The Thirty-Sixth AAAI Conference on Artificial Intelligence (AAAI-22)

Perceiving Stroke-Semantic Context: Hierarchical Contrastive Learning for Robust Scene Text Recognition

Hao Liu[1†*], **Bin Wang**[1*], **Zhimin Bao**[1], **Mobai Xue**[1,2‡], **Sheng Kang**[1,2‡],
Deqiang Jiang[1], **Yinsong Liu**[1], **Bo Ren**[1]
[1]Tencent YouTu Lab
[2]University of Science and Technology of China
{ivanhliu, bingolwang, zhiminbao, dqiangjiang, jasonysliu, timren}@tencent.com, {xmb15, ksc}@mail.ustc.edu.cn

Abstract

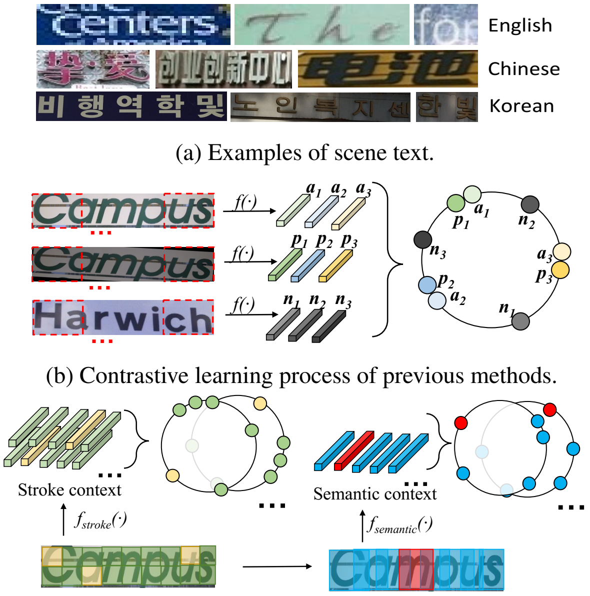
We introduce Perceiving Stroke-Semantic Context (PerSec), a new approach to self-supervised representation learning tailored for Scene Text Recognition (STR) task. Considering scene text images carry both visual and semantic properties, we equip our PerSec with dual context perceivers which can contrast and learn latent representations from low-level stroke and high-level semantic contextual spaces simultaneously via hierarchical contrastive learning on unlabeled text image data. Experiments in un- and semi-supervised learning settings on STR benchmarks demonstrate our proposed framework can yield a more robust representation for both CTC-based and attention-based decoders than other contrastive learning methods. To fully investigate the potential of our method, we also collect a dataset of 100 million unlabeled text images, named UTI-100M, covering 5 scenes and 4 languages. By leveraging hundred-million-level unlabeled data, our PerSec shows significant performance improvement when fine-tuning the learned representation on the labeled data. Furthermore, we observe that the representation learned by PerSec presents great generalization, especially under few labeled data scenes.

Introduction

Scene Text Recognition (STR) aims at reading text from the cropped text region detected by text detector, which has wide applications, ranging from translation by recognizing foreign languages to street sign recognition for autonomous. However, it is still an intractable problem as scene text image can carry both *visual* and *semantic* information, which is demonstrated in Fig. 1(a).

From visual perspective, the difficulty lies in the internal appearance property of scene text image presented, *e.g.*, text font, text color and writing style (handwritten or printed), as well as various external factors such as illumination, occlusion and low-resolution. Several previous works (Shi et al. 2016; Jaderberg et al. 2015; Yang et al. 2017; Cheng et al. 2018) focus on designing discriminative feature extractors or rectification algorithms on irregular-shape text line.

As for the semantics contained in scene text image, its content could be from any scenario with great diversity.



(a) Examples of scene text.

(b) Contrastive learning process of previous methods.

(c) Contrastive learning process of the PerSec.

Figure 1: Demonstration of scene text examples and contrastive learning processes. (a) Examples of scene text. (b) The contrastive learning process of the previous methods. The methods greedily contrast instances merely from high-level sequential features across different text images (c) The contrastive learning process of our proposed PerSec. Our method drives each element of features at high and low levels to distinguish itself from its context within one same image via hierarchical contrastive learning.

Many researches (Li et al. 2019; Sheng, Chen, and Xu 2019; Wang et al. 2020; Yu et al. 2020; Qiao et al. 2020) attempt to enhance the semantic ability of algorithm by incorporating language model into it. Nevertheless, most of previous methods are in the supervised learning paradigm, whose success is largely attributable to the availability of large amounts of annotated data. In many scenarios, the collection of data is costly, and the annotation may require expert knowledge, which hinders the applicability of such paradigm.

Recently, many self-supervised contrastive learning methods (He et al. 2020; Falcon and Cho 2020; Rao et al. 2021;

[*]Equal contribution. [†]Contact person. [‡]Interns at YouTu Lab.

Vincent et al. 2008; Chen et al. 2020; Baevski et al. 2020) in the community have achieved considerable success. Most of these methods follow the "pre-training and fine-tuning" paradigm. However, this learning paradigm has been rarely studied in the STR field and only a few methods (Aberdam et al. 2021; Chen et al. 2020) have been proposed.

The pioneer work (Aberdam et al. 2021) (Fig. 1(b)) sliding-windowed every few consecutive frames in a high-level sequence feature map output by encoder $f(\cdot)$ (*e.g.,* CNN) as an instance and regarded instances from two augmentations of one input image as anchors ($\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{a}_3$) and corresponding positive instances ($\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3$). Otherwise, the instances from another text images are constructed as negative ones ($\boldsymbol{n}_1, \boldsymbol{n}_2, \boldsymbol{n}_3$). The contrastive learning was then performed on them. This pipeline conducting cross-sample contrastive task may have the following three drawbacks. The first one lies in the positive pair from two augmentations at the same location, which may suffer from the instance misalignment problem. Secondly, the selection upon negative pairs is cross different samples with uncontrollable discrepancy in writing style or content. As a consequence, the semantic continuity within one image could be corrupted and the model could not yield sequential representations with sufficiently discriminative semantics, which is vital for the STR task. Last but not least, all contrasting operations are conducted on high-level features, while low-level ones containing partial pattern and stroke information of text are not paid enough attention.

Based on the above intuition, we propose a novel hierarchical contrastive learning strategy to learn robust representation from unlabeled data for STR task, termed Perceiving Stroke-Semantic Context (PerSec). Compared with the previous method greedily contrasting instances from high-level sequential features of different text images, our PerSec (Fig. 1(c)) aims to learn a series of sub-hyperspheres (illustrated by circles) where each element of features at different levels to distinguish itself from its context within one same image via hierarchical contrasting, which is more consistent with the process of text recognition. For low-level features extracted by $f_{stroke}(\cdot)$, each element can be either an anchor (in yellow color) or one of *stroke context* (in green color). In a similar sense, contrast is performed between each anchor (in red color) from high-level sequential features extracted by $f_{semantic}(\cdot)$ and its *semantic context* (in blue color). Through this way, the learned representation can be discriminative in both low-level stroke space and high-level semantic space.

To implement the above process, we randomly mask the anchor element of features, which is similar to masked language modeling in Bert (Devlin et al. 2018) to discern "slow features" (Wiskott and Sejnowski 2002) from context. In the scene text image, the"slow features" can be either the stroke pattern (*e.g.,* the radical of Chinese character) or content semantics. However, unlike Bert leveraging pre-exist vocabulary as input unit, the feature space in our case could be extremely complex. If we directly perform contrast on the raw features, the quality of learned representation would be sensitive to various distractors (*e.g.*, noise, blur and *etc.*). Alternatively, we propose a *context perceiver* module to reduce the difficulty of learning in high and low level feature space. More concretely, context perceiver maps input features into learnable discrete units as pseudo labels. Then, the contrast is performed between each element of feature and pseudo labels after the context information is aggregated to feature elements. Benefiting from the hierarchical contrast mechanism, our PerSec can achieve better performance than other methods under both un- and semi-supervised learning settings, as validated by experimental results.

To further explore the potential of the proposed learning paradigm, we collect a dataset of 100 million unlabeled text images, named UTI-100M, covering 5 scenes and 4 languages. By leveraging hundred-million-level unlabeled data, our PerSec shows inspiring transferability and significant performance improvement on learned representation quality. In summary, our contributions are:

- We present a new hierarchical self-supervised learning method, named Perceiving Stroke-Semantic Context (PerSec), which can simultaneously learn robust representation from stroke and semantic context.
- We coin a novel context perceiver module equipped with learnable quantizer and context aggregator, which can effectively make each element distinguishable from its context. And it can also serve as a plug-and-play module to augment most prevalent text recognizers.
- Experimental results on public benchmarks demonstrate that our method can achieve state-of-the-art performance in both un- and semi-supervised learning settings.
- We collect a large scale dataset (UTI-100M) with hundred-million-level unlabeled real data, which can substantially boost the performance of PerSec.

Method

Architecture Overview

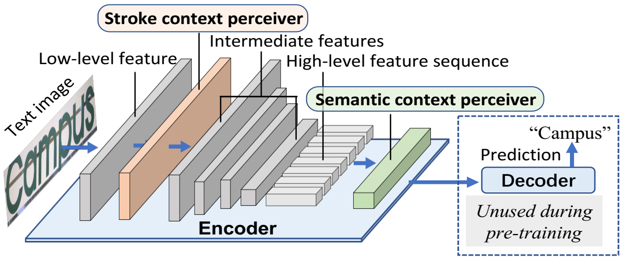


Figure 2: Architecture of the proposed PerSec. For the commonly used "encoder-decoder" STR pipeline, the encoder is first pretrained by the PerSec equipped with stroke and semantic context perceivers through hierarchical self-supervised learning. Afterward, the decoder is appended upon the pretrained encoder to fine-tune the whole pipeline.

Scene text recognition (STR) often follows the "encoder-decoder" pipeline. Given an input text image, the encoder extracts features from it as image representations, and the decoder is responsible for decoding text content from representations. As shown in Fig. 2, the encoder is firstly pre-trained on unlabeled data by our proposed Perceiving

Layer	Configuration			Output size
	k, c	*s*	*p*	
conv_1	3×3, 128	(1,1)	1	$H \times W$
maxpool_1	2×2	(2,2)	0	$H/2 \times W/2$
conv_2	3×3, 128	(1,1)	1	$H/2 \times W/2$
maxpool_2	2×2	(2,2)	0	$H/4 \times W/4$
conv_3	3×3, 256	(1,1)	1	$H/4 \times W/4$
conv_4	3×3, 512	(1,1)	1	$H/4 \times W/4$
maxpool_3	2×1	(2,1)	0	$H/8 \times W/4$
conv_5	3×3, 512	(1,1)	1	$H/8 \times W/4$
maxpool_4	2×1	(2,1)	0	$H/16 \times W/4$
conv_6	2×2, 512	(1,1)	0	$H/32 \times W/4$

Table 1: CNN-based encoder taking input image with H height and W width. *k, c* represent the *k* size convolution kernel with *c* dimension, *s* and *p* represent the stride and padding respectively, output size is in height × width.

Stroke-Semantic Context (PerSec) method. The PerSec is equipped with stroke and semantic context perceivers, which aims at learning robust text image representation through hierarchical contrastive learning. For the downstream STR task, we append the decoder module, which can be either CTC-based or attention-based, upon pretrained encoder and fine-tune the whole pipeline on the labeled data.

Base Encoder

CNN-based Encoder. In this work, we firstly adopt the Convolutional Neural Network (CNN) as the base encoder, of which the detailed architecture is given in Tab. 1. It consists of 6 convolutional layers and 4 max-pooling layers with a ReLU non-linearity layer interpolated after each convolutional layer. Considering the trade-off between performance and computational complexity, low-level stroke encoder $f_{stroke}(\cdot)$ is composed of "conv_1∼ conv_3" layers, which is appended by stroke context perceiver. For the high-level semantic encoder $f_{semantic}(\cdot)$, it consists of "conv_4∼ conv_6" layers, of which the output feature is processed by semantic context perceiver.

ViT-based Encoder. Compared with CNN introducing strong inductive bias (*e.g.*, local behavior), Vision Transformer (ViT) (Dosovitskiy et al. 2020) prefers to learn suitable inductive bias from data, which has shown its superior performance on computer vision tasks. Thus, we also adopt ViT-based backbone as another base encoder alternative. More concretely, we introduce Pyramid Vision Transformer (PVT) (Wang et al. 2021) and adapt it to the STR task. The details are given in Tab. 2. Correspondingly, layers of "Stage1" construct the low-level stroke encoder $f_{stroke}(\cdot)$ while the high-level semantic encoder $f_{semantic}(\cdot)$ consists of the rest "Stage2∼Stage4".

Context Perceiver

In order to learn robust hierarchical text representation, we design Context Perceiver (CP) module and apply it on the output features of both low-level stroke encoder $f_{stroke}(\cdot)$ and high-level semantic encoder $f_{semantic}(\cdot)$. Compared with the previous work SeqCLR (Aberdam et al. 2021) performing cross-sample contrasting, our proposed CP enables each element of input features to distinguish itself from context. That is, contrast operation merely happens within the feature elements from one same sample. As illustrated in Fig. 3, our CP has a dual-branch structure including "context aggregator" (pink part) and "quantizer" (blue part).

	Layer	Configuration	Output Size
Stage1	PatchEmb_1	$k=7, c=64$ $s=(4,4), p=3$	$\frac{H}{4} \times \frac{W}{4}$
	TrmEnc_1_x	$\begin{bmatrix} R_1=4 \\ N_1=1 \\ E_1=4 \end{bmatrix} \times 2$	
Stage2	PatchEmb_2	$k=3, c=128,$ $s=(2,1), p=1$	$\frac{H}{8} \times \frac{W}{4}$
	TrmEnc_2_x	$\begin{bmatrix} R_2=4 \\ N_2=2 \\ E_2=4 \end{bmatrix} \times 2$	
Stage3	PatchEmb_3	$k=3, c=320,$ $s=(2,1), p=1$	$\frac{H}{16} \times \frac{W}{4}$
	TrmEnc_3_x	$\begin{bmatrix} R_3=2 \\ N_3=4 \\ E_3=4 \end{bmatrix} \times 2$	
Stage4	PatchEmb_4	$k=3, c=512$ $s=(2,1), p=1$	$\frac{H}{32} \times \frac{W}{4}$
	TrmEnc_4_x	$\begin{bmatrix} R_4=1 \\ N_4=8 \\ E_4=4 \end{bmatrix} \times 2$	

Table 2: ViT-based encoder (Wang et al. 2021) taking input image with H height and W width. "PatchEmb" and "TrmEnc" are short for "patch embedding" and "transformer encoder" submodule. $R_\sim$, $N_\sim$ and $E_\sim$ represent the reduction ratio, head number and expansion ratio respectively. For the patch embedding of stage, *k* and *c* are the kernel parameters while *s* and *p* represent the stride and padding respectively, output size is in height × width.

Context Aggregator. To drive each element distinguish from context, CP first conducts masking on a proportion of features $\mathbf{F} \in \mathbb{R}^{w \times h \times d}$ output by encoder $f_\sim(\cdot)$, in the similar spirit with masked language model (Devlin et al. 2018). Afterwards, the masked features $\mathbf{F}^{'} \in \mathbb{R}^{w \times h \times d}$ applied by position encoding are fed into context aggregator with N stacked transformer (Vaswani et al. 2017) blocks involving Windowed Multi-Head Self-Attention (W-MHSA) and Feed-Forward Networks (FFN) as *de facto* components. Considering the CP should reconcile the context aggregation from both low-level feature maps and high-level feature sequences, we in this work adopt 2-D position encoding (Zhang and Yang 2021) which is more flexible to handle input features with arbitrary size. It can be denoted as:

$$\hat{\mathbf{F}} = \mathbf{F} * \sigma(\mathrm{DWConv}(\mathbf{F})), \tag{1}$$

where $\sigma(\cdot)$ is sigmoid function and $\mathrm{DWConv}(\cdot)$ is the 3×3 depth-wise convolution.

In the features of a scene text image, the desired "slow features" are often contained in the local context. In low-level stroke feature space, the "slow features" represent

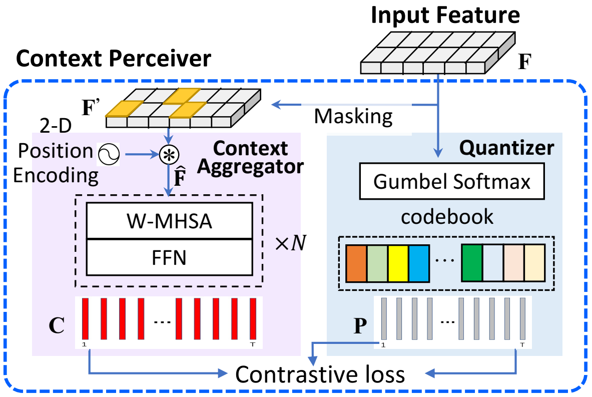


Figure 3: Architecture of the proposed context perceiver. The context perceiver has a dual-branch structure including "context aggregator" and "quantizer".

stroke patterns (*e.g.*, radicals of Chinese) while they represent semantic segment information in high-level semantic feature space. To better capture "slow features" from local context, we design the W-MHSA, which is different from vanilla MHSA: each element of the masked feature map is only aggregated contextual ones within a controllable range on it. Specifically, W-MHSA is defined as:

$$\text{W-MHSA}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}} + \mathbf{M}_{win}\right)\mathbf{V}. \tag{2}$$

Input features are first linear transformed to obtain queries $\mathbf{Q} \in \mathbb{R}^{T\times d}$, keys $\mathbf{K} \in \mathbb{R}^{T\times d}$ and values $\mathbf{V} \in \mathbb{R}^{T\times d}$, where $T = w \cdot h$. $\mathbf{M}_{win} \in \mathbb{R}^{T\times T}$ stands for window mask limiting the aggregation scope of self-attention. In $\mathbf{M}_{win}$, value is set to $-\infty$ if the corresponding feature element locates outside the window with size ω, where $1 \leq \omega \leq T$. As for the FFN, we inherit the similar layer with vanilla transformer (Vaswani et al. 2017). Note, each transformer block is also equipped with layer normalization and residual connection. Finally, the features $\mathbf{C} \in \mathbb{R}^{T\times d}$ with local context aggregated are obtained.

Quantizer. Inspired by the success of product quantization (Jegou, Douze, and Schmid 2010) coding high-dimensional visual features in fast image retrieval (Yu et al. 2018), we introduce it to map the input feature into discrete units serving as pseudo labels. In detail, the quantizer chooses representations from G codebooks where each of them contains V entries $e \in \mathbb{R}^{V\times d/G}$, and fuses them by concatenation. We assign one Gumbel-Softmax (Jang, Gu, and Poole 2016) operator to select the entries of each codebook separately, and then concatenate them into a vector of dimension d. For the j-th entry in one codebook, the probability for selecting it is:

$$p_j = \frac{\exp\left(l_j + v_j\right)/\tau}{\sum_{k=1}^{V}\exp\left(l_k + v_k\right)/\tau}, \tag{3}$$

where l is the mapped logit from the input features and $v = -\log(-\log(u))$, u are uniform samples from $\mathcal{U}(0, 1)$, τ is a non-negative temperature. Through this way, pseudo labels $\mathbf{P} \in \mathbb{R}^{T\times d}$ are yielded and the contrastive task is performed between $\mathbf{P}$ and context aggregated feature $\mathbf{C}$.

Pre-training Strategy

Masking Tricks. As introduced above, in context perceiver, the elements of stroke feature maps and semantic feature sequence are performed masking operation separately in the pre-training phase. Considering the 2-D shape of stroke feature maps, we also apply a 2-D mask in size of $m_{\text{low}} \times m_{\text{low}}, 1 \leq m_{\text{low}} \leq h$ on it, where h is the height of the feature map. Note, there may exist multiple masks on one feature map, and the number of them is in a certain proportion p_{low} to the feature map size $w \cdot h$. As for the high-level feature sequence with T time-steps, we set the mask to m_{high} consecutive time steps, where $1 \leq m_{\text{high}} \leq T$. Respectively, the mask number is $p_{\text{high}} \cdot T$. If the elements of feature are masked, they would be replaced with a trainable feature vector shared between all masked ones. Note, we make sure each mask never has overlap with each other.

Loss functions. During pre-training stage, our PerSec learns representations of scene text images by solving the contrastive tasks in stroke space and the semantic space simultaneously. As defined in Eqn. (4), we introduce the contrastive loss ($\mathcal{L}^{(\text{con})}_{stroke}, \mathcal{L}^{(\text{con})}_{semantic}$) and diversity loss ($\mathcal{L}^{(\text{div})}_{semantic}, \mathcal{L}^{(\text{div})}_{semantic}$) (Baevski et al. 2020), which is combined by weight parameters α and β, as our hierarchical contrastive learning loss:

$$\mathcal{L} = \overbrace{\mathcal{L}^{(\text{con})}_{stroke} + \alpha\mathcal{L}^{(\text{div})}_{stroke}}^{stroke-level} + \underbrace{\mathcal{L}^{(\text{con})}_{semantic} + \beta\mathcal{L}^{(\text{div})}_{semantic}}_{semantic-level}. \tag{4}$$

Specifically, the contrastive loss is denoted as:

$$\mathcal{L}^{(\text{con})}_{\sim}(i) = -\log\frac{\exp\left(\text{sim}\left(\boldsymbol{c}_i, \boldsymbol{p}_i\right)/\tau\right)}{\sum_{k\in\mathbf{I}_{\text{mask}}}\exp\left(\text{sim}\left(\boldsymbol{c}_i, \boldsymbol{p}_k\right)/\tau\right)}, \tag{5}$$

where $\boldsymbol{c}_i$ and $\boldsymbol{p}_i$ are the i-th masked element from context aggregated feature $\mathbf{C}$ and pseudo labels $\mathbf{P}$ respectively, which construct the positive pair. $\boldsymbol{p}_k$ represents the k-th element from other masked ones containing both positive and negative samples and $\mathbf{I}_{\text{mask}}$ are the index set of all masked elements. sim is the cosine similarity which can be computed as $\text{sim}(\mathbf{a}, \mathbf{b}) = \mathbf{a}^\top\mathbf{b}/\|\mathbf{a}\|\|\mathbf{b}\|$. This loss identifies i-th element from its distractors falling in the same mask.

The diversity loss aims to utilize each entry in the codebook as equally as possible. Given G codebooks in which each of them contains V entries, the diversity loss maxmizes the entropy of the averaged probability of selecting each entry of each codebook $p_{g,v}$. It can be defined as:

$$\mathcal{L}^{(\text{div})}_{\sim} = \frac{1}{GV}\sum_{g=1}^{G}\sum_{v=1}^{V} p_{g,v}\log p_{g,v}. \tag{6}$$

Experiments

Datasets and Metrics

In this work, we adopt STR public datasets to evaluate the performance of the pre-trained model. The datasets cover three categories: 1) regular scene-text datasets including IC13 (Karatzas et al. 2013), IIIT5K (Mishra, Alahari, and Jawahar 2012) and SVT (Wang, Babenko, and Belongie 2011) ;2) irregular scene-text datasets: IC15 (Karatzas et al. 2015), SVTP (Phan et al. 2013) and CT80 (Risnumawan et al. 2014); 3) handwritten text datasets: IAM (Marti and Bunke 2002) and CVL (Kleber et al. 2013) . For regular and irregular scene-text recognition, we exploit the synthetic dataset ST (Gupta, Vedaldi, and Zisserman 2016) and MJ (Jaderberg et al. 2014) as training sets. As for the metric for evaluation, we adopt word-level accuracy (Acc) for all experiments, similar to work (Aberdam et al. 2021).

A New Dataset: UTI-100M

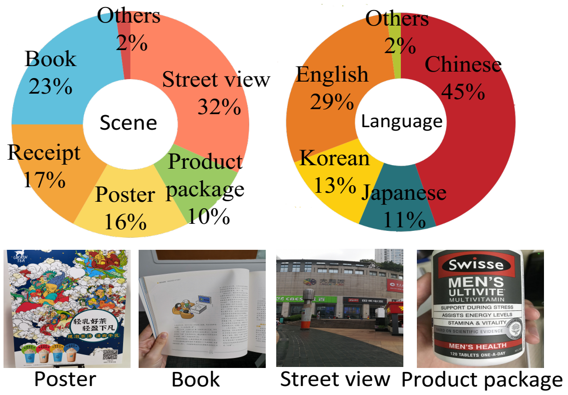


Figure 4: UTI-100M dataset.

As suggested in literature (Baek, Matsui, and Aizawa 2021), training model on real data can yield better results than synthetic data. Therefore, we collect a large-scale real dataset containing about 100 million unlabeled text line images, named Unlabeled Text Image 100M (UTI-100M), to explore the potential of the proposed hierarchical contrastive learning paradigm.

More concretely, the data collection and processing are conducted as follows. First, we collect round 1 million real scene images captured by mobile camera device, covering 5 scenes, *i.e.,* street view, receipt, product package, book, and poster, which are mainly written in Chinese, English, Japanese and Korean. Then we utilize one of prevalent text detectors, DBNet (Liao et al. 2020), to detect text instances from scene images and warp them into rectangular text line images. Considering the warped text lines have a large variety in image scale, all the images are normalized to (32×384) through cropping and stitching. The data proportion of each scene is shown in Fig. 4. In particular, the text detector inevitably produces a small amount of noisy samples, such as misaligned text or background with text-like patterns, whereas their influence on the self-supervised training is neglectable.

Training Configurations

Pre-training. In the pre-training stage, we normalize the input images to 32×384. In both stroke and semantic context perceivers, the head number of W-MHSA is set to 8 while the dimensions of all linear layers are set to 128. As for the window size ω in W-MHSA, we empirically set it to $1/2$ of input feature height for stroke context perceiver and set it to 10 for semantic context perceiver.

In the context perceivers, we set mask proportion p_{low} to 0.2 and p_{high} to 0.15 at stroke and semantic levels, respectively. Correspondingly, the size of low-level stroke feature mask m_{low} and size of the high-level one m_{high} are both set to 1. For quantizers at stroke and semantic level, there are 2 codebooks with 256 entries in each. In the Eqn. (4), loss weight parameter α is set to 0.2, while β is set to 0.1.

The proposed self-supervised learning framework is implemented by Pytorch (Paszke et al. 2019). And the training batch size is set to 2,048. All experiments are conducted on a total of 32 NVIDIA A100 GPUs with 80 GB RAM each. We train for 3.5 days on our proposed UTI-100M dataset and for round 1.2 days on the public datasets. Image augmentations are also employed in the pre-training phase, including brightness adjustment, noise disturbance, quality reduction, image stretching, distortion, and perspective as described in (Luo et al. 2020). We use Adam (Kingma and Ba 2014) optimizer and warm-up strategy with 1e-4 as the initial learning rate. Note, we scale the backpropagated gradient at stroke context perceiver by 0.2 to stabilize the model training.

Fine-tuning. We inherit the configurations of CTC-based and attention-based decoders from SeqCLR (Aberdam et al. 2021), which are separately appended upon the pre-trained encoder. Afterwards, we fine-tune the whole pipeline on the labeled data under un- and semi-supervised settings. More concretely, we remove the "quantizer" in context perceiver and reserve the "context aggregator" as a part of encoder for fine-tuning. At this stage, all input images are normalized to 32×128 and the irregular text images are performed the similar transformation in SeqCLR (Aberdam et al. 2021) before fed into the model. We use SGD optimizer with the 5e-3 initial learning rate. The pipeline with CTC-based decoder is optimized by CTC loss (Shi, Bai, and Yao 2016) while the one with attention-based decoder utilizes cross-entropy loss (Gehring et al. 2017) for optimization. The training batch size is 2,048 and fine-tuning is also implemented by Pytorch, which is conducted on the same platform with pre-training.

Representation Quality Evaluation

To evaluate the quality of text representation learned on unlabeled data by our PerSec framework, we fine-tune the decoder (either CTC-based or attention-based) on top of encoder under both un- and semi-supervised learning settings. Note, in this fine-tuning stage, the trained context aggregators in context perceivers are reserved as parts of base encoder while the quantizers are removed.

Unsupervised Learning. Under this setting, the base encoder is unsupervised pre-trained, in which all parameters are then frozen, and we only train a decoder with labeled data on top of it. The comparison results between our PerSec and other state-of-the-art methods are shown in Tab. 4.

Method	Decoder	Regular			Irregular			Handwritten	
		IC13	IIIT5K	SVT	IC15	SVTP	CT80	IAM	CVL
Supervised baseline (Random init.)	CTC	83.3	75.8	76.4	57.4	66.0	57.3	74.6	75.2
SimCLR (Chen et al. 2020)		79.4	69.1	-	-	-	-	65.0	74.1
SeqCLR (Aberdam et al. 2021)		86.3	80.9	-	-	-	-	76.7	76.9
PerSec-CNN		87.9	82.2	83.1	62.3	70.4	63.5	77.9	78.1
PerSec-CNN + UTI-100M		90.1	83.9	83.7	66.7	72.9	66.7	78.3	79.2
PerSec-ViT		89.7	83.7	83.0	64.6	71.4	65.2	78.0	78.8
PerSec-ViT + UTI-100M		**92.8**	**85.4**	**86.1**	**70.3**	**73.9**	**69.2**	**79.9**	**80.5**
Supervised baseline (Random init.)	Attention	85.4	83.1	80.8	64.7	69.2	64.9	77.8	77.3
SimCLR (Chen et al. 2020)		86.3	80.9	-	-	-	-	70.7	75.7
SeqCLR (Aberdam et al. 2021)		87.9	82.9	-	-	-	-	79.9	77.8
PerSec-CNN		88.9	84.2	82.4	68.2	73.6	68.4	80.8	80.2
PerSec-CNN + UTI-100M		89.7	85.5	85.4	71.7	76.2	70.1	82.3	81.4
PerSec-ViT		89.2	85.2	84.9	70.9	75.9	69.1	81.8	80.8
PerSec-ViT + UTI-100M		**94.2**	**88.1**	**86.8**	**73.6**	**77.7**	**72.7**	**83.7**	**82.9**

Table 3: Word accuracy (in %) comparison between PerSec and state-of-the-art methods under semi-supervised setting.

Method	Dec.	Scene-Text Dataset		
		IIIT5K	IC03	IC13
SimCLR (Chen et al. 2020)	CTC	0.3	0.0	0.3
SeqCLR (Aberdam et al. 2021)		35.7	43.6	43.5
PerSec-CNN		37.9	45.7	46.4
PerSec-CNN + UTI-100M		39.2	47.8	48.2
PerSec-ViT		38.4	46.2	46.7
PerSec-ViT + UTI-100M		**43.4**	**50.6**	**51.2**
SimCLR (Chen et al. 2020)	Attn.	2.4	3.7	3.1
SeqCLR (Aberdam et al. 2021)		49.2	63.9	59.3
PerSec-CNN		50.7	65.7	61.1
PerSec-CNN + UTI-100M		53.6	67.7	63.2
PerSec-ViT		52.3	66.6	62.3
PerSec-ViT + UTI-100M		**55.4**	**70.9**	**66.2**

Table 4: Word accuracy (in %) comparison between PerSec and state-of-the-art methods under unsupervised setting."Dec."and "Attn." are short for "Decoder" and "Attention"."-CNN" and "-ViT" represent "CNN-based encoder" and "ViT-based encoder". "+UTI-100M" means introducing UTI-100M as extra pre-training dataset.

We can observe that our method can achieve round 2% average accuracy improvement than the second best method SeqCLR (Aberdam et al. 2021) when adopting both CTC-based and attention-based decoders on all three datasets. We attribute the improvement to the hierarchical contrastive learning mechanism in our PerSec. Besides, the Tab. 4 also verifies that ViT base encoder in PerSec can yield better results than CNN base encoder, which is reasonable on account of more flexible inductive bias in ViT. Especially when the unsupervised pre-training dataset is extended by introducing large-scale UTI-100M dataset, the results in Tab. 4 witness more than 4% average accuracy increase on all datasets by using PerSec-ViT, which surpass other methods by a large margin.

Semi-supervised Learning. We further unfreeze the parameters of base encoder and fine-tune it together with decoder. Tab. 3 shows the performance comparison between our PerSec and other methods. "Superived baseline" adopts CNN as encoder with context aggregator inserted, in which parameters are randomly initialized. On the regular and irregular scene-text datasets as well as handwritten dataset, the comparison results between "PerSec-CNN" and "Supervised baseline" demonstrate the effectiveness and generalization ability of our hierarchical contrastive learning paradigm. Compared with the SeqCLR, PerSec-CNN can outperform it by round 2%. By pre-training on a larger scale dataset UTI-100M, the performance of PerSec can be further improved under the semi-supervised learning setting, especially using ViT as base encoder. Additionally, we also conduct more exploratory experiments to study the performance of our PerSec when the whole network is only fine-tuned on a small amount of labeled data.

Method	STCP	SECP	Decoder	
			CTC	Attention
PerSec-CNN PerSec-ViT	✗	✗	83.3 84.2	85.4 86.2
PerSec-CNN PerSec-ViT	✗	✓	86.3 87.5	87.1 88.0
PerSec-CNN PerSec-ViT	✓	✗	86.0 87.2	86.7 87.9
PerSec-CNN PerSec-ViT	✓	✓	87.9 89.7	88.9 89.2

Table 5: Ablation studies of context perceivers on IC13 dataset. "STCP" and "SECP" represent stroke and semantic context perceivers respectively.

Analysis on PerSec

Ablation Studies on Context Perceivers. Context perceiver is the core module of our PerSec, we thus ablate either stroke context perceiver (STCP) or semantic context perceiver (SECP) or both of them in the PerSec and fine-tune the whole network on IC13 dataset to investigate the effect of each context perceiver to the semi-supervised learning.

The results are summarized in Tab. 5. From the results, we can find that solely removing STCP or SECP can be detrimental to the PerSec, but still can achieve better results than the one with both perceivers removed. We also remove the quantizer and only contrast the input feature with output of context aggregator. Unfortunately, the model can not converge in this configuration.

Pre-training Losses. As shown in Fig. 5, we observe that contrastive loss at semantic-level presents faster convergence and lower values than that of stroke-level. The diversity loss of semantic-level is slightly higher than that of stroke-level, indicating that stroke context perceiver has higher codebook usage than semantic one in pre-training.

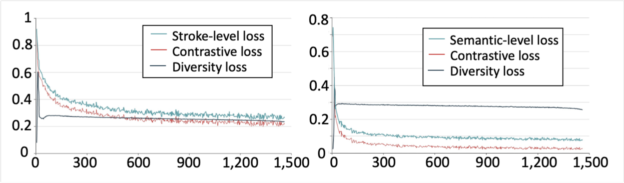


Figure 5: Training losses of context perceivers.

What does PerSec Learn from Stroke-Semantic Context? Context aggregator and quantizer are two crucial components of PerSec. In Fig. 6(a) and (b), we separately visualize the attention maps from W-MHSA in context aggregator pre-trained on UTI-100M at stroke and semantic levels. For brevity, we selectively visualize attention maps from 2 heads and the non-overlapped windows. Each box represents a local region performed W-MHSA, where the window size equals to the box size. The attention map (after softmax) in terms of the box center is resized by bilinear interpolation and projected on the raw image. It can be observed that, through our PerSec self-supervised learning, stroke-level attention can spontaneously focus on the stroke of characters while the semantic-level attention is often active at each whole character region.

The learnable codebooks in the quantizer have a maximum capacity of $256 \times 256 = 65,536$ codes. We also visualize their t-SNE (Van der Maaten and Hinton 2008) plot in Fig. 6(d) and (e) showing the clusters formed by the learned codes. Both stroke and semantic level codebooks show the desirable diversity. Moreover, the semantic-level codebook has more inter-cluster overlap than the stroke-level one, which means the storke-level quantizer has higher codebook usage. This is consistent with the behavior of diversity loss in pre-training phase. We attribute this phenomenon to that the stroke feature space is more complicated but extracted by model with shallow depth.

From Fig. 6(f), we find if the stroke context perceiver (STCP) is removed in our method, the stroke features of test data from IC13 become less discriminative, which demonstrates the indispensability of STCP. Besides, in Fig. 6(c), we visualize the locations of some stroke features corresponding to index $(3, 135)$(red) and $(69, 23)$(green) in the stroke codebook on the raw image, which locate at the stroke joints and endpoints separately. This phenomenon vividly demonstrates our model can well capture the stroke patterns of text cases through pre-training on unlabeled data.

Method	Regular			Irregular		
	IC13	IIIT5K	SVT	IC15	SVTP	CT80
RobustScanner	92.8	94.6	88.1	76.9	79.5	89.7
RobustScanner†	95.1	95.2	91.2	78.1	81.2	91.3
SATRN	94.2	94.7	92.1	82.1	86.4	87.6
SATRN†	97.2	96.3	94.6	84.4	89.5	90.2
SAR	91.2	91.3	84.7	70.7	76.9	83.0
SAR†	93.7	95.6	90.1	76.3	81.1	88.2
NRTR	93.6	93.7	90.2	74.5	78.3	86.1
NRTR†	96.1	95.1	91.5	77.3	80.4	89.1

Table 6: Performance improvements on the state-of-the-arts including RobustScanner (Yue et al. 2020), SATRN (Lee et al. 2020), SAR (Li et al. 2019) and NRTR (Sheng, Chen, and Xu 2019). † indicates that the corresponding encoder is pre-trained on UTI-100M by our PerSec and fine-tuned on MJ and ST datasets.

Improvements on SOTA Methods. As the context perceiver in our PerSec is a plug-and-play module, we also employ PerSec to pre-train the encoders of many prevalent text recognizers, including RobustScanner (Yue et al. 2020), SATRN (Lee et al. 2020), SAR (Li et al. 2019) and NRTR (Sheng, Chen, and Xu 2019). The results in Tab. 6 witness an obvious performance boosting, which verifies the adaptability of PerSec.

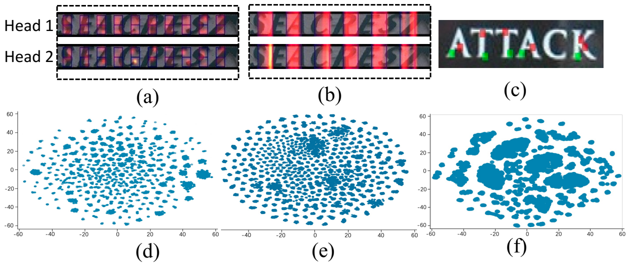


Figure 6: (a) Stroke-level attention. (b) Semantic-level attention. (c) Samples with different indexes in stroke codebook. (d) Stroke codebook. (e) Semantic codebook. (f) Stroke features w/o STCP. Best viewed in color and zoomed in.

Conclusions

In this work, we propose a novel Perceiving Stroke-Semantic (PerSec) to learn robust scene text representations via hierarchical contrastive learning on unlabeled text image data. The effectiveness of this learning paradigm has been verified by the experimental results on STR benchmarks. We also contribute a hundred-million-level UTI-100M dataset for pre-training, which can further boost the performance of PerSec. Moreover, our PerSec presents great generalization, especially under few labeled data scenes.

References

Aberdam, A.; Litman, R.; Tsiper, S.; Anschel, O.; Slossberg, R.; Mazor, S.; Manmatha, R.; and Perona, P. 2021. Sequence-to-sequence contrastive learning for text recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 15302–15312.

Baek, J.; Matsui, Y.; and Aizawa, K. 2021. 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*, 3113–3122.

Baevski, A.; Zhou, Y.; Mohamed, A.; and Auli, M. 2020. wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations. *Advances in Neural Information Processing Systems*, 33.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, 1597–1607. PMLR.

Cheng, Z.; Xu, Y.; Bai, F.; Niu, Y.; Pu, S.; and Zhou, S. 2018. Aon: Towards arbitrarily-oriented text recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5571–5579.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Falcon, W.; and Cho, K. 2020. A framework for contrastive self-supervised learning and designing a new approach. *arXiv preprint arXiv:2009.00104*.

Gehring, J.; Auli, M.; Grangier, D.; Yarats, D.; and Dauphin, Y. N. 2017. Convolutional Sequence to Sequence Learning. In *Proc. of ICML*.

Gupta, A.; Vedaldi, A.; and Zisserman, A. 2016. Synthetic data for text localisation in natural images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2315–2324.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9729–9738.

Jaderberg, M.; Simonyan, K.; Vedaldi, A.; and Zisserman, A. 2014. Synthetic data and artificial neural networks for natural scene text recognition. *arXiv preprint arXiv:1406.2227*.

Jaderberg, M.; Simonyan, K.; Zisserman, A.; and Kavukcuoglu, K. 2015. Spatial transformer networks. *arXiv preprint arXiv:1506.02025*.

Jang, E.; Gu, S.; and Poole, B. 2016. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*.

Jegou, H.; Douze, M.; and Schmid, C. 2010. Product quantization for nearest neighbor search. *IEEE transactions on pattern analysis and machine intelligence*, 33(1): 117–128.

Karatzas, D.; Gomez-Bigorda, L.; Nicolaou, A.; Ghosh, S.; Bagdanov, A.; Iwamura, M.; Matas, J.; Neumann, L.; Chandrasekhar, V. R.; Lu, S.; et al. 2015. ICDAR 2015 competition on robust reading. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, 1156–1160. IEEE.

Karatzas, D.; Shafait, F.; Uchida, S.; Iwamura, M.; i Bigorda, L. G.; Mestre, S. R.; Mas, J.; Mota, D. F.; Almazan, J. A.; and De Las Heras, L. P. 2013. ICDAR 2013 robust reading competition. In *2013 12th International Conference on Document Analysis and Recognition*, 1484–1493. IEEE.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kleber, F.; Fiel, S.; Diem, M.; and Sablatnig, R. 2013. Cvl-database: An off-line database for writer retrieval, writer identification and word spotting. In *2013 12th international conference on document analysis and recognition*, 560–564. IEEE.

Lee, J.; Park, S.; Baek, J.; Oh, S. J.; Kim, S.; and Lee, H. 2020. On recognizing texts of arbitrary shapes with 2D self-attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 546–547.

Li, H.; Wang, P.; Shen, C.; and Zhang, G. 2019. Show, attend and read: A simple and strong baseline for irregular text recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 8610–8617.

Liao, M.; Wan, Z.; Yao, C.; Chen, K.; and Bai, X. 2020. Real-time scene text detection with differentiable binarization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 11474–11481.

Luo, C.; Zhu, Y.; Jin, L.; and Wang, Y. 2020. Learn to augment: Joint data augmentation and network optimization for text recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13746–13755.

Marti, U.-V.; and Bunke, H. 2002. The IAM-database: an English sentence database for offline handwriting recognition. *International Journal on Document Analysis and Recognition*, 5(1): 39–46.

Mishra, A.; Alahari, K.; and Jawahar, C. 2012. Scene text recognition using higher order language priors. In *BMVC-British Machine Vision Conference*. BMVA.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32: 8026–8037.

Phan, T. Q.; Shivakumara, P.; Tian, S.; and Tan, C. L. 2013. Recognizing text with perspective distortion in natural scenes. In *Proceedings of the IEEE International Conference on Computer Vision*, 569–576.

Qiao, Z.; Zhou, Y.; Yang, D.; Zhou, Y.; and Wang, W. 2020. Seed: Semantics enhanced encoder-decoder framework for scene text recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13528–13537.

Rao, H.; Xu, S.; Hu, X.; Cheng, J.; and Hu, B. 2021. Augmented skeleton based contrastive action learning with momentum lstm for unsupervised action recognition. *Information Sciences*, 569: 90–109.

Risnumawan, A.; Shivakumara, P.; Chan, C. S.; and Tan, C. L. 2014. A robust arbitrary text detection system for natural scene images. *Expert Systems with Applications*, 41(18): 8027–8048.

Sheng, F.; Chen, Z.; and Xu, B. 2019. NRTR: A no-recurrence sequence-to-sequence model for scene text recognition. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, 781–786. IEEE.

Shi, B.; Bai, X.; and Yao, C. 2016. 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.

Shi, B.; Wang, X.; Lyu, P.; Yao, C.; and Bai, X. 2016. Robust scene text recognition with automatic rectification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4168–4176.

Van der Maaten, L.; and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11).

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.

Vincent, P.; Larochelle, H.; Bengio, Y.; and Manzagol, P.-A. 2008. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, 1096–1103.

Wang, K.; Babenko, B.; and Belongie, S. 2011. End-to-end scene text recognition. In *2011 International Conference on Computer Vision*, 1457–1464. IEEE.

Wang, T.; Zhu, Y.; Jin, L.; Luo, C.; Chen, X.; Wu, Y.; Wang, Q.; and Cai, M. 2020. Decoupled attention network for text recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 12216–12224.

Wang, W.; Xie, E.; Li, X.; Fan, D.-P.; Song, K.; Liang, D.; Lu, T.; Luo, P.; and Shao, L. 2021. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. *arXiv preprint arXiv:2102.12122*.

Wiskott, L.; and Sejnowski, T. J. 2002. Slow feature analysis: Unsupervised learning of invariances. *Neural computation*, 14(4): 715–770.

Yang, X.; He, D.; Zhou, Z.; Kifer, D.; and Giles, C. L. 2017. Learning to Read Irregular Text with Attention Mechanisms. In *IJCAI*, volume 1, 3.

Yu, D.; Li, X.; Zhang, C.; Liu, T.; Han, J.; Liu, J.; and Ding, E. 2020. Towards accurate scene text recognition with semantic reasoning networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12113–12122.

Yu, T.; Yuan, J.; Fang, C.; and Jin, H. 2018. Product Quantization Network for Fast Image Retrieval. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Yue, X.; Kuang, Z.; Lin, C.; Sun, H.; and Zhang, W. 2020. RobustScanner: Dynamically Enhancing Positional Clues for Robust Text Recognition. In *European Conference on Computer Vision*.

Zhang, Q.; and Yang, Y. 2021. ResT: An Efficient Transformer for Visual Recognition. *arXiv preprint arXiv:2105.13677*.