Focus on the Whole Character: Discriminative Character Modeling for Scene Text Recognition

Bangbang Zhou, **Yadong Qu**, **Zixiao Wang**, **Zicheng Li**, **Boqiang Zhang**, **Hongtao Xie***
University of Science and Technology of China, Hefei, China
{bangzhou01, qqqyd, wzx99, lizicheng, cyril}@mail.ustc.edu.cn, htxie@ustc.edu.cn

Abstract

Recently, scene text recognition (STR) models have shown significant performance improvements. However, existing models still encounter difficulties in recognizing challenging texts that involve factors such as severely distorted and perspective characters. These challenging texts mainly cause two problems: (1) Large Intra-Class Variance. (2) Small Inter-Class Variance. An extremely distorted character may prominently differ visually from other characters within the same category, while the variance between characters from different classes is relatively small. To address the above issues, we propose a novel method that enriches the character features to enhance the discriminability of characters. Firstly, we propose the Character-Aware Constraint Encoder (CACE) with multiple blocks stacked. CACE introduces a decay matrix in each block to explicitly guide the attention region for each token. By continuously employing the decay matrix, CACE enables tokens to perceive morphological information at the character level. Secondly, an Intra-Inter Consistency Loss (I^2CL) is introduced to consider intra-class compactness and inter-class separability at feature space. I^2CL improves the discriminative capability of features by learning a long-term memory unit for each character category. Trained with synthetic data, our model achieves state-of-the-art performance on common benchmarks (94.1% accuracy) and Union14M-Benchmark (61.6% accuracy). Code is available at https://github.com/bang123-box/CFE.

1 Introduction

Scene Text Recognition (STR) aims to recognize character sequences from cropped text images [Bautista and Atienza, 2022; Zhang *et al.*, 2023; Cheng *et al.*, 2023; Fan *et al.*, 2023]. Existing STR methods adeptly read texts encompassing billboards, road signs, checks, *etc*. However, with societal advancements, the demands on STR models are no longer limited to performing well in simple texts, but also need to improve their performance in challenging texts.

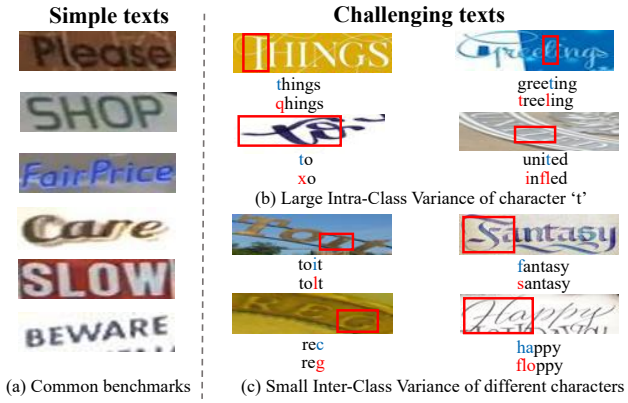


Figure 1: Differences between simple and challenging texts. (a) Simple texts are singular in style and uniform in size. (b) With variations in appearances and size, the character 't' is misrecognized. (c) The similarity in appearances of different category characters leads to wrong recognition. The first line is the label and the second line is the prediction with our baseline model. The incorrectly recognized characters are highlighted in red.

The recent methods have obtained superior performance on six common benchmarks [Risnumawan *et al.*, 2014; Mishra *et al.*, 2012; Wang *et al.*, 2011; Phan *et al.*, 2013; Karatzas *et al.*, 2013; Karatzas *et al.*, 2015] which are easy to recognize as shown in Figure 1(a). However, with the introduction of a challenging Union14M-Benchmark [Jiang *et al.*, 2023], existing STR models perform poorly on Curve, Artistic, and Contextless datasets. For example, MGP-Base [Wang *et al.*, 2022] only achieves the accuracy rates of 55.2%, 52.8%, and 48.4% on them, respectively. We believe that the poor performance on challenging texts is primarily due to two ignored issues.

The first issue is summarized as Large Intra-class Variance (LISV). Figure 1(b) illustrates the different instances of the character 't'. There are variations in the visual appearances, shape, and size, which lead to errors in recognition results. We attribute the misrecognition of characters to the large variance of the same category characters. Due to the existence of LICV, the discriminative features of characters

*Corresponding Author

will be weakened, eventually leading to the misrecognition of characters. To solve LICV, it's essential to enhance the discriminability of features by encoding local patterns or structures within the character. This helps model the correlation between the components of a character, capturing discriminative information more comprehensively. However, existing STR encoders have difficulty learning the intra-character local patterns. CNN-based encoders [Shi *et al.*, 2016; Baek *et al.*, 2019] can learn local features, but the receptive field is too small to perceive the information of the whole character. Transformer-based encoders [Sheng *et al.*, 2019; Bautista and Atienza, 2022] hardly focus on the local information at character level due to the global modeling during self-attention. Thus, improving the ability to encode local patterns (*e.g.*, stroke, morphology, *etc.*) at character level is a crucial step in solving LICV.

The second problem is Small Inter-Class Variance (SICV). As shown in Figure 1(c), our baseline recognizes some characters as other different category characters due to their similar visual appearances. For example, the word 'happy' is misrecognized as 'floppy', which also conforms to linguistic rules. Therefore, using additional language models or linguistic information does not help solve the SICV problem. Furthermore, we argue that SICV can lead to a mixed distribution of different category character features in the decoding space (demonstrated formally in Sec. 4.6). To solve SICV and LICV together, CornerTransformer [Xie *et al.*, 2022] designs a Character Contrastive loss (CC loss) to bring the same category characters closer and separate characters from different classes. However, since CC loss only considers the character feature distribution within a batch, it greatly limits the diversity and richness of the global character feature distribution. Hence, how to utilize global character feature distribution is a great challenge for solving LICV and SICV.

To address the above issues, we propose a Character Features Enriched model (CFE) to obtain the discriminative character features from two aspects. Firstly, the Character-Aware Constraint Encoder (CACE) is proposed to perceive the local patterns such as the morphological information. For CACE, the discriminative features are extracted by multiple stacked blocks. In each block, we design a decay matrix $\mathbf{D}$ to attenuate the self-attention mechanism according to the spatial distances between visual tokens. The greater the distances between tokens, the less attention is paid to them. Compared to the vanilla self-attention, the block can alleviate the noise interference from other characters and pay more attention to the region near each token. In this way, CACE encodes the local patterns and relationships between the components of each character. Additionally, to fully utilize the visual features output from the multiple blocks, a fusion strategy is employed to merge them. Secondly, an Intra-Inter Consistency Loss (I^2CL) is introduced to solve LICV and SICV. Based on the contrastive learning [Qi and Su, 2017; Xie *et al.*, 2022; Zhang *et al.*, 2022], I^2CL further predefines a long-term memory unit for each character category. For each character in a batch, its positive example refers to the unit with the same character category, while the other units serve as negative samples. In the training, I^2CL updates these memory units based on each character. Different from [Xie *et al.*, 2022; Zhang *et al.*, 2022] that only consider the local distribution, I^2CL can efficiently represent all characters by learning a discrete distribution for long-term memory units. Finally, all characters will be tightly distributed around the memory units according to their categories. This ensures the intra-class compactness and inter-class separability, and improves the discriminability of characters. Compared with previous methods, we have fewer training parameters, while achieving better performance.

The main contributions of our work are as follows:

- We point out that the LICV and SICV issues in challenging texts lead to poor performance of the STR models, and propose a novel approach to effectively handle the two issues.
- We design a Character-Aware Constraint Encoder to focus on the local patterns of character level, which utilizes the morphological information to enrich features.
- We introduce an Intra-Inter Consistency Loss to reduce intra-class variance and increase inter-class variance by learning a set of long-term memory units.
- Experiments on common benchmarks and Union14-Benchmark demonstrate that our CFE surpasses state-of-the-art performance, with accuracy rates of 94.1% and 61.6%, respectively.

2 Related Work

2.1 Scene Text Recognition

In scene text recognition models, the visual encoder is an essential component. It aims to provide discriminative visual feature representation for subsequent CTC [Graves *et al.*, 2006] decoders, attention decoders, or Transformer decoders. Early methods employing CNN as encoder have been widely applied in various networks and applications [Shi *et al.*, 2016; Shi *et al.*, 2018; Baek *et al.*, 2019]. However, CNN-based approaches typically compress the height dimension of images into 1 during feature extraction, causing each visual feature to correspond to a thin-slice region in the image. This limitation leads to poor performance on irregular datasets. Recently, due to the significant advancements of Transformer in the visual domain, many recent works [Atienza, 2021; Du *et al.*, 2022; Bautista and Atienza, 2022; Wang *et al.*, 2023] have opted to use Vision Transformer as the visual encoder. These methods demonstrate good performance on irregular datasets. ViTSTR [Atienza, 2021] utilizes the Vision Transformer as the encoder to model relationships between different visual tokens. SVTR [Du *et al.*, 2022] employs a pyramid-style Transformer as the visual encoder to guide the model in establishing global relationships between characters and local relationships within character. Although SVTR and ViTSTR are both pure visual models, there exist some performance differences. We attribute this to SVTR enhancing its ability to model the relationships between the character components. For the problem of LISV, we also need to focus on the local features to obtain discriminative features for recognizing characters. Therefore, we propose the Charatcer-Aware Constraint Encoder. It can perceive the local patterns within character by utilizing the decay matrix in each block.

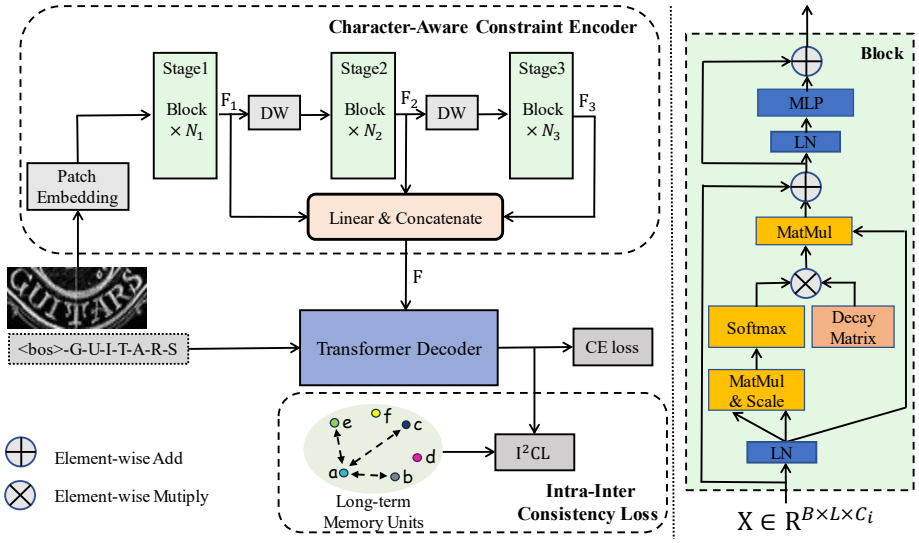


Figure 2: The framework of our CFE. The pipeline is composed of two key components: CACE and I^{2}CL. CACE explores the local patterns within character by utilizing the decay matrix. I^{2}CL uses a set of learnable long-term memory units to represent the global character feature distribution in the decoding space. CE loss denotes the cross entropy loss. DW means the 2x downsampling at height dimension using CNN.

2.2 Contrastive Learning in STR

In scene text recognition, the optimized object is not only cross entropy or CTC loss [Graves *et al.*, 2006], but also combines contrastive learning to handle different problems. ConCLR [Zhang *et al.*, 2022] uses contrastive loss to bring identical characters closer and separate different characters in the embedding space. DiG [Yang *et al.*, 2022] incorporates a contrastive learning branch to mimic human-like reading behavior and learn text discrimination. In the meantime, it employs a momentum branch to create a comprehensive and reliable dictionary on-the-fly. CornerTransformer [Xie *et al.*, 2022] designs a Character Contrastive loss that implicitly learns common features for each character class for solving artistic text recognition. CLIP-OCR [Wang *et al.*, 2023] introduces a linguistic consistency loss for aligning the intra-relationship and inter-relationship. We argue that contrastive learning is also helpful in solving the problems of LICV and SICV. Hence, we introduce the Intra-Inter Consistency Loss to take intra-class compactness and inter-class separability into account. This loss function can learn a long-term memory unit for each class, simultaneously ensuring that the memory units of different categories are far apart.

3 Proposed Method

In this section, we first detail the pipeline of the proposed method CFE in Sec. 3.1, and then introduce Character-Aware Constraint Encoder (CACE) and Intra-Inter Consistency Loss (I^{2}CL) in Sec. 3.2 and Sec. 3.3 respectively.

3.1 Pipeline

The pipeline of CFE is illustrated in Figure 2 and it can be viewed as an encoder-decoder architecture. Given a batch of images with size $B \times H \times W \times 3$, after three stages in CACE, we acquire the three visual sequences: $\mathbf{F_1} \in \mathbf{R}^{B\times \frac{HW}{16}\times C_1}$, $\mathbf{F_2} \in \mathbf{R}^{B\times \frac{HW}{32}\times C_2}$, $\mathbf{F_3} \in \mathbf{R}^{B\times \frac{HW}{64}\times C_3}$. Subsequently, we linearly map and concatenate them to get the final output visual sequences $\mathbf{F} \in \mathbf{R}^{B\times \frac{7}{64}HW\times C}$. Next, $\mathbf{F}$ is fed into the Transformer Decoder to generate recognition features $\mathbf{O} \in \mathbf{R}^{B\times T\times C}$. Finally, we calculate the CE loss and I^2**CL** separately and sum them up to obtain the training objective.

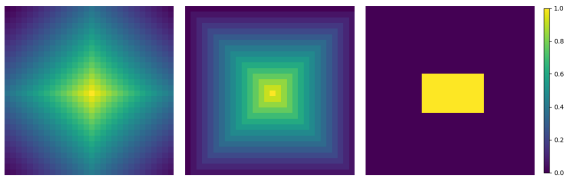
3.2 Character-Aware Constraint Encoder

To address the issue of LICV, we present a novel Character-Aware Constraint Encoder (CACE). In this encoder, the visual features are extracted from three stages, and each stage consists of multiple stacked blocks. CACE introduces an explicit decay matrix $\mathbf{D}$ into the block to encode the local patterns (*e.g.*, stroke, morphology, *etc.*) and relationships between the inter-character components. Additionally, to fully utilize the visual features output from the three stages, we employ a simple multi-scale fusion strategy to merge them.

The block is depicted in Figure 2, we summarize how the block works in Eq 1:

$$\begin{aligned}
\mathbf{Q} &= (\mathrm{LN}(\mathbf{X}))\mathbf{W_Q} \odot \Theta, \\
\mathbf{K} &= (\mathrm{LN}(\mathbf{X}))\mathbf{W_K} \odot \hat{\Theta}, \\
\mathbf{V} &= \mathrm{LN}(\mathbf{X})\mathbf{W_V}, \\
\mathbf{X} &= \mathbf{X} + (Softmax(QK^T/d) \odot \mathbf{D})\mathbf{V}, \\
\mathbf{X} &= \mathbf{X} + MLP(\mathrm{LN}(X)),
\end{aligned} \tag{1}$$

where $\mathbf{X} \in \mathbf{R}^{B\times L\times C_i}$ is the input of the block and LN stands for Layer Normalization. $\mathbf{W_Q}$, $\mathbf{W_k}$, $\mathbf{W_V}$ are the learnable projection matrices. Θ represents the position embedding and $\hat{\Theta}$ is its complex conjugate followed [Sun *et al.*, 2023]. $\mathbf{D} \in \mathbf{R}^{B\times L\times L}$ represents the decay matrix which values in [0,



(a) option 1 (b) option 2 (c) option 3

Figure 3: Visualization of three different options for generating decay matrix.

1]. We design three different ways to yield $\mathbf{D}$:

$$D_{ij} = \begin{cases} \gamma^{(|x_i - x_j| + |y_i - y_j|)}, & \text{option 1} \\ \gamma^{max(|x_i - x_j|, |y_i - y_j|)}, & \text{option 2} \\ [|x_i - x_j| \leq w \ \& \ |y_i - y_j| \leq h], & \text{option 3} \end{cases} \quad (2)$$

where x_i and y_i denote the X and Y coordinates, respectively, of the i-th token in 2D space. γ values range between 0 and 1 [Sun *et al.*, 2023]. $[\cdot]$ is an indicator function, if the condition is satisfied returning 1 else 0. w, h are used to control the region during self-attention which are set 5 and 3 (insensitive) empirically. Further, we visualize $\mathbf{D}$ in the form of heatmap. As shown in Figure 3, the first two options correspond to dynamic decay, which is inversely proportional to the distances. Option 3 employs fixed decay which utilizes a predefined binary window to control the attention region for each token like in SVTR [Du *et al.*, 2022]. In this paper, we adopt option 2 as our choice. We argue that dynamic decay is compatible with human perception of distances. Ablation study in Sec. 4.5 demonstrates its effectiveness.

In Eq. 2, $\mathbf{D}$ introduces spatial distances prior knowledge into the self-attention mechanism. The longer distances between token i and j, $\mathbf{D}_{i,j}$ become smaller. Its purpose is to avoid the influence of irrelevant noisy tokens over long distances, allowing each token to model the different components within the whole character. Finally, the decay matrix enables the visual encoder to learn more distinctive visual representations for characters. Consequently, compared to global modeling without $\mathbf{D}$, it exhibits greater proficiency in capturing local patterns in each character. In practice, we employ $\mathbf{D}$ into the first x blocks for intra-character components modeling, while the remaining blocks do not use $\mathbf{D}$ for inter-character modeling.

To better capture the different patterns of a certain character, we design a fusion strategy to obtain multi-scale character features. Specifically, it fuses the features $\mathbf{F_1}$, $\mathbf{F_2}$, $\mathbf{F_3}$ by projecting them into the same hidden dimension. Overall, on the one hand, by adding the decay matrix, CACE enables each token to perceive local patterns within character, allowing for better extraction of discriminative features. On the other hand, through multi-scale fusion, CACE can integrate the local patterns of a character at different scales, which can enhance the diversity of character features.

3.3 Intra-Inter Consistency Loss

Although CACE can help alleviate the problem of LICV, the existence of SICV still prevents the model from reaching its optimal performance. The SICV issue leads to the phenomenon of mixture distribution between character features. Therefore, it is necessary to increase the distance between characters of different categories and decrease the distance between characters of the same category. We believe that using the trait of contrastive learning to solve the SICV problem is a good choice. However, existing contrastive learning methods mostly cluster characters within a batch, lacking the perception of global character feature distribution. Hence, we introduce the Intra-Inter Consistency Loss (I^2CL) to explore the distribution of each character category by learning a set of long-term memory units.

Formally, the I^2CL is defined as illustrated in Eq 3:

$$L_{cl} = \frac{1}{2} \sum_{i=1}^{BT} \frac{||O_i - c_{y_i}||_2^2}{(\Sigma_{j=1, j \neq y_i}^{V} ||O_i - c_j||_2^2) + \delta}, \quad (3)$$

where L_{cl} represents the I^2CL. Initially, we reshape the output feature O into $\mathbf{R}^{BT \times C}$. BT denotes the number of characters in a batch and T is the maximum length of character sequences in a text. $\text{O}_i \in \mathbf{R}^C$ signifies the i-th character feature, y_i denotes the label of O_i. c_{y_i} represents the memory unit of the y_i-th character category in decoding space which can be updated during training. V denotes the vocabulary size. δ is a constant used to prevent the denominator from equaling 0 and we set $\delta = 1$ by default.

Compared with DiG [Yang *et al.*, 2022] and CornerTransformer [Xie *et al.*, 2022], although these approaches all use contrastive learning loss, there remain some differences. DiG and CornerTransformer only consider the intra-class compactness and inter-class separability of features in a batch. But I^2CL considers the clustering between all the training samples. After training, the memory units of our I^2CL will be a discrete distribution because there is a penalty for too small distance between different memory units. In addition, we can use these memory units to represent the global character feature distribution. From one perspective, this discrete distribution will increase the distance between characters of different categories and solve the SICV problem. From another perspective, all characters will be tightly distributed around the memory units corresponding to their categories, alleviating the LICV problem.

3.4 Training Objective

The final objective function of the proposed method is formulated in Eq. 4. L_{ce} represents the cross entropy loss. λ denotes the scalar used to balance the two loss functions and is set to 0.2.

$$L = L_{ce} + \lambda L_{cl}. \quad (4)$$

4 Experiment

4.1 Datasets

Following the setup of [Wang *et al.*, 2022; Fang *et al.*, 2021], we conduct experiments using MJSynth [Jaderberg *et al.*, 2014; Jaderberg *et al.*, 2016] and SynthText [Gupta *et al.*, 2016] as training data. The training data consists of 16M synthetic text images. We evaluate the model on 6 common benchmarks containing IIIT [Mishra *et al.*, 2012],

Models	$[C_1, C_2, C_3]$	$[N_1, N_2, N_3]$	Heads	C	Decay Order	Params(M)
CFE-Tiny	[64, 128, 256]	[3,6,3]	[2,4,8]	128	6-6	4.5
CFE-Small	[96, 192, 256]	[3,6,6]	[3,6,8]	192	8-7	9.2
CFE-Base	[128, 256, 384]	[3,6,9]	[4,8,12]	256	8-10	23.9

Table 1: Various configurations of our CFE. Heads denote the number of attention heads used in each stage. For Decay Order x-y, x signifies using $\mathbf{D}$ in the first x blocks, while y denotes not using $\mathbf{D}$ in the last y blocks. Params(M) indicates the trainable parameters of the model.

Method	Common Benchmarks							Union14M-Benchmark								P
	IC13	SVT	IIIT	IC15	SVTP	CT	WAVG	Cur	M-O	Art	Con	Sal	M-W	Gen	AVG	(M)
RobustScanner* [Yue *et al.*, 2020]	94.8	88.1	95.3	77.1	79.5	90.3	88.4	43.6	7.9	41.2	42.6	44.9	46.9	39.5	38.1	-
PARSeq$^{\#}_{A}$ [Bautista and Atienza, 2022]	97.0	93.6	97.0	86.5	88.9	92.3	93.3	58.2	17.2	54.2	59.4	67.7	55.8	61.1	53.4	23.8
SVTR* [Du *et al.*, 2022]	97.1	91.5	96.0	85.2	89.9	91.7	92.3	63.0	**32.1**	37.9	44.2	67.5	49.1	52.8	49.5	24.6
CLIP-OCR† [Wang *et al.*, 2023]	97.7	94.7	97.3	87.2	89.9	93.1	93.8	59.4	15.9	57.6	59.2	69.2	62.6	62.3	55.2	31.1
VisionLAN* [Wang *et al.*, 2021]	95.7	91.7	95.8	83.7	86.0	88.5	91.2	57.7	14.2	47.8	48.0	64.0	47.9	52.1	47.4	32.8
ABINet* [Fang *et al.*, 2021]	97.4	93.5	96.2	86.0	89.3	89.2	92.3	59.5	12.7	43.3	38.3	62.0	50.8	55.6	46.0	36.7
MATRN* [Na *et al.*, 2022]	**97.9**	**95.0**	96.6	86.6	90.6	93.5	93.5	63.1	13.4	43.8	41.9	66.4	53.2	57.0	48.4	44.2
SRN* [Yu *et al.*, 2020]	95.5	91.5	94.8	82.7	85.1	87.8	90.4	63.4	25.3	34.1	28.7	56.5	26.7	46.3	39.6	54.7
MGP-Base† [Wang *et al.*, 2022]	97.3	94.7	96.4	87.2	91.0	90.3	93.3	55.2	14.0	52.8	48.4	65.1	48.1	59.0	48.9	148.0
LPV-Tiny† [Zhang *et al.*, 2023]	96.7	92.9	96.3	86.4	86.7	90.6	92.5	55.0	11.7	51.4	53.9	65.6	52.1	58.1	49.7	8.1
LPV-Small† [Zhang *et al.*, 2023]	96.8	93.7	96.7	87.1	89.8	92.4	93.3	61.5	15.7	55.9	58.8	69.0	62.1	60.3	54.8	14.0
LPV-Base† [Zhang *et al.*, 2023]	97.6	94.6	97.3	**87.5**	90.9	94.8	94.0	68.3	21.0	59.6	65.1	**76.2**	63.6	62.0	59.4	35.1
LISTER-Tiny† [Cheng *et al.*, 2023]	97.7	93.5	96.5	86.5	87.8	87.9	92.8	49.3	13.2	49.1	58.2	58.2	64.2	60.8	50.4	19.9
LISTER-Base† [Cheng *et al.*, 2023]	**97.9**	93.8	96.9	**87.5**	89.6	90.6	93.5	54.8	17.2	51.3	61.5	62.6	61.3	62.9	53.1	49.9
CFE-Tiny(Ours)	96.7	93.5	96.9	85.7	86.8	91.7	92.7	56.6	14.0	53.6	64.7	67.0	63.6	61.4	54.4	4.5
CFE-Small(Ours)	96.7	93.7	97.2	86.8	89.9	93.1	93.4	63.4	17.3	57.5	71.5	73.2	64.2	63.8	58.7	9.2
CFE-Base(Ours)	97.6	94.3	**97.9**	86.9	**91.8**	**95.5**	**94.1**	**70.0**	20.8	**62.4**	**72.0**	75.2	**65.7**	**65.1**	**61.6**	23.9

Table 2: Performance of models trained on synthetic datasets. * means the results on Union14M-Benchmark is from MAERec [Jiang *et al.*, 2023]. † signifies we use the released checkpoints to test Union14M-Benchmark. # implies we retrain the model on the synthetic datasets and then test the result on Union14M-Benchmark. Cur, M-O, Art, Ctl, Sal, M-W, and Gen respectively represent Curve, Multi-Oriented, Artistic, Contextless, Salient, Multi-Words, and General. For simplicity, they have the same meaning in the following. P(M) indicates the trainable parameters. Bold values denote the first accuracy in each column.

IC13 [Karatzas *et al.*, 2013], IC15 [Karatzas *et al.*, 2015], SVT [Wang *et al.*, 2011], SVTP [Phan *et al.*, 2013] and CT [Risnumawan *et al.*, 2014]. To further valid the effectiveness of our CFE, we extra test performance on Union14M-Benchmark [Jiang *et al.*, 2023], ArT [Chng *et al.*, 2019], COCO-Text [Veit *et al.*, 2016], Uber-Text [Zhang *et al.*, 2017], and WordArt [Xie *et al.*, 2022]. Beyond that, we also supply a few experiments trained on Union14M-L.

4.2 Implementation Details

To balance the accuracy and speed, we develop three variations with varying numbers of parameters similar to SVTR and LPV [Zhang *et al.*, 2023]. The specific network configurations are detailed in Table 1. The images are resized to 32×128. For data augmentation, we follow the random data augmentation methods [Bautista and Atienza, 2022] including Invert, GaussianBlur, Sharpness, and PoissonNoise. The vocabulary size V is 96, which comprises mixed-case alphanumeric characters, punctuation marks, [BOS] for the beginning symbol, and [EOS] for the ending symbol. The maximum label length T is set to 25. We train 20 epochs with a warm-up of 1.5 epochs, utilizing the Adam optimizer [Kingma and Ba, 2014] with a learning rate of 5e-4. We add L_{cl} in the last 25% time of the training process. By this point, the model has already demonstrated convincing recognition capabilities, allowing more accurate learning for memory units and clustering. We use the Transformer Decoder with 1 layer as the recognition decoder and adopt autoregressive decoding for training. The experiments are conducted on 4 NVIDIA 4090 GPUs with a batch size of 384.

4.3 Evaluation Metric

For validation, we configure the vocabulary size to 36, encompassing 0-9 and a-z. We employ the word accuracy as the evaluation metric. Consistent with [Baek *et al.*, 2021], we record the weighted average score (WAVG) on common benchmarks based on the number of samples. As for Union14M-Benchmark, we report the average score (AVG) like [Jiang *et al.*, 2023].

4.4 Comparisons with State-of-the-Arts

In Table 2, we compare our CFE with multiple recent state-of-the-art methods on common benchmarks. All the methods are trained by synthetic image texts for fair comparison. CFE-Base shows significant performance, especially in datasets of regular IIIT and irregular SVTP and CT. It achieves SOTA accuracy of 94.1% while keeping low parameters (23.9M) compared to other STR models. Further, CFE-Tiny outperforms LPV-Tiny with only 4.5M parameters, and

Method	ArT	COCO	Uber	WordArt
ABINet† [Fang *et al.*, 2021]	65.4	57.1	34.9	67.4
PARSeq$_A$ [Bautista and Atienza, 2022]	70.7	64.4	42.0	-
ConnerTransfomer [Xie *et al.*, 2022]	-	-	-	70.8
MGP-Base† [Wang *et al.*, 2022]	69.0	65.4	40.7	72.4
LISTER-Tiny† [Cheng *et al.*, 2023]	69.0	64.1	48.0	67.6
LISTER-Base† [Cheng *et al.*, 2023]	70.1	65.8	**49.0**	69.8
CLIP-OCR [Wang *et al.*, 2023]	70.5	**66.5**	42.4	73.9
CFE-Tiny(Ours)	69.8	62.8	42.3	71.0
CFE-Small(Ours)	71.8	64.2	42.8	74.0
CFE-Base (Ours)	**72.8**	66.3	43.6	**75.7**

Table 3: Comparison with SOTA methods on challenging datasets. † means to test performance using open-released model weights.

Method	Cur Sal	M-O M-W	Art Gen	Con AVG
MGP-Base# [Wang *et al.*, 2022]	78.8 75.7	74.3 60.0	67.7 80.1	68.7 72.2
LPV-Base# [Zhang *et al.*, 2023]	85.2 83.3	75.9 82.2	74.8 82.8	80.5 80.7
CLIP-OCR# [Wang *et al.*, 2023]	84.6 81.3	**83.1** 81.8	76.3 83.9	80.0 81.6
LISTER-Base# [Cheng *et al.*, 2023]	70.9 66.9	51.1 77.6	65.4 77.9	73.3 69.0
MAERec-Base [Jiang *et al.*, 2023]	76.5 74.6	67.5 77.7	65.7 81.8	75.5 74.2
CFE-Tiny	77.3 74.2	62.1 77.9	69.6 79.9	79.1 74.3
CFE-Small	84.4 81.3	73.4 83.3	75.4 82.8	84.7 80.8
CFE-Base	**86.8** **83.5**	80.4 **85.9**	**77.5** **84.4**	**85.5** **83.4**

Table 4: Performance on models trained on Union14M-L. # implies we retrain the model on Union14M-L and then test on Union14M-Benchmark.

CACE	I^{2}CL	Cur	M-O	Art	Ctl	Sal	M-W	Gen	AVG
-	-	68.5	20.2	58.6	66.9	74.4	65.4	64.7	59.8
✓	-	68.3	21.4	59.8	70.1	74.8	66.1	64.9	60.8
-	✓	67.6	20.7	59.4	71.4	76.0	64.2	65.2	60.6
✓	✓	70.0	20.8	62.4	72.0	75.2	65.7	65.1	**61.6**

Table 5: The effectiveness of CACE and I^{2}CL.

CFE-Small acquires a good performance of 93.4% with 9.2M parameters. Besides, CFE-Base achieves SOTA performance on Union14M-Benchmark, with an average score of 61.6%. For challenging datasets in Table 3, although CFE-Base attains first or second accuracy, there remains a large gap with LISTER-Base on the Uber-Text dataset which contains many vertical texts. This is because LISTER-Base performs a rotation operation based on the original aspect ratio of the images, enhancing its ability to recognize vertical texts.

The performance trained on Union14M-L is shown in Table 4. Compared to the recent methods with additional linguistic information, our CFE-Base can continue to keep the best accuracy on most datasets of Union14M-Benchmark. These results all imply that our CFE is effective in enhancing the discriminative character features for solving challenging scene text recognition and can maintain superior performance in recognizing simple texts.

Method	Cur	M-O	Art	Ctl	Sal	M-W	Gen	AVG
-	68.3	21.4	59.8	70.1	74.8	66.1	64.9	60.8
CC loss	66.4	19.1	58.0	69.2	73.6	68.0	64.8	59.9
I^{2}CL	70.0	20.8	62.4	72.0	75.2	65.7	65.1	**61.6**

Table 6: Comparision with other contrastive loss.

D	M-S	Cur	M-O	Art	Ctl	Sal	M-W	Gen	AVG
-	-	67.6	20.7	59.4	71.4	76.0	64.2	65.2	60.6
✓	-	68.0	20.7	58.5	73.3	75.5	66.6	65.2	61.1
-	✓	66.3	18.9	58.8	70.7	73.2	68.7	64.7	60.2
✓	✓	70.0	20.8	62.4	72.0	75.2	65.7	65.1	**61.6**

Table 7: The effectiveness of decay matrix $\mathbf{D}$ and multi-scale fusion in CACE. M-S represents the multi-scale fusion.

4.5 Ablation Study

The Effectiveness of CACE and I^{2}CL

To investigate the effectiveness of CACE and I^{2}CL, we conduct experiments in Table 5. The baseline refers to not using CACE and I^{2}CL which achieves an accuracy of 59.8%. After adding CACE to guide the encoder to perceive the morphological information and character region, we can get an average accuracy of 60.8% (+1.0%). This improvement implies that the encoder can model the local patterns of each character by using $\mathbf{D}$ and multi-scale fusion. On the other hand, only adopting I^{2}CL obtains the accuracy of 60.6% (+0.8%) which means learning long-term memory units is effective for solving LISV and SICV. When employing them together, we further improve the performance to 61.6% (+1.8%). These results demonstrate that our CFE can utilize the enriched features to enhance the discriminability of characters and that CACE and I^{2}CL can cooperate well to optimize our model.

Moreover, Table 6 compares our I^{2}CL with the Character Contrastive loss (CC loss) designed by CornerTransformer. The result indicates that our I^{2}CL surpasses the conventional contrastive learning method CC loss. Compared to the performance without contrastive learning loss, the introduction of CC loss leads to a decrease of 0.9% (59.9%). We argue that achieving small intra-class differences and large inter-class differences is impractical by clustering characters only within a batch. Hence, CC loss fails to learn the global feature distribution of each category. In contrast, our I^{2}CL employs the trainable long-term memory units to represent the global distribution of each category, aiding the model in better achieving intra-class compactness and inter-class separability. This verifies that our I^{2}CL can enhance the global character feature distribution for solving challenging STR.

The Components of CACE

To evaluate CACE, we compare the results with decay matrix $\mathbf{D}$ or multi-scale fusion in Table 7. The first row which only relies on L_{cl} for training, obtains a performance of 60.6%. When introducing $\mathbf{D}$ in the block, the model exhibits 0.5% (61.1%) improvement. The inclusion of $\mathbf{D}$ enables the encoder to perceive the local patterns within the character and generate discriminative features for recognizing characters.

Option	Cur	M-O	Art	Ctl	Sal	M-W	Gen	AVG
1	67.1	19.6	59.8	71.6	75.4	71.2	65.0	61.4
2	70.0	20.8	62.4	72.0	75.2	65.7	65.1	**61.6**
3	68.3	19.8	56.8	69.2	74.3	67.2	64.4	60.0

Table 8: Different options of decay matrix.

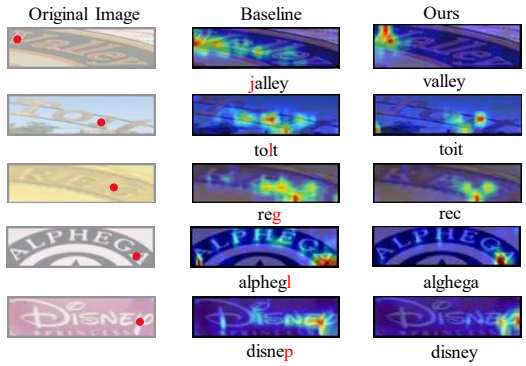
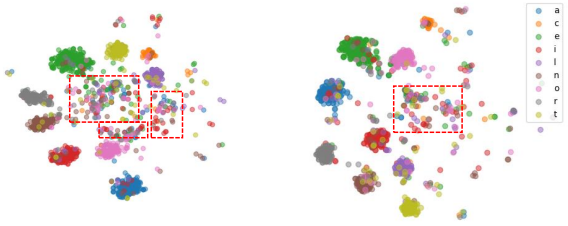


Figure 4: Visualization of attention maps in CACE. In the first column, the red point in each image is the query token. The second column images imply we use the baseline to calculate the attention scores between the red point and all points. The third column images mean CFE is used to calculate the attention scores.



(a) Without I²CL (b) With I²CL

Figure 5: Visualization of character feature distribution. The feature points in the red rectangle mean the mixture distribution. Zoom in for better visualization.

However, when only adding multi-scale fusion, CFE-Base experiences a decrease of 0.4% (60.2%). We argue that relying solely on multi-scale fusion leads the model to ignore the local character-level features, decreasing performance on challenging datasets. A total improvement of 1.0% (61.6%) is achieved when employing them together. We attribute the improvement to two reasons: 1) Decay matrix allows the encoder to perceive the morphological information at character level, helping to distinguish the characters with different appearances. 2) Multi-scale fusion provides diverse character features by integrating the features from the three stages.

The Different Options of Decay Matrix D

In Table 8, we study different options of yielding $\mathbf{D}$ and find that option 2 achieves the best performance. When considering the tokens i and j are components of a character, option 2 generates a larger $\mathbf{D}_{i,j}$. So option 2 can focus more on the character region than option 1. The accuracy rates of option 2 is 61.6%, slightly higher than that of option 1 (61.4%). However, option 3 performs less competitively than options 1 and 2 with 60.0% accuracy. The reason is that dynamic decay will generate more precise attention region for each token.

4.6 Visualization and Analysis

The Visualization of CACE

From the perspective of understanding how to focus on the character region, we visualize the self-attention scores in Figure 4. Specifically, we use the token that corresponds to the red point as the query to calculate the attention scores with all visual tokens and then reshape them to 2D. For the first three rows, while focusing on the character region, the baseline also interacts with nearby characters, leading to recognition errors. This problem can be put down to insufficient exploration of the local patterns. In contrast, CACE recognizes the texts correctly by concentrating exclusively on the character region. For the last two rows, the baseline is prone to interference from distant noise, ultimately causing errors. Conversely, CACE avoids interference from distant noise, ensuring accurate recognition. Through this, we conclude that CACE can perceive the region of each character and then model the components of the character.

The Qualitative Analysis of I²CL

To evaluate the effectiveness of our I²CL and justify its design, we use t-SNE [Van der Maaten and Hinton, 2008] to reduce the feature dimension to 2D space for visualization. Figure 5 illustrates the feature distribution of 9 easily misrecognized characters. In Figure 5(a) without I²CL, we observe that the distribution of different category characters is severely mixed, as symbolized in the red rectangle. After introducing I²CL, the phenomenon of mixed distribution can be alleviated. In addition, the distance of characters between different categories is widened as shown in Figure 5(b). By training a discrete distribution for long-term memory units, all characters of different categories can be compactly distributed around the memory units, while ensuring the separability between classes. These phenomena prove the effectiveness of I²CL and are consistent with our design.

5 Conclusion

In this paper, we notice the two problems of Large Intra-Class Variance (LICV) and Small Inter-Class Variance (SICV) in challenging texts. To address these issues, a Character Features Enriched model (CFE) is proposed to obtain the discriminative features via Character-Aware Constraint Encoder (CACE) and Intra-Inter Consistency Loss (I²CL). Firstly, CACE enables visual tokens to perceive character morphological information by introducing the decay matrix, which enhances the discriminability of character features. Secondly, I²CL helps achieve intra-class compactness and inter-class separability by learning a discrete distribution for long-term memory units. The experimental results show that CFE can not only effectively improve the performance on challenging texts, but also maintain high accuracy on simple texts, taking a further step toward STR with strong robustness.

Acknowledgments

This work is supported by the National Key Research and Development Program of China (2022YFB3104700), the National Nature Science Foundation of China (62121002, U23B2028, 62102384). This research was support by the Supercomputiong Center of the USTC. We acknowledge the support of GPU cluster built by MCC Lab of Information Science and Technology Institution, USTC.

References

[Atienza, 2021] Rowel Atienza. Vision transformer for fast and efficient scene text recognition. In *International Conference on Document Analysis and Recognition*, pages 319–334. Springer, 2021.

[Baek *et al.*, 2019] Jeonghun Baek, Geewook Kim, Junyeop Lee, Sungrae Park, Dongyoon Han, Sangdoo Yun, Seong Joon Oh, and Hwalsuk Lee. What is wrong with scene text recognition model comparisons? dataset and model analysis. In *International Conference on Computer Vision (ICCV)*, 2019.

[Baek *et al.*, 2021] Jeonghun Baek, Yusuke Matsui, and Kiyoharu Aizawa. What if we only use real datasets for scene text recognition? toward scene text recognition with fewer labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3113–3122, 2021.

[Bautista and Atienza, 2022] Darwin Bautista and Rowel Atienza. Scene text recognition with permuted autoregressive sequence models. In *European Conference on Computer Vision*, pages 178–196. Springer, 2022.

[Cheng *et al.*, 2023] Changxu Cheng, Peng Wang, Cheng Da, Qi Zheng, and Cong Yao. Lister: Neighbor decoding for length-insensitive scene text recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19541–19551, 2023.

[Chng *et al.*, 2019] Chee Kheng Chng, Yuliang Liu, Yipeng Sun, Chun Chet Ng, Canjie Luo, Zihan Ni, ChuanMing Fang, Shuaitao Zhang, Junyu Han, Errui Ding, et al. Icdar2019 robust reading challenge on arbitrary-shaped text-rrc-art. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1571–1576. IEEE, 2019.

[Du *et al.*, 2022] Yongkun Du, Zhineng Chen, Caiyan Jia, Xiaoting Yin, Tianlun Zheng, Chenxia Li, Yuning Du, and Yu-Gang Jiang. Svtr: Scene text recognition with a single visual model. *arXiv preprint arXiv:2205.00159*, 2022.

[Fan *et al.*, 2023] Qihang Fan, Huaibo Huang, Mingrui Chen, Hongmin Liu, and Ran He. Rmt: Retentive networks meet vision transformers. *arXiv preprint arXiv:2309.11523*, 2023.

[Fang *et al.*, 2021] Shancheng Fang, Hongtao Xie, Yuxin Wang, Zhendong Mao, and Yongdong Zhang. Read like humans: Autonomous, bidirectional and iterative language modeling for scene text recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7098–7107, 2021.

[Graves *et al.*, 2006] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376, 2006.

[Gupta *et al.*, 2016] Ankush Gupta, Andrea Vedaldi, and Andrew Zisserman. Synthetic data for text localisation in natural images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2315–2324, 2016.

[Jaderberg *et al.*, 2014] 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.

[Jaderberg *et al.*, 2016] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Reading text in the wild with convolutional neural networks. *International journal of computer vision*, 116(1):1–20, 2016.

[Jiang *et al.*, 2023] Qing Jiang, Jiapeng Wang, Dezhi Peng, Chongyu Liu, and Lianwen Jin. Revisiting scene text recognition: A data perspective. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20543–20554, 2023.

[Karatzas *et al.*, 2013] 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 *2013 12th international conference on document analysis and recognition*, pages 1484–1493. IEEE, 2013.

[Karatzas *et al.*, 2015] 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 *2015 13th international conference on document analysis and recognition (ICDAR)*, pages 1156–1160. IEEE, 2015.

[Kingma and Ba, 2014] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[Mishra *et al.*, 2012] Anand Mishra, Karteek Alahari, and CV Jawahar. Scene text recognition using higher order language priors. In *BMVC-British machine vision conference*. BMVA, 2012.

[Na *et al.*, 2022] Byeonghu Na, Yoonsik Kim, and Sungrae Park. Multi-modal text recognition networks: Interactive enhancements between visual and semantic features. In *European Conference on Computer Vision*, pages 446–463. Springer, 2022.

[Phan *et al.*, 2013] Trung Quy Phan, Palaiahnakote Shivakumara, Shangxuan Tian, and Chew Lim Tan. Recognizing text with perspective distortion in natural scenes. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 569–576, 2013.

[Qi and Su, 2017] Ce Qi and Fei Su. Contrastive-center loss for deep neural networks. In *2017 IEEE international conference on image processing (ICIP)*, pages 2851–2855. IEEE, 2017.

[Risnumawan *et al.*, 2014] 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.

[Sheng *et al.*, 2019] Fenfen Sheng, Zhineng Chen, and Bo Xu. Nrtr: A no-recurrence sequence-to-sequence model for scene text recognition. In *2019 International conference on document analysis and recognition (ICDAR)*, pages 781–786. IEEE, 2019.

[Shi *et al.*, 2016] Baoguang Shi, Xiang Bai, and Cong Yao. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE transactions on pattern analysis and machine intelligence*, 39(11):2298–2304, 2016.

[Shi *et al.*, 2018] 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.

[Sun *et al.*, 2023] Yutao Sun, Li Dong, Shaohan Huang, Shuming Ma, Yuqing Xia, Jilong Xue, Jianyong Wang, and Furu Wei. Retentive network: A successor to transformer for large language models. *arXiv preprint arXiv:2307.08621*, 2023.

[Van der Maaten and Hinton, 2008] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[Veit *et al.*, 2016] Andreas Veit, Tomas Matera, Lukas Neumann, Jiri Matas, and Serge Belongie. Coco-text: Dataset and benchmark for text detection and recognition in natural images. *arXiv preprint arXiv:1601.07140*, 2016.

[Wang *et al.*, 2011] 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.

[Wang *et al.*, 2021] Yuxin Wang, Hongtao Xie, Shancheng Fang, Jing Wang, Shenggao Zhu, and Yongdong Zhang. From two to one: A new scene text recognizer with visual language modeling network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14194–14203, 2021.

[Wang *et al.*, 2022] Peng Wang, Cheng Da, and Cong Yao. Multi-granularity prediction for scene text recognition. In *European Conference on Computer Vision*, pages 339–355. Springer, 2022.

[Wang *et al.*, 2023] Zixiao Wang, Hongtao Xie, Yuxin Wang, Jianjun Xu, Boqiang Zhang, and Yongdong Zhang. Symmetrical linguistic feature distillation with clip for scene text recognition. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 509–518, 2023.

[Xie *et al.*, 2022] Xudong Xie, Ling Fu, Zhifei Zhang, Zhaowen Wang, and Xiang Bai. Toward understanding wordart: Corner-guided transformer for scene text recognition. In *European Conference on Computer Vision*, pages 303–321. Springer, 2022.

[Yang *et al.*, 2022] Mingkun Yang, Minghui Liao, Pu Lu, Jing Wang, Shenggao Zhu, Hualin Luo, Qi Tian, and Xiang Bai. Reading and writing: Discriminative and generative modeling for self-supervised text recognition. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 4214–4223, 2022.

[Yu *et al.*, 2020] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12113–12122, 2020.

[Yue *et al.*, 2020] Xiaoyu Yue, Zhanghui Kuang, Chenhao Lin, Hongbin Sun, and Wayne Zhang. Robustscanner: Dynamically enhancing positional clues for robust text recognition. In *European Conference on Computer Vision*, pages 135–151. Springer, 2020.

[Zhang *et al.*, 2017] Ying Zhang, Lionel Gueguen, Ilya Zharkov, Peter Zhang, Keith Seifert, and Ben Kadlec. Uber-text: A large-scale dataset for optical character recognition from street-level imagery. In *SUNw: Scene Understanding Workshop-CVPR*, volume 2017, page 5, 2017.

[Zhang *et al.*, 2022] Xinyun Zhang, Binwu Zhu, Xufeng Yao, Qi Sun, Ruiyu Li, and Bei Yu. Context-based contrastive learning for scene text recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 3353–3361, 2022.

[Zhang *et al.*, 2023] Boqiang Zhang, Hongtao Xie, Yuxin Wang, Jianjun Xu, and Yongdong Zhang. Linguistic more: Taking a further step toward efficient and accurate scene text recognition. *arXiv preprint arXiv:2305.05140*, 2023.