46.9	
MGP-STR [46]	ECCV22	ViT-B	97.9	97.8	97.1	89.6	95.2	96.9	95.75	85.2	83.7	72.6	75.1	75.1	79.8	71.1	83.1	78.65	0.0	78.7	148	120
CPPD [12]	Preprint	SVTR-B	99.0	97.8	98.2	90.4	94.0	99.0	96.40	86.2	78.7	76.5	82.9	83.5	81.9	83.5	81.91	0.0	79.6	27.0	125	
LPV [58]	IJCAI23	SVTR-B	98.6	97.8	98.1	89.8	93.6	97.6	95.93	86.2	78.7	75.8	80.2	82.9	81.6	82.9	81.20	0.0	77.7	30.5	82.6	
MAERec [25]	ICCV23	ViT-S	99.2	97.8	98.2	90.4	94.3	98.3	96.36	89.1	87.1	79.0	84.2	86.3	85.9	84.6	85.17	9.8	76.4	35.7	17.1	
LISTER [8]	ICCV23	FocalNet-B	98.8	97.5	98.6	90.0	94.4	96.9	96.03	78.7	68.8	73.7	81.6	74.8	82.4	83.5	77.64	36.3	77.1	51.1	44.6	
CDistNet [62]	IJCV24	ResNet+TF ₃	98.7	97.1	97.8	89.6	93.5	96.9	95.59	81.7	77.1	72.6	78.2	79.9	79.7	81.1	78.62	0.0	71.8	43.3	15.9	
CAM [54]	PR24	ConvNeXtV2	98.2	96.1	96.6	89.0	93.5	96.2	94.94	85.4	89.0	72.0	75.4	84.0	74.8	83.1	80.52	0.7	74.2	58.7	28.6	
BUSNet [49]	AAAI24	ViT-S	98.3	98.1	97.8	90.2	95.3	96.5	96.06	83.0	82.3	70.8	77.9	78.8	71.2	82.6	78.10	0.0	78.7	32.1	83.3	
OTE [52]	CVPR24	SVTR-B	98.6	96.6	98.0	90.1	94.0	97.2	95.74	86.0	75.8	74.6	74.7	81.0	65.3	82.3	77.09	0.0	77.8	20.3	55.2	
C	CRNN [39]	TPAMI16	ResNet+LSTM	95.8	91.8	94.6	84.9	83.1	91.0	90.21	48.1	13.0	51.2	62.3	41.4	60.4	68.2	49.24	47.2	58.0	16.2	172
T	SVTR [11]	IJCAI22	SVTR-B	98.0	97.1	97.3	88.6	90.7	95.8	94.58	76.2	44.5	67.8	78.7	75.2	77.9	77.8	71.17	45.1	69.6	18.1	161
C	SVTRv2	-	SVTRv2-T	98.6	96.6	98.0	88.4	90.5	96.5	94.78	83.6	76.0	71.2	82.4	77.2	82.3	80.7	79.05	47.8	71.4	5.1	201

Table 3. All the models and SVTRv2 are trained on *U14M-Filter*. TF_n denotes the n -layer Transformer block [42]. $Size$ denotes the model size (M). FPS is uniformly measured on one NVIDIA 1080Ti GPU. In addition, we discuss the results of SVTRv2 trained on synthetic datasets [19, 24] in *Supplementary*.

is equipped with an advanced permuted language model.

Adaptability to different visual models. We further examine MSR, FRM, and SGM on five frequently used visual models [10, 11, 20, 50, 53]. As presented in the bottom part of Tab. 1, these modules consistently enhance the performance (ViT [10] and SVTR [11] employ absolute positional coding and do not compatible with MSR). When both FRM and MSR modules incorporated, ResNet+TF₃ [20], Focal-

Net [53], and ConvNeXtV2 [54] exhibit significant accuracy improvements, either matching or even exceeding the accuracy of their EDTR counterparts (see Tab. 3). The results highlight the versatility of the three proposed modules.

4.3. Comparison with State-of-the-arts

To demonstrate the effectiveness of SVTRv2 in English, we compare it with 24 popular STR methods. All the mod-

els are tested on the newly constructed *U14M* and the results are given in Tab. 3. SVTRv2-B ranks the top in 12 of the 15 evaluated scenarios, It almost outperforms all the EDTRs in every scenario, showing a clear accuracy advantage. Meanwhile, it still enjoys a small model size and a significant speed advantage. Specifically, compared to MAERec, the best-performed existing model on *U14M*, SVTRv2-B shows an accuracy improvement of 0.97% and 8 \times faster inference speed. Compared to CPPD, which is known for its accuracy-speed tradeoff, SVTR-B runs faster than 10%, along with a 4.23% accuracy increase on *U14M*. Regarding OST, as illustrated in the right part of Fig. 4, SVTR-B relies solely on a single visual model but achieves comparable accuracy to PARSeq, which employed the advanced permuted language model and is the best-performed existing model on OST. In the case of long text recognition, where a large portion of EDTRs are incapable of recognizing, SVTR-B outperforms LISTER, the best EDTR method on LTB, by 13%, demonstrating the remarkable scalability of SVTRv2. In addition, SVTRv2-T and SVTRv2-S, the two smaller models also show leading accuracy compared with models of similar sizes, offering solutions with different accuracy-speed tradeoff.

Two observations are derived when looking into the results on *Curve* and *MO*. First, SVTRv2 models significantly surpass existing CTC models. For example, compared to SVTR-B, SVTRv2-B gains prominent accuracy improvements of 14.4% and 44.5%, respectively. Second, as shown in Tab. 4, comparing with previous methods employing the rectification modules [11, 36, 40, 54, 59, 61, 62] and the attention-based decoder [5, 25, 29, 31, 38, 47, 51, 52, 54, 56] to recognize irregular text, SVTRv2 also performs better than these methods on *Curve* and *MO*. In Fig. 4, the rectification module (*TPS*) and the attention-based decoder (*MAERec**) do not recognize the extremely curved and rotated text correctly, in contrast, SMTR successes. Moreover, as demonstrated by the results on LTB in Tab. 4 and Fig. 5, *TPS* and *MAERec** both do not effectively recognize long text, while SVTRv2 circumvents this limitation. These results indicate that our proposed modules successfully address the challenge of handling irregular text that existing CTC models encountered, while preserving CTC’s proficiency in recognizing long text.

SVTRv2 models also exhibit strong performance in Chinese text recognition (see Tab. 5), where SVTRv2-B achieve state of the art. The result underscores its great adaptability to different languages. To sum, we evaluate SVTRv2 across a wide range of scenarios. The results consistently confirm that this CTC model beats leading EDTRs.

5. Conclusion

In this paper, we have presented SVTRv2, an accurate and efficient CTC-based STR method. SVTRv2 is featured by

	R₁	R₂	R₃	R₄	<i>Curve</i>	<i>MO</i>	<i>Com</i>	<i>U14M</i>	<i>LTB</i>
SVTRv2	90.8	89.0	90.4	91.0	90.64	89.04	96.57	86.14	50.2
TPS	SVTR [11]	86.8	82.3	77.3	75.7	82.19	86.12	94.62	78.44
	SVTRv2	89.5	85.1	78.4	83.8	84.71	88.97	94.62	79.94
MAE*	SVTR [11]	81.3	87.6	87.6	88.3	87.88	78.74	96.32	83.23
	SVTRv2	88.0	88.9	89.4	88.3	89.96	87.56	96.42	85.67

Table 4. SVTRv2 and SVTR comparisons on irregular text and LTB, where the rectification module (TPS) and the attention-based decoder (MAERec*) are employed.

⁴⁵ SWEET LADY LOOK DOWN FROM THY WINDOW ON ME?
CRNN: "SWEET LADY IDOK DOWN FROM THY WOYDOW ON XE"
SVTR: "SWEET LADY LOOK DOWN FRO_ THY W_NDOW OW ME,
LISTER: "SWEET LADY LOOK _____ WINDOW ON ME?"
SVTRv2: " SWEET LADY LOOK DOWN FROM THY WINDOW ON ME "
w/ TPS: C
w/ MAERec*: "mayLosMocanos.com
EDITED WITH INTRODUCTION BY ROY TORGESON
CRNN: EDITED WITH INTRODUCTION BY ROY TORGESON
SVTR: EDITED W_ TH INTRODUCTION BY ROY TORGESON
LISTER: EDITED WITH INTRODUCTION B_ O_ TORGESON
SVTRv2: EDITED WITH INTRODUCTION BY ROY TORGESON
w/ TPS: CIYYS
w/ MAERec*: EDITED WITH IN _____ I N _____ G SON

Figure 5. Long text recognition results. *TPS* and *MAERec** denote SVTRv2 integrated with TPS and the decoder of MAERec.

Method	<i>Scene</i>	<i>Web</i>	<i>Doc</i>	<i>HW</i>	Avg	<i>Scene_{L>25}</i>	<i>Size</i>
ASTER [40]	61.3	51.7	96.2	37.0	61.55	-	27.2
MORAN [32]	54.6	31.5	86.1	16.2	47.10	-	28.5
SAR [29]	59.7	58.0	95.7	36.5	62.48	-	27.8
SEED [36]	44.7	28.1	91.4	21.0	46.30	-	36.1
MASTER [31]	62.8	52.1	84.4	26.9	56.55	-	62.8
ABINet [15]	66.6	63.2	98.2	53.1	70.28	-	53.1
TransOCR [5]	71.3	64.8	97.1	53.0	71.55	-	83.9
CCR-CLIP [56]	71.3	69.2	98.3	60.3	74.78	-	62.0
DCTC [60]	73.9	68.5	99.4	51.0	73.20	-	40.8
CAM [54]	76.0	69.3	98.1	59.2	76.80	-	135
PARSeq* [4]	84.2	82.8	99.5	63.0	82.37	0.0	28.9
CPPD* [12]	82.7	82.4	99.4	62.3	81.72	0.0	32.1
MAERec* [25]	84.4	83.0	99.5	65.6	83.13	4.1	40.8
LISTER* [8]	79.4	79.5	99.2	58.0	79.02	13.9	55.0
CRNN* [39]	63.8	68.2	97.0	46.1	68.76	37.6	19.5
SVTR-B* [11]	77.9	78.7	99.2	62.1	79.49	22.9	19.8
SVTRv2-T	77.8	78.8	99.2	62.0	79.45	47.8	6.8
SVTRv2-S	81.1	81.2	99.3	65.0	81.64	50.0	14.0
SVTRv2-B	83.5	83.3	99.5	67.0	83.31	52.8	22.5

Table 5. Results on Chinese text dataset. * denotes that the model is retrained using the same setting as SVTRv2 (Sec. 4.1).

developing the MSR and FRM modules to tackle the text irregular challenge, and devising the SGM module to endow linguistic context to the visual model. These upgrades maintain the simple architecture of CTC models, thus they remain quite efficient. More importantly, our thorough validation on multiple benchmarks demonstrates the effectiveness of SVTRv2. It achieves leading accuracy and inference

speed in various challenging scenarios covering regular, irregular, occluded, Chinese and long text, convincingly indicating that SVTRv2 has beat EDTRs in scene text recognition. In addition, we retrain 24 methods from scratch on *UI4M-Filter* without data leakage, constituting a comprehensive and reliable benchmark. We hope that SVTRv2 and this benchmark will further advance the development of the OCR community.

References

- [1] R. Anhar, S. Palaiahnakote, C. S. Chan, and C. L. Tan. A robust arbitrary text detection system for natural scene images. *Expert Syst. Appl.*, 41(18):8027–8048, 2014. [3](#), [5](#), [6](#)
- [2] J. Baek, G. Kim, J. Lee, S. Park, D. Han, S. Yun, S. J. Oh, and H. Lee. What is wrong with scene text recognition model comparisons? dataset and model analysis. In *ICCV*, pages 4714–4722, 2019. [2](#)
- [3] H. Bao, L. Dong, S. Piao, and F. Wei. BEiT: BERT pre-training of image transformers. In *ICLR*, 2022. [3](#)
- [4] D. Bautista and R.I Atienza. Scene text recognition with permuted autoregressive sequence models. In *ECCV*, pages 178–196, 2022. [1](#), [2](#), [3](#), [5](#), [6](#), [7](#), [8](#)
- [5] J. Chen, B. Li, and X. Xue. Scene Text Telescope: Text-focused scene image super-resolution. In *CVPR*, pages 12021–12030, 2021. [7](#), [8](#)
- [6] J. Chen, H. Yu, J. Ma, M. Guan, X. Xu, X. Wang, S. Qu, B. Li, and X. Xue. Benchmarking chinese text recognition: Datasets, baselines, and an empirical study. *CoRR*, abs/2112.15093, 2021. [2](#), [5](#), [6](#)
- [7] X. Chen, L. Jin, Y. Zhu, C. Luo, and T. Wang. Text recognition in the wild: A survey. *ACM Comput. Surv.*, 54(2):42:1–42:35, 2022. [1](#), [2](#)
- [8] C. Cheng, P. Wang, C. Da, Q. Zheng, and C. Yao. LISTER: Neighbor decoding for length-insensitive scene text recognition. In *ICCV*, pages 19484–19494, 2023. [1](#), [2](#), [7](#), [8](#), [5](#)
- [9] C. Da, P. Wang, and C. Yao. Levenshtein OCR. In *ECCV*, pages 322–338, 2022. [2](#), [5](#)
- [10] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. [7](#)
- [11] Y. Du, Z. Chen, C. Jia, X. Yin, T. Zheng, C. Li, Y. Du, and Y. Jiang. SVTR: Scene text recognition with a single visual model. In *IJCAI*, pages 884–890, 2022. [1](#), [2](#), [3](#), [4](#), [7](#), [8](#), [5](#)
- [12] Y. Du, Z. Chen, C. Jia, X. Yin, C. Li, Y. Du, and Y. Jiang. Context perception parallel decoder for scene text recognition. *CoRR*, abs/2307.12270, 2023. [1](#), [2](#), [7](#), [8](#), [5](#)
- [13] Y. Du, Z. Chen, C. Jia, X. Gao, and Y. Jiang. Out of length text recognition with sub-string matching. *CoRR*, abs/2407.12317, 2024. [2](#), [5](#)
- [14] Y. Du, Z. Chen, Y. Su, C. Jia, and Y. Jiang. Instruction-guided scene text recognition. *CoRR*, abs/2401.17851, 2024. [3](#)
- [15] S. Fang, H. Xie, Y. Wang, Z. Mao, and Y. Zhang. Read Like Humans: Autonomous, bidirectional and iterative language modeling for scene text recognition. In *CVPR*, pages 7098–7107, 2021. [2](#), [3](#), [6](#), [7](#), [8](#), [5](#)
- [16] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *ICML*, page 369–376, 2006. [1](#)
- [17] T. Guan, C. Gu, J. Tu, X. Yang, Q. Feng, Y. Zhao, and W. Shen. Self-Supervised implicit glyph attention for text recognition. In *CVPR*, pages 15285–15294, 2023. [2](#), [5](#)
- [18] Tongkun Guan, Wei Shen, Xue Yang, Qi Feng, Zekun Jiang, and Xiaokang Yang. Self-Supervised Character-to-Character distillation for text recognition. In *ICCV*, pages 19473–19484, 2023. [2](#), [5](#)
- [19] A. Gupta, A. Vedaldi, and A. Zisserman. Synthetic data for text localisation in natural images. In *CVPR*, pages 2315–2324, 2016. [2](#), [7](#), [5](#)
- [20] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. [7](#), [1](#)
- [21] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. B. Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, pages 15979–15988, 2022. [3](#)
- [22] A. G Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. MobileNets: Efficient convolutional neural networks for mobile vision applications. *CoRR*, abs/1704.04861, 2017. [1](#)
- [23] W. Hu, X. Cai, J. Hou, S. Yi, and Z. Lin. GTC: Guided training of ctc towards efficient and accurate scene text recognition. In *AAAI*, pages 11005–11012, 2020. [1](#), [2](#), [3](#), [6](#)
- [24] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman. Synthetic data and artificial neural networks for natural scene text recognition. *CoRR*, abs/1406.2227, 2014. [2](#), [7](#), [5](#)
- [25] Q. Jiang, J. Wang, D. Peng, C. Liu, and L. Jin. Revisiting scene text recognition: A data perspective. In *ICCV*, pages 20486–20497, 2023. [1](#), [2](#), [3](#), [5](#), [6](#), [7](#), [8](#)
- [26] D. Karatzas, L. Gomez-Bigorda, A. Nicolaou, S. Ghosh, A. Bagdanov, M. Iwamura, J. Matas, L. Neumann, V. R. Chandrasekhar, S. Lu, F. Shafait, S. Uchida, and E. Valveny. ICDAR 2015 competition on robust reading. In *ICDAR*, pages 1156–1160, 2015. [5](#), [7](#)
- [27] D. KaratzasAU, F. ShafaitAU, S. UchidaAU, M. IwamuraAU, L. G. i. BigordaAU, S. R. MestreAU, J. MasAU, D. F. MotaAU, J. A. AlmazánAU, and L. P. de las Heras. ICDAR 2013 robust reading competition. In *ICDAR*, pages 1484–1493, 2013. [5](#), [6](#)
- [28] C. Li, W. Liu, R. Guo, X. Yin, K. Jiang, Y. Du, Y. Du, L. Zhu, B. Lai, X. Hu, D. Yu, and Y. Ma. PP-OCRv3: More attempts for the improvement of ultra lightweight ocr system. *CoRR*, abs/2206.03001, 2022. [1](#), [2](#), [3](#), [6](#)
- [29] H. Li, P. Wang, C. Shen, and G. Zhang. Show, attend and read: A simple and strong baseline for irregular text recognition. In *AAAI*, pages 8610–8617, 2019. [2](#), [3](#), [7](#), [8](#), [5](#)
- [30] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. [6](#)
- [31] N. Lu, W. Yu, X. Qi, Y. Chen, P. Gong, R. Xiao, and X. Bai. MASTER: Multi-aspect non-local network for scene text recognition. *Pattern Recognit.*, 117:107980, 2021. [7](#), [8](#)

- [32] C. Luo, L. Jin, and Z. Sun. MORAN: A multi-object rectified attention network for scene text recognition. *Pattern Recognit.*, 90:109–118, 2019. [2](#), [3](#), [4](#), [7](#), [8](#), [5](#)
- [33] A. Mishra, A. Kartikey, and C. V. Jawahar. Scene text recognition using higher order language priors. In *BMVC*, pages 1–11, 2012. [5](#), [6](#)
- [34] B. Na, Y. Kim, and S. Park. Multi-modal Text Recognition Networks: Interactive enhancements between visual and semantic features. In *ECCV*, pages 446–463, 2022. [2](#), [3](#), [7](#), [5](#)
- [35] T. Q. Phan, P. Shivakumara, S. Tian, and C. L. Tan. Recognizing text with perspective distortion in natural scenes. In *CVPR*, pages 569–576, 2013. [3](#), [5](#), [6](#)
- [36] Z. Qiao, Y. Zhou, D. Yang, Y. Zhou, and W. Wang. SEED: Semantics enhanced encoder-decoder framework for scene text recognition. In *CVPR*, pages 13525–13534, 2020. [2](#), [3](#), [7](#), [8](#), [5](#)
- [37] M. Rang, Z. Bi, C. Liu, Y. Wang, and K. Han. An empirical study of scaling law for scene text recognition. In *CVPR*, pages 15619–15629, 2024. [2](#), [5](#)
- [38] F. Sheng, Z. Chen, and B. Xu. NRTTR: A no-recurrence sequence-to-sequence model for scene text recognition. In *ICDAR*, pages 781–786, 2019. [2](#), [3](#), [7](#), [8](#), [5](#)
- [39] B. Shi, X. Bai, and C. Yao. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(11):2298–2304, 2017. [1](#), [3](#), [7](#), [8](#), [5](#)
- [40] B. Shi, M. Yang, X. Wang, P. Lyu, C. Yao, and X. Bai. ASTER: An attentional scene text recognizer with flexible rectification. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(9):2035–2048, 2019. [2](#), [3](#), [4](#), [7](#), [8](#), [5](#)
- [41] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. [1](#)
- [42] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *NIPS*, pages 5998–6008, 2017. [4](#), [7](#)
- [43] I. Loshchilov and F. Hutter. SGDR: stochastic gradient descent with warm restarts. In *ICLR*, 2017. [6](#)
- [44] Z. Wan, F. Xie, Y. Liu, X. Bai, and C. Yao. 2d-ctc for scene text recognition. *CoRR*, abs/1907.09705, 2019. [2](#)
- [45] K. Wang, B. Babenko, and S. Belongie. End-to-end scene text recognition. In *ICCV*, pages 1457–1464, 2011. [5](#), [6](#)
- [46] P. Wang, C. Da, and C. Yao. Multi-Granularity Prediction for scene text recognition. In *ECCV*, pages 339–355, 2022. [2](#), [7](#), [5](#)
- [47] T. Wang, Y. Zhu, L. Jin, C. Luo, X. Chen, Y. Wu, Q. Wang, and M. Cai. Decoupled attention network for text recognition. In *AAAI*, pages 12216–12224, 2020. [3](#), [7](#), [8](#), [5](#)
- [48] Y. Wang, H. Xie, S. Fang, J. Wang, S. Zhu, and Y. Zhang. From Two to One: A new scene text recognizer with visual language modeling network. In *ICCV*, pages 14194–14203, 2021. [2](#), [3](#), [5](#), [6](#), [7](#)
- [49] J. Wei, H. Zhan, Y. Lu, X. Tu, B. Yin, C. Liu, and U. Pal. Image as a language: Revisiting scene text recognition via balanced, unified and synchronized vision-language reasoning network. In *AAAI*, pages 5885–5893, 2024. [2](#), [7](#), [5](#)
- [50] S. Woo, S. Debnath, R. Hu, X. Chen, Z. Liu, I. S. Kweon, and S. Xie. Convnext V2: co-designing and scaling convnets with masked autoencoders. In *CVPR*, pages 16133–16142, 2023. [7](#)
- [51] X. Xie, L. Fu, Z. Zhang, Z. Wang, and X. Bai. Toward Understanding WordArt: Corner-guided transformer for scene text recognition. In *ECCV*, pages 303–321, 2022. [2](#), [3](#), [7](#), [8](#), [5](#)
- [52] J. Xu, Y. Wang, H. Xie, and Y. Zhang. Ote: Exploring accurate scene text recognition using one token. In *CVPR*, pages 28327–28336, 2024. [2](#), [3](#), [7](#), [8](#), [5](#)
- [53] Jianwei Yang, Chunyuan Li, Xiyang Dai, and Jianfeng Gao. Focal modulation networks. In *NeurIPS*, 2022. [7](#)
- [54] M. Yang, B. Yang, M. Liao, Y. Zhu, and X. Bai. Class-aware mask-guided feature refinement for scene text recognition. *Pattern Recognition*, 149:110244, 2024. [2](#), [3](#), [7](#), [8](#), [5](#)
- [55] D. Yu, X. Li, C. Zhang, T. Liu, J. Han, J. Liu, and E. Ding. Towards accurate scene text recognition with semantic reasoning networks. In *CVPR*, pages 12113–12122, 2020. [2](#), [3](#), [7](#), [5](#)
- [56] H. Yu, X. Wang, B. Li, and X. Xue. Chinese text recognition with a pre-trained CLIP-Like model through image-ids aligning. In *ICCV*, pages 11909–11918, 2023. [7](#), [8](#)
- [57] X. Yue, Z. Kuang, C. Lin, H. Sun, and W. Zhang. RobustScanner: Dynamically enhancing positional clues for robust text recognition. In *ECCV*, pages 135–151, 2020. [2](#), [3](#), [7](#), [5](#)
- [58] B. Zhang, H. Xie, Y. Wang, J. Xu, and Y. Zhang. Linguistic More: Taking a further step toward efficient and accurate scene text recognition. In *IJCAI*, pages 1704–1712, 2023. [2](#), [3](#), [5](#), [7](#)
- [59] H. Zhang, Q. Yao, M. Yang, Y. Xu, and X. Bai. AutoSTR: Efficient backbone search for scene text recognition. In *ECCV*, pages 751–767. Springer, 2020. [2](#), [3](#), [7](#), [8](#), [5](#)
- [60] Z. Zhang, N. Lu, M. Liao, Y. Huang, C. Li, M. Wang, and W. Peng. Self-distillation regularized connectionist temporal classification loss for text recognition: A simple yet effective approach. In *AAAI*, pages 7441–7449, 2024. [8](#), [5](#)
- [61] Tianlun Zheng, Zhineng Chen, Jinfeng Bai, Hongtao Xie, and Yu-Gang Jiang. TPS++: Attention-enhanced thin-plate spline for scene text recognition. In *IJCAI*, pages 1777–1785, 2023. [2](#), [3](#), [8](#)
- [62] T. Zheng, Z. Chen, S. Fang, H. Xie, and Y. Jiang. CDistNet: Perceiving multi-domain character distance for robust text recognition. *Int. J. Comput. Vis.*, 132(2):300–318, 2024. [2](#), [3](#), [7](#), [8](#), [5](#)
- [63] B. Zhou, Y. Qu, Z. Wang, Z. Li, B. Zhang, and H. Xie. Focus on the whole character: Discriminative character modeling for scene text recognition. *CoRR*, 2407.05562, 2024. [2](#), [3](#)

SVTRv2: CTC Beats Encoder-Decoder Models in Scene Text Recognition

Supplementary Material

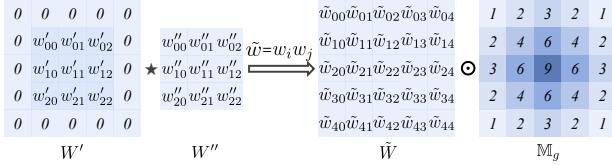


Figure 6. The illustrate of GPCM.

6. More detail of Local Mixing

Local mixing is performed using two consecutive convolutions. It is important to highlight that unlike previous CNNs [20, 22, 41], there is no normalization or activation layer between these two consecutive convolutions. Therefore, due to the associative property of the convolution operation, two consecutive convolutions (\mathbb{C}_1 and \mathbb{C}_2) with kernel size K can be combined into a single equivalent convolution (\mathbb{C}) with kernel size $2K-1$, as shown in Eq. 7:

$$\begin{aligned} O(x, y) &= [I(x, y) \star \mathbb{C}_1] \star \mathbb{C}_2 \\ &= I(x, y) \star (\mathbb{C}_1 \star \mathbb{C}_2) \\ &= I(x, y) \star \mathbb{C} \end{aligned} \quad (7)$$

We define the parameter for \mathbb{C}_1 and \mathbb{C}_2 as $W' \in \mathbb{R}^{K \times K \times D \times D}$ and $W'' \in \mathbb{R}^{K \times K \times D \times D}$, respectively. Each element $w'_{i,j} \in \mathbb{R}^{D \times D}$ represents the parameter matrix located at the i -th row and j -th column of W' . Similarly, $w''_{i,j}$ denotes the corresponding elements in W'' . Assuming that $W \in \mathbb{R}^{(2K-1) \times (2K-1) \times D \times D}$ is denoted as the parameter of the new convolution \mathbb{C} . Then W can be represented in terms of the parameters of \mathbb{C}_1 and \mathbb{C}_2 through a convolution operation. For instance, each element $w_{u,v}$ of W can be represented as the accumulation of products of the elements $w'_{i,j}$ and $w''_{m,n}$, as detailed in Eq. 8, where K is set to 3:

$$w_{u,v} = \sum_{m=0}^2 \sum_{n=0}^2 \bar{w}'_{u-m+1, v-n+1} w''_{m,n} \quad (8)$$

where $u, v \in \{0, 1, 2, 3, 4\}$,

$$\bar{w}'_{i,j} = \begin{cases} 0, & \text{if } (i \text{ or } j) \leq 0 \text{ or } (i \text{ or } j) \geq 4, \\ w'_{i-1, j-1}, & \text{otherwise.} \end{cases}$$

The term $\bar{w}'_{i,j}$ ensures that out-of-bounds indices (i.e., when i or j fall outside the valid range) are handled by assigning zero values, while valid indices correspond to the elements of $w'_{i,j}$.

We further assume that each product $w'_{i,j} w''_{m,n}$ within $w_{u,v}$ follows the same distribution. Denoting any instance

Local Mixing	Com	Union14M Benchmarks								Avg
Baseline	94.56	76.1	53.5	70.0	78.9	75.9	78.8	78.8	73.16	
Local Attn [11]	95.51	82.2	65.3	71.2	80.0	80.9	80.6	80.5	77.24	
Conv $k=3$	95.03	82.5	64.3	70.3	79.8	80.2	80.2	80.7	76.86	
Conv $k=5$	95.03	82.1	64.7	70.9	78.6	80.2	79.9	80.6	76.71	
GPCM	95.58	82.9	65.6	73.2	80.0	80.5	81.6	80.8	77.78	
+BN&ReLU	94.96	81.2	65.3	70.3	79.3	78.3	80.8	80.0	76.47	

Table 6. Comparison with other local mixing methods.

of such a product as $\tilde{w}_{u,v}$, we can express $w_{u,v}$ as a product of a constant $\alpha_{u,v}$ and $\tilde{w}_{u,v}$, where $\alpha_{u,v}$ represents the number of non-zero terms $w'_{i,j} w''_{m,n}$ contributing to $w_{u,v}$. Therefore, \mathbb{C} can be represented as $\tilde{W} \odot \mathbb{M}_g$, where $\tilde{W}(u, v) = \tilde{w}_{u,v}$, $M_g(u, v) = \alpha_{u,v}$, as shown in Fig. 6 and Eq. 9.

$$w_{0,0} = w'_{0,0} w''_{0,0} \implies 1 * \tilde{w}_{0,0}, \quad (9)$$

where $\alpha_{0,0} = 1$

$$w_{0,1} = w'_{0,1} w''_{0,0} + w'_{0,0} w''_{0,1} \implies 2 * \tilde{w}_{0,1},$$

where $\alpha_{0,1} = 2$

$$w_{0,2} = w'_{0,2} w''_{0,0} + w'_{0,1} w''_{0,1} + w'_{0,0} w''_{0,2} \implies 3 * \tilde{w}_{0,2},$$

where $\alpha_{0,2} = 3$

$$\dots$$

$$w_{2,2} = \sum_{m=0}^2 \sum_{n=0}^2 w'_{2-m, 2-n} w''_{m,n} \implies 9 * \tilde{w}_{2,2},$$

where $\alpha_{2,2} = 9$

$$\mathbf{GPCM} = \tilde{W} \odot \mathbb{M}_g \stackrel{\tilde{w}_{u,v} = w'_{i,j} w''_{m,n}}{\iff} \mathbb{C} = \mathbb{C}_1 \star \mathbb{C}_2$$

Here, \mathbb{M}_g follows a Gaussian distribution, serving as a regularization mechanism and mitigating the over-smoothing often observed in convolutional networks and effectively preserves fine-grained local features, such as character strokes and structure. We refer to this operation, involving two consecutive convolutions combined with a Gaussian prior convolution kernel, as Gaussian Prior Convolutional Mixing (GPCM).

Effectiveness of GPCM. We configured all mixing blocks to the global type as the baseline. Subsequently, we replaced the first eight global mixing blocks with GPCM and several alternative methods. As shown in Tab. 6, all local mixing methods outperform the baseline, particularly in complex scenarios, highlighting the critical role of local feature extraction in enhancing recognition. Specifically, GPCM yields an improvement of 0.72% on *Com* and 4.62% on *U14M* and also outperforms other methods. No

Models	$[D_0, D_1, D_2]$	$[N_1, N_2, N_3]$	Heads	Permutation
SVTRv2-T	[64,128,256]	[3,6,3]	[2,4,8]	$[L]_6[G]_6$
SVTRv2-S	[96,192,384]	[3,6,3]	[3,6,12]	$[L]_6[G]_6$
SVTRv2-B	[128,256,384]	[6,6,6]	[4,8,12]	$[L]_8[G]_{10}$

Table 7. Architecture specifications of SVTRv2 variants.

	Curve	MO	Artistic	Cless	Salient	MW	General
	2,426	1,369	900	779	1,585	829	400,000
Real [4]	1,276	440	432	326	431	193	254,174
REBU-Syn [37]	1,285	443	462	363	442	289	260,575
U14M-Train [25]	9	3	30	37	11	96	6,401

Table 8. Overlapping analysis between the test set of Union14M-L (*U14M*) and three real-world training sets.

ably, for *Artistic*, GPCM achieves a significant 3.2% increase, whereas the other methods only show improvements 0.3%-1.2%. This result suggests that the Gaussian kernel in GPCM effectively captures the intrinsic details of characters and robustly suppresses interference from artistic fonts. Additionally, we can consider the \mathbb{M}_g of $\text{Conv}_{k=5}$ as a constant matrix with all elements equal to one. After applying Batch Normalization (BN) and the ReLU activation function between two consecutive convolutions in GPCM, the Gaussian distribution of \mathbb{M}_g is disrupted, indicating that neither method possesses Gaussian prior kernels. In terms of average accuracy, Conv_5 and +BN&ReLU fall short of GPCM by 1.07% and 1.31%, respectively. These comparative experiments validate the efficacy of GPCM’s Gaussian prior kernel.

7. SVTRv2 Variants

There are several hyper-parameters in SVTRv2, including the depth of channel (D_i) and the number of heads at each stage, the number of mixing blocks (N_i) and their permutation. By varying them, SVTRv2 architectures with different capacities could be obtained and we construct three typical ones, i.e., SVTRv2-T (Tiny), SVTRv2-S (Small), SVTRv2-B (Base). Their detail configurations are shown in Tab. 7.

$[L]_m[G]_n$ denotes that the first m Mixing Blocks in SVTRv2 utilize Local Mixing, while the last n Mixing Blocks employ Global Mixing. Specifically, in SVTRv2-T and SVTR-S, all Blocks in the first stage and the first three Blocks in the second stage use Local Mixing. Conversely, the last three Blocks in the second stage, as well as all Blocks in the third stage, implement Global Mixing. In the case of SVTRv2-B, all Blocks in the first stage and the first two Blocks in the second stage use Local Mixing, whereas the last four Blocks in the second stage and all Blocks in the third stage adopt Global Mixing.

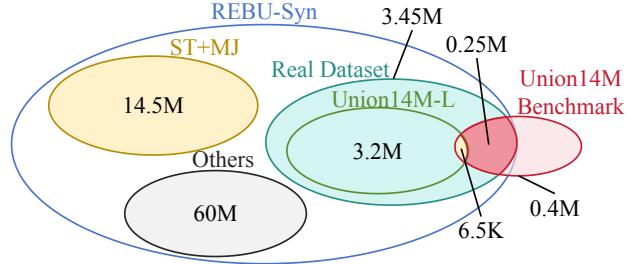


Figure 7. The relationship between the three real-world training sets.

8. More detail of real-world datasets

For English recognition, we train models on real-world datasets, from which the models exhibit stronger recognition capability [4, 25, 37]. There are three large-scale real-world training sets, i.e., the *Real* dataset [4], *REBU-Syn* [37], and the training set of *Union14M-L* (*U14M-Train*) [25]. However, as shown in Fig. 7 and Tab. 8, the former two significantly overlap with *Union14M-Benchmarks*, thus not suitable for model training. Surprisingly, *U14M-Train* is also overlapped with *Union14M-Benchmarks*, in nearly 6.5k text instances across the seven subsets. It means the models trained based on *U14M-Train* suffer from data leakage when tested on *Union14M-Benchmarks*, thus the results reported by [25] should be updated. To this end, we create a filtered version of *Union14M-L* training set, termed as *U14M-Filter*, by filtering these overlapping instances. This new dataset is used to train SVTRv2 and other 24 methods we reproduced.

9. More detail of Inference Time

In term of the inference time, we do not utilize any acceleration framework and instead employ PyTorch’s dynamic graph mode on one NVIDIA 1080Ti GPU. We first measure the inference time for 3,000 images with a batch size of 1, calculating the average inference time for each text length. We then computed the arithmetic mean of the average time across all text lengths to determine the overall inference time of the model. Algorithm 1 details the process of measuring inference time.

10. More Ablation Study

Tab. 10 presents the impact of the proposed strategies on different subsets of benchmarks, along with the corresponding changes in model parameters and processing speed. Notably, none of the strategies exhibit undesirable side effects, with the exception of the FRM strategy, which introduces a significant increase in model complexity, affecting both parameters and processing speed. However, in terms of accuracy, all four strategies yield positive results across var-

Algorithm 1: Inference Time

Input : A set of images \mathcal{I} with size $|\mathcal{I}| = 3000$,
batch size $B = 1$, N text lengths
Output: Overall inference time of the model

Initialize two lists: `total_time_list` and
`count_list` of size N , initialized to 0;
for each image I_j in \mathcal{I} where $j \in \{1, 2, \dots, 3000\}$
do

Determine the text length l_i for image I_j ;
Perform inference on I_j with text length l_i ;
Record inference time t_{ij} ;
 $\text{total_time_list}[l_i] += t_{ij}$;
 $\text{count_list}[l_i] += 1$;

Initialize `avg_time_list`;
for each text length l_i where $i \in \{1, 2, \dots, N\}$ **do**

if $\text{count_list}[i] > 0$ **then**

$\text{avg_time_list}[i] = \frac{\text{total_time_list}[i]}{\text{count_list}[i]}$;

Compute the final average inference time:

$$\text{inference_time} = \frac{1}{N} \sum_{i=1}^N \text{avg_time_list}[i]$$

return `inference_time`;

ious subsets of the Union14M benchmark. Specifically, the MSR shows significant effectiveness on two irregular benchmarks: *Curve* and *Multi-Oriented*, with accuracy improvements of 4.5% and 18.1%, respectively. This demonstrates its robustness in handling irregular patterns that challenge previous models. The FRM, despite its impact on processing speed and model parameters, proves most effective on the *Multi-Oriented* benchmark, achieving a 2.5% accuracy improvement. Meanwhile, the SGM achieves an improvement of 5.11% on the OST benchmark, underscoring its ability to address specific challenges in occluded text. These results suggest that the identified challenges have a critical influence on the performance of SVTRs. By addressing these challenges, our proposed strategies offer targeted solutions, leading to significant improvements. As a result, SVTRv2 outperforms EDTRs, advancing the state of the art in single vision model performance. The evidence presented confirms the effectiveness of our approach in overcoming limitations present in current models, marking a clear path for future enhancements in model design and optimization.

11. Results when trained on synthetic datasets

Previous research typically follows a typical evaluation protocol, where models are trained on synthetic datasets and validated using six widely recognized real-world benchmarks. Building on this protocol, we trained the SVTRv2 model on synthetic datasets. In addition to evaluating SVTRv2 on the six common benchmarks, we also assess its performance on challenging benchmarks, i.e. Union14M-Benchmark, offering a comprehensive understanding of the model's generalization capabilities. For methods that have not reported performance on challenging benchmarks, we conducted additional evaluations using their publicly available models and present these results for comparative analysis. As illustrated in Tab. 11, models trained on synthetic datasets exhibit notably reduced performance compared to those trained on large-scale real-world datasets (see Tab. 3). This performance drop is particularly pronounced on challenging benchmarks. These findings emphasize the critical importance of real datasets in improving recognition accuracy for challenging text scenarios.

Despite the challenges associated with synthetic datasets, SVTRv2 exhibits superior performance across both average accuracy metrics, surpassing the previously best-performing method by 0.22% and 2.73%, respectively. On irregular text benchmarks, such as *Curve* and *Multi-Oriented*, SVTR achieves strong results, largely due to its integrated correction module, which is particularly adept at handling irregular text patterns, even when trained on synthetic datasets. Notably, SVTRv2 achieves a substantial 4.8% improvement over SVTR on the *Curve* benchmark, further demonstrating its enhanced capacity to address irregular text. Overall, these results demonstrate that, even when trained solely on synthetic datasets, SVTRv2 exhibits strong generalization capabilities, effectively handling complex and challenging text recognition scenarios.

12. Qualitative Analysis of Recognition Results

The SVTRv2 model achieved an average accuracy of 96% on a standard dataset. To investigate the underlying causes of the remaining 4% of recognition errors, we conducted a detailed analysis of misclassified samples across six standard datasets, as illustrated in Fig. 8 and Fig. 9. While previous research has typically categorized common benchmarks into *regular* and *irregular* text. However, these error samples indicates that the majority of incorrectly recognized text is not irregular. This suggests that, under the current training paradigm using large-scale real-world datasets, a more rigorous manual screening process for common benchmarks is warranted.

Consequently, we identified five primary causes of recognition errors in these samples: (1) blurred, (2) artistic,

	Blurred	Artistic	Incomplete	Other	Total	Label_{err}
IIIT5k	0	16	1	4	21	4
SVT	4	4	4	0	12	0
ICDAR 2013	2	2	4	2	10	2
ICDAR 2015	48	19	42	13	122	35
SVTP	7	6	12	7	32	4
CUTE80	0	1	0	0	1	1
Total	61 30.81%	48 24.24%	63 31.82%	26 13.13%	198 100%	46

Table 9. Distribution of bad cases for SVTRv2 on *Common* benchmarks.

(3) incomplete text, (4) image text labeling inconsistency, and (5) others. Specifically, the blurring text category encompasses issues such as low resolution, motion blur, or extreme lighting conditions. The artistic text category includes unconventional print fonts, predominantly found in business signage, as well as a proportion of handwritten text. Incomplete text refers to characters obscured by other objects or missing due to improper cropping, where missing information must be inferred from context. Image text labeling inconsistency means that there is an error in the given text label or there are some characters with phonetic symbols. As shown in Tab. 9, after excluding samples where the errors were due to labeling inconsistency, the remaining errors could be attributed to blurred (30.81%), artistic (24.24%), and incomplete text (31.82%), respectively. This classification allows us to conclude that SVTRv2’s recognition performance, particularly in complex scenarios involving blurred, artistic, or incomplete text, requires further enhancement.

	IIT5k	SVT	ICDAR2013	ICDAR2015	SVTP	CUTE80		Curve	Multi-Oriented	Artistic	Contextless	Salient	Multi-Words	General						
	Common Benchmarks					Avg	Union14M Benchmarks					Avg	OST_w	OST_h	Avg	Size (M)	Time (ms)			
Baseline	98.0	96.8	97.7	88.7	90.1	96.2	94.56	76.1	53.5	70.0	78.9	75.9	78.8	78.8	73.16	76.2	59.4	67.78	17.96	6.7
+GPCM	98.7	97.1	97.1	89.6	91.6	97.6	95.28	82.9	65.6	73.2	80.0	80.5	81.6	80.8	77.78	78.8	63.4	71.09	17.77	6.3
+MSR	98.7	98.0	97.4	89.4	91.6	97.6	95.44	87.4	83.7	75.4	80.9	81.9	83.5	82.8	82.22	81.3	63.8	72.53	17.77	6.3
+FRM	98.8	98.1	98.4	89.8	92.9	99.0	96.16	88.2	86.2	77.5	83.2	83.9	84.6	83.5	83.86	82.9	67.0	74.92	19.76	7.0
+SGM	99.2	98.0	98.7	91.1	93.5	99.0	96.57	90.6	89.0	79.3	86.1	86.2	86.7	85.1	86.14	86.3	73.8	80.03	19.76	7.0

Table 10. Ablation study of the proposed strategies on each benchmark subset, along with variations in the model parameters and speeds.

	IIT5k	SVT	ICDAR2013	ICDAR2015	SVTP	CUTE80		Curve	Multi-Oriented	Artistic	Contextless	Salient	Multi-Words	General				
Method	Venue	Encoder	Common Benchmarks					Avg	Union14M Benchmarks					Avg	Size			
ASTER [40]	TPAMI2019	ResNet+LSTM	93.3	90.0	90.8	74.7	80.2	80.9	84.98	34.0	10.2	27.7	33.0	48.2	27.6	39.8	31.50	27.2
NRTR [38]	ICDAR2019	Stem+TF ₆	90.1	91.5	95.8	79.4	86.6	80.9	87.38	31.7	4.40	36.6	37.3	30.6	54.9	48.0	34.79	31.7
MORAN [32]	PR2019	ResNet+LSTM	91.0	83.9	91.3	68.4	73.3	75.7	80.60	8.90	0.70	29.4	20.7	17.9	23.8	35.2	19.51	17.4
SAR [29]	AAAI2019	ResNet+LSTM	91.5	84.5	91.0	69.2	76.4	83.5	82.68	44.3	7.70	42.6	44.2	44.0	51.2	50.5	40.64	57.7
DAN [47]	AAAI2020	ResNet+FPN	93.4	87.5	92.1	71.6	78.0	81.3	83.98	26.7	1.50	35.0	40.3	36.5	42.2	42.1	32.04	27.7
SRN [55]	CVPR2020	ResNet+FPN	94.8	91.5	95.5	82.7	85.1	87.8	89.57	63.4	25.3	34.1	28.7	56.5	26.7	46.3	40.14	54.7
SEED* [36]	CVPR2020	ResNet+LSTM	93.8	89.6	92.8	80.0	81.4	83.6	86.87	40.4	15.5	32.1	32.5	54.8	35.6	39.0	35.70	24.0
AutoSTR* [59]	ECCV2020	NAS+LSTM	94.7	90.9	94.2	81.8	81.7	-	-	47.7	17.9	30.8	36.2	64.2	38.7	41.3	39.54	6.00
RoScanner [57]	ECCV2020	ResNet	95.3	88.1	94.8	77.1	79.5	90.3	87.52	43.6	7.90	41.2	42.6	44.9	46.9	39.5	38.09	48.0
ABINet [15]	CVPR2021	ResNet+TF ₃	96.2	93.5	97.4	86.0	89.3	89.2	91.93	59.5	12.7	43.3	38.3	62.0	50.8	55.6	46.03	36.7
VisionLAN [48]	ICCV2021	ResNet+TF ₃	95.8	91.7	95.7	83.7	86.0	88.5	90.23	57.7	14.2	47.8	48.0	64.0	47.9	52.1	47.39	32.8
PARSeq* [4]	ECCV2022	ViT-S	97.0	93.6	97.0	86.5	88.9	92.2	92.53	63.9	16.7	52.5	54.3	68.2	55.9	56.9	52.62	23.8
MATRN [34]	ECCV2022	ResNet+TF ₃	96.6	95.0	97.9	86.6	90.6	93.5	93.37	63.1	13.4	43.8	41.9	66.4	53.2	57.0	48.40	44.2
MPGP-STR* [46]	ECCV2022	ViT-B	96.4	94.7	97.3	87.2	91.0	90.3	92.82	55.2	14.0	52.8	48.5	65.2	48.8	59.1	49.09	148
LevoOCR* [9]	ECCV2022	ResNet+TF ₃	96.6	94.4	96.7	86.5	88.8	90.6	92.27	52.8	10.7	44.8	51.9	61.3	54.0	58.1	47.66	109
CornerTF* [51]	ECCV2022	CornerEncoder	95.9	94.6	97.8	86.5	91.5	92.0	93.05	62.9	18.6	56.1	58.5	68.6	59.7	61.0	55.07	86.0
CPPD [12]	Preprint	SVTR-B	97.6	95.5	98.2	87.9	90.9	92.7	93.80	65.5	18.6	56.0	61.9	71.0	57.5	65.8	56.63	26.8
SIGA* [17]	CVPR2023	ViT-B	96.6	95.1	97.8	86.6	90.5	93.1	93.28	59.2	22.3	49.0	50.8	66.4	58.4	56.2	51.85	113
CCD* [18]	ICCV2023	ViT-B	97.2	94.4	97.0	87.6	91.8	93.3	93.55	66.6	24.2	63.9	64.8	74.8	62.4	64.0	60.10	52.0
LISTER* [8]	ICCV2023	FocalNet-B	96.9	93.8	97.9	87.5	89.6	90.6	92.72	56.5	17.2	52.8	63.5	63.2	59.6	65.4	54.05	49.9
LPV-B* [58]	IJCAI2023	SVTR-B	97.3	94.6	97.6	87.5	90.9	94.8	93.78	68.3	21.0	59.6	65.1	76.2	63.6	62.0	59.40	35.1
CDistNet* [62]	IJCV2024	ResNet+TF ₃	96.4	93.5	97.4	86.0	88.7	93.4	92.57	69.3	24.4	49.8	55.6	72.8	64.3	58.5	56.38	65.5
CAM* [54]	PR2024	ConvNeXtV2-B	97.4	96.1	97.2	87.8	90.6	92.4	93.58	63.1	19.4	55.4	58.5	72.7	51.4	57.4	53.99	135
BUSNet [49]	AAAI2024	ViT-S	96.2	95.5	98.3	87.2	91.8	91.3	93.38	-	-	-	-	-	-	-	56.8	
DCTC [60]	AAAI2024	SVTR-L	96.9	93.7	97.4	87.3	88.5	92.3	92.68	-	-	-	-	-	-	-	40.8	
OTE [52]	CVPR2024	SVTR-B	96.4	95.5	97.4	87.2	89.6	92.4	93.08	-	-	-	-	-	-	-	25.2	
CRNN [39]	TPAMI2016	ResNet+LSTM	82.9	81.6	91.1	69.4	70.0	65.5	76.75	7.50	0.90	20.7	25.6	13.9	25.6	32.0	18.03	8.30
SVTR* [11]	IJCAI2022	SVTR-B	96.0	91.5	97.1	85.2	89.9	91.7	91.90	69.8	37.7	47.9	61.4	66.8	44.8	61.0	55.63	24.6
SVTrv2	-	SVTrv2-B	97.7	94.0	97.3	88.1	91.2	95.8	94.02	74.6	25.2	57.6	69.7	77.9	68.0	66.9	62.83	19.8

Table 11. Results of SVTrv2 and existing models when trained on synthetic datasets (*ST + MJ*) [19, 24]. * represents that the results on Union14M-Benchmarks are evaluated using the model they released.



Figure 8. The bad cases of SVTRv2 in IIIT5k [33], SVT [45], ICDAR 2013 [27], SVTP [35] and CUTE80 [1]. labels and predicted results, and predicted scores are denoted as Text_{label} | Text_{pred} | Score_{pred} . Yellow, red, blue, and green boxes indicate blurred, artistic fonts, incomplete text, and label-inconsistent samples, respectively. Other samples have no boxes.

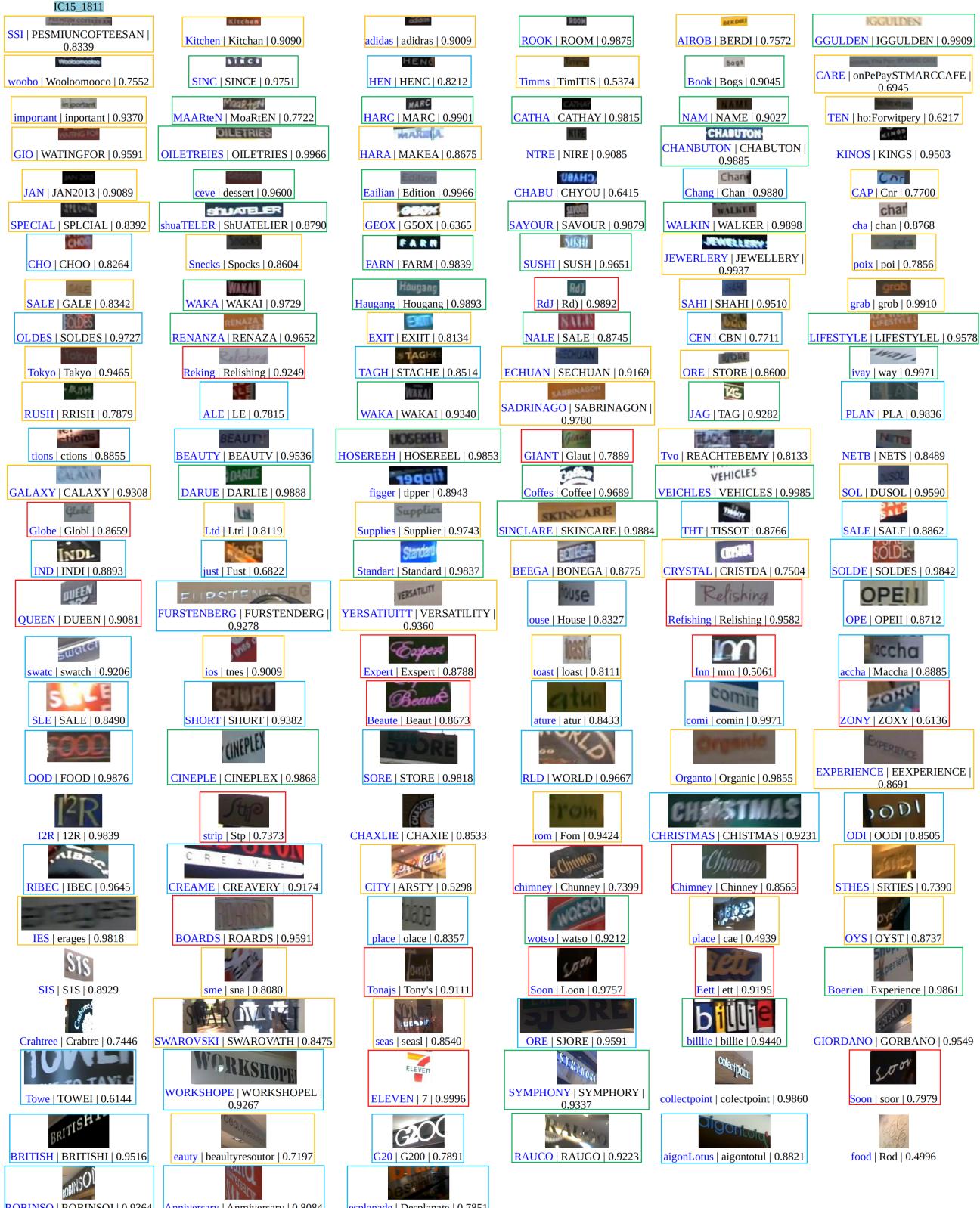


Figure 9. The bad cases of SVTRv2 in ICDAR 2015 [26]. labels and predicted results, and predicted scores are denoted as $\text{Text}_{label} | \text{Text}_{pred} | \text{Score}_{pred}$. Yellow, red, blue, and green boxes indicate blurred, artistic fonts, incomplete text, and label-inconsistent samples, respectively. Other samples have no boxes.