The Thirty-Eighth AAAI Conference on Artificial Intelligence (AAAI-24)

Self-Distillation Regularized Connectionist Temporal Classification Loss for Text Recognition: A Simple Yet Effective Approach

Ziyin Zhang*, Ning Lu*, Minghui Liao, Yongshuai Huang, Cheng Li, Min Wang, Wei Peng

Huawei Technologies Co., Ltd., Shenzhen, China
{zhangziyin1, luning12, liaominghui1, huangyongshuai1, licheng81, wangmin5, peng.wei1}@huawei.com

Abstract

Text recognition methods are gaining rapid development. Some advanced techniques, e.g., powerful modules, language models, and un- and semi-supervised learning schemes, consecutively push the performance on public benchmarks forward. However, the problem of how to better optimize a text recognition model from the perspective of loss functions is largely overlooked. CTC-based methods, widely used in practice due to their good balance between performance and inference speed, still grapple with accuracy degradation. This is because CTC loss emphasizes the optimization of the entire sequence target while neglecting to learn individual characters. We propose a self-distillation scheme for CTC-based model to address this issue. It incorporates a frame-wise regularization term in CTC loss to emphasize individual supervision, and leverages the maximizing-a-posteriori of latent alignment to solve the inconsistency problem that arises in distillation between CTC-based models. We refer to the regularized CTC loss as **D**istillation **C**onnectionist **T**emporal **C**lassification (DCTC) loss. DCTC loss is module-free, requiring no extra parameters, longer inference lag, or additional training data or phases. Extensive experiments on public benchmarks demonstrate that DCTC can boost text recognition model accuracy by up to 2.6%, without any of these drawbacks.

Introduction

Text Recognition (TR) is an indispensable technology that facilitates intelligent auto-driving (Zhu et al. 2018), revealing precise semantic information (Chen et al. 2021b) for sensitive information auditing, saving labor forces for financial processes, etc. Methods for scene text recognition (STR) are blooming at a breathless pace these years. For example, (Du et al. 2022; Da, Wang, and Yao 2022; Lu et al. 2021) focus on designing sophisticated architectures by inventing powerful modules; (Wang et al. 2022a,b) integrate a language model into a text recognition model to enable explicit language modeling; (Patel, Allebach, and Qiu 2023; Yang et al. 2022) learn better sequential features with an un-supervised or semi-supervised learning scheme by leveraging a large amount of label-free or partial labeled data; However, the problem that how to better optimize a text recognition model from a perspective of loss functions is out of in the cold. It is also worth lots of effort since the dedicated designed loss function may be free of extra parameters, extra inference latency, extra training data, or extra training phases.

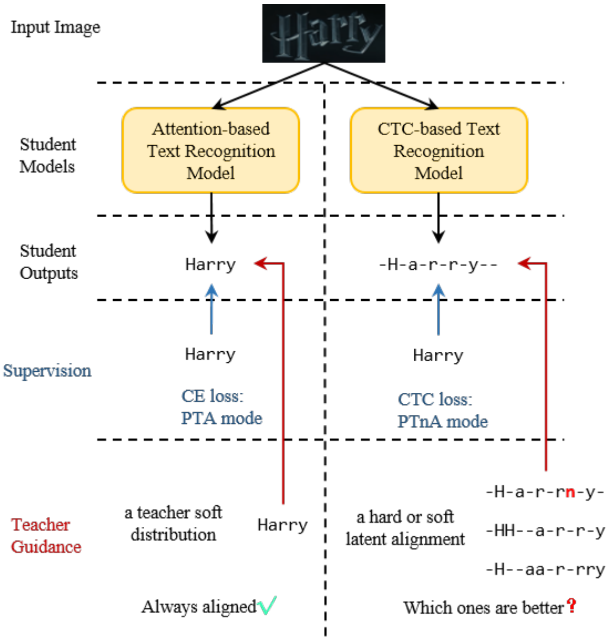

Figure 1: An illustraion of optimization and distillation on CTC- and attention-based models. Also shows the alignment inconsistency problem

Recent text recognition methods are often supervised by two loss functions, the Connectionist Temporal Classification (CTC) loss and the Cross-Entropy (CE) loss, which correspond to CTC-based and attention-based models, respectively. As illustrated in Fig. 1, CTC loss and CE loss optimize models in a Prediction-Target non-Aligned (PTnA) and Prediction-Target Aligned (PTA) mode respectively. Although much recent research empirically shows that CE-based models, which run in the PTA mode, can outperform CTC-based models (Cong et al. 2019; Shi et al. 2016; Baek et al. 2019), CTC-based models have three non-negligible advantages: 1) The CTC decoder is more robust to varying input sequence lengths than the attention-based de-

*These authors contributed equally.

coder (Cong et al. 2019; Chen et al. 2021a); 2) Compared to attention-based models, CTC-based models, getting rid of auto-regression, decode each time-step simultaneously, which can achieve better inference efficiency (Long, He, and Yao 2021; Chen et al. 2021a); 3) Due to its concise model design (Li et al. 2022; Kuang et al. 2021), CRNN (Shi, Bai, and Yao 2017), the classical CTC-based TR model, is still a mainstream industrial model. These practical advantages attract our research focus back to the CTC loss function, motivating this paper.

CTC loss models the negative log total probability of all feasible paths that can be collapsed into the label sequence. However, some of the paths are more plausible. These paths are certain particular alignments of all positions along the sequence. Once discovered and additionally trained with such alignments, the model should be benefited from that. The process of discovering and training those more plausible alignments is known as Knowledge Distillation (KD) (Hinton, Vinyals, and Dean 2015). The more plausible alignments are also a form of "dark knowledge" (Hinton, Vinyals, and Dean 2015) in the context of KD. Nevertheless, a common issue when applying KD to CTC-based models is "alignment inconsistency" (Ding, Chen, and Huo 2020). This issue occurs when the features or outputs of the teacher model are found to be inaccurate or inconsistent. This inconsistency can arise due to the limitations of the teacher alignment, which cannot guarantee full correctness or consistency during training or across multiple teacher models. As a result, this can negatively impact the performance of the distillation process.

The key to success in distillation on CTC models is to find the proper alignments, i.e., the latent alignments. Previous works (Ding, Chen, and Huo 2020; Huang et al. 2018) are module-dependent. They estimate the latent alignment directly from other teacher models' outputs or intermediate features. To obtain more accurate latent alignments, these methods often require complex and well-trained teacher models. To further stabilize the estimate, some (Kim and Rush 2016; Ding, Chen, and Huo 2019) use specifically designed heuristic mechanism to adjust the original one or use an ensemble of a group of raw estimates. Some methods (Kurata and Audhkhasi 2018; Ding, Chen, and Huo 2019) use an ensemble of teachers to improve guidance accuracy. However, 1)they used extra complex teacher models, which increases computing resource demand; 2) they can hardly relieve the intrinsic inaccuracy as a result of directly taking the outputs of the teacher models as the latent alignment; and 3) they incurred distillation instability when using ensembles of teachers because of inconsistent peak positions (Kurata and Audhkhasi 2018), causing unstable collapsed latent alignments.

We propose **D**istillation **CTC** loss (DCTC), a frame-wise, self-distillation scheme for the CTC-based models. By modeling latent alignment distribution as maximizing the posterior probability given the ground truth and model outputs, we derive a simple, effective, and module-free method to generate high-quality estimated latent alignment at each training iteration. This method is closed-formed and does not require any additional module to perform. In summary, our contributions are as follows:

1. We propose a self-distillation scheme, DCTC, to conduct frame-wise regularization for CTC-based models. It can directly apply to existing CTC-based text recognition model without introducing extra teacher models, training phases, or training data.
2. To our knowledge, it is the first work that uses MAP to perform latent alignment estimate. Our method well addresses the alignment inconsistency problem by generating high-quality estimated latent alignment most of the training time, which is supported by our quantitative analysis.
3. Exhaustive experiments over models and CTC loss variants demonstrate that our proposed DCTC loss effectively boost the performance of various text recognition models on both English and Chinese text recognition benchmarks.

Related Works

Text Recognition

Text Recognition is vital in the Optical Character Recognition (OCR) area. In the deep-learning era, how to design powerful modules attracts lots of interest. Shi et al. (Shi, Bai, and Yao 2017) proposed a segmentation-free method, CRNN, which models sequential relationships between frames and employs CTC loss (Graves et al. 2006), adaptively aligning features to targets to train a neural network. This method gained huge success and opened a new era for STR. (Du et al. 2022) used ViT (Dosovitskiy et al. 2021) to develop a single powerful visual model for recognition. It also employs CTC loss to align targets. (Lu et al. 2021; Fang et al. 2021; Yu et al. 2020; Bhunia et al. 2021b) formulated text recognition problem as a translation task that translates a cropped image into a string, using an encoder-decoder framework, along with an attention mechanism (Baek et al. 2019). Recently, thanks to Self-Attention (Vaswani et al. 2017), (Lu et al. 2021; Li et al. 2021) proposed transformer-based STR models to solve the attention drift problem (Cheng et al. 2017). Besides, Liao et al. (Liao et al. 2019) proposed to segment and recognize text from two-dimensional perspective. Another active direction is to lay their hope in a language model. (Qiao et al. 2020) claimed that the encoder-decoder framework only focuses on the local visual feature while ignoring global semantic information. So they used a pre-trained language model to guide the decoding process to improve the model's performance. (Fang et al. 2021) integrated a language model into a vision-based recognition model to enhance its feature representative ability, which iteratively refines the model's prediction. A more advanced work by Bautista et al. (Bautista and Atienza 2022) used permutation language modeling to refine recognition results. To leverage large unlabelled data, (Aberdam et al. 2021) proposed a contrastive pre-training learning scheme to boost performance. Recently, (Guan et al. 2022a,b; Yang et al. 2022) used self-supervision framework to refine visual and language features at a fine-granularity level to improve recognition accuracy.

CTC-Related Text Recognition Methods

Many endeavors have been devoted to improving CTC-based text recognition models. (Feng, Yao, and Zhang 2019) proposed FocalCTC to aim at the imbalance problem of Chinese words, introducing loss (Lin et al. 2020) into CTC loss to modulate the importance of hard and easy word examples. Naturally, CTC loss is not designed for 2D spatial prediction. Xie et al. (Xie et al. 2019) proposed an easy-to-apply aggregated cross entropy (ACE) loss to better solve 2D prediction problems with fast and lightweight implementation. (Wan et al. 2019) extended vanilla CTC as 2D-CTC to adapt to 2D text images by modeling a spatial decoding strategy. To encourage cohesive features, Center loss (Wen et al. 2016) is introduced to CTC loss as Center-CTC loss (Du et al. 2021). (Gao, Zhang, and Liu 2021) provided an expectation-maximum view of CTC loss and a novel voting algorithm to improve decoding performance. Based on maximum entropy regularization (Jaynes 1957), (Liu, Jin, and Zhang 2018) proposed EnCTC to address peaky distribution problem (Graves et al. 2006). VarCTC (Chao, Chen, and Chu 2020) is also proposed to relieve the problem. Tanaka et al. (Tanaka, Ono, and Furuhata 2019) used the framework of virtual adversarial training (Miyato et al. 2017) to develop a fast regularization algorithm FDS on CTC loss by smoothing posterior distributions around training data points.

Knowledge Distillation on Text Recognition or CTC-Based Models

There are a lot of works attempted to apply KD on TR models or CTC-based models. (Bhunia et al. 2021a) creatively employed a knowledge distillation loss to train a unified model for scene and handwritten TR tasks. However, this method needs two additional teacher models, leading to a complicated training procedure. (Takashima, Li, and Kawai 2018) investigated frame- and sequence-level KD on CTC-based acoustic models. (Kim and Rush 2016) proposed a word-level and a sequence-level distillation method and apply them to neural machine translation task. They used beam search to generate hypotheses from output probabilities and kept a K-best list to approximate the teacher distribution. (Ding, Chen, and Huo 2019) used N-best hypotheses imitation to do frame- and segment-wise distillation from a complex teacher model. (Kurata and Audhkhasi 2018) proposed an alignment-consistent ensemble technique to relieve unstable ensemble alignment problem. (Moriya et al. 2020) uses self-distillation KD on the CTC-based ASR system by using a Transformer (Vaswani et al. 2017) module to generate latent alignment. Recently, CCD (Guan et al. 2022b) used a self-distillation module to perform character-level distillation. SIGA (Guan et al. 2022a) used a self-supervised implicit glyph attention module to relief the alignment-drifted issue, which can be also seen as an character-level self-distillation. The aforementioned methods are all module-dependent and need extra teacher models to provide accurate estimated latent alignment. Also, they can hardly give a closed form for the estimated latent alignments.

Methods

A key problem in CTC distillation is alignment inconsistency (Kurata and Audhkhasi 2018). The problem can be described as to find a proper latent alignment $\mathbf{z} \in {V'}^T$, from which the student model can distill and whose length is equal to that of the logits sequence $\mathbf{U} \in \mathbb{R}^{K+1,T}$. V is the character vocabulary, $V' = V \cup \{\text{blank}\}$ is the augmented vocabulary in the CTC setting, and there are $|V| = K$, $|V'| = K + 1$. L is the length of the label sequence, T is the number of time steps, or the length of the logit sequence. It is required that $T > L$ in the CTC setting, causing non-unique alignment, which is the source of alignment inconsistency. We need a way to estimate proper alignments for $\mathbf{U}$ to perform frame-wise KD, which motivates our work.

The Distillation Loss Term in CTC Scenario

Given a sequence of the logits sequence $\mathbf{U} \in \mathbb{R}^{K+1,T}$, the output probability sequence $\mathbf{P} = \mathrm{Softmax}_{V'}(\mathbf{U})$, the ground truth label sequence $\mathbf{y} \in V^L$, and the latent alignment $\mathbf{z} \in {V'}^T$. The true label sequence $\mathbf{y}$ can be regarded as an oracle teacher from which we want the student logits sequence to distill. Originally it was impossible because the lengths of the true label sequence and the logits sequence are different ($L < T$). However, bridging by $\mathbf{z}$ as an agent, the distillation loss term can be formulated as:

$$\mathcal{L}_{\text{distill}}(\mathbf{P}, \mathbf{z}) = \mathcal{L}_{\text{CE}}(\mathbf{P}, \mathbf{z}) = -\sum_{t=1}^{T} z_t \log \mathbf{P}(z_t, t) \quad (1)$$

Distillation CTC is defined as follows:

$$\mathcal{L}_{\text{DCTC}}(\mathbf{U}, \mathbf{P}, \mathbf{y}, \mathbf{z}) = \mathcal{L}_{\text{CTC}}(\mathbf{U}, \mathbf{y}) + \lambda \mathcal{L}_{\text{distill}}(\mathbf{P}, \mathbf{z}) \quad (2)$$

where λ is the coefficient controlling the amplitude of distillation.

The question is how to give a proper latent alignment $\mathbf{z}$. In the self-distillation scheme, it can be generated by a layer or an additional MLP head of the student model. Nevertheless, module-dependent methods often yield bad generation quality, which will be shown in experiments. Aiming to solve this problem, instead of using an module-dependent method, we deduce a closed-form estimation of $\mathbf{z}$ via Maximum-A-Posteriori (MAP).

Estimation of Latent Alignment

For every t from 1 to T, we want to find a certain value for z_t, which can most likely be decoded into the given true label sequence $\mathbf{y}$. Denote the best estimation of $\mathbf{z}$ as $\mathbf{z}^*$, then $\mathbf{z}^*$ is given by

$$\mathbf{z}^* = \arg\min_{V'} \frac{\mathbf{G}}{\mathbf{P}} \quad (3)$$

where $\mathbf{G}$ is the gradient tensor of CTC loss with respect to the logits sequence, that is:

$$\mathbf{G} = \frac{\partial \mathcal{L}_{\text{CTC}}(\mathbf{U}, \mathbf{y})}{\partial \mathbf{U}} \quad (4)$$

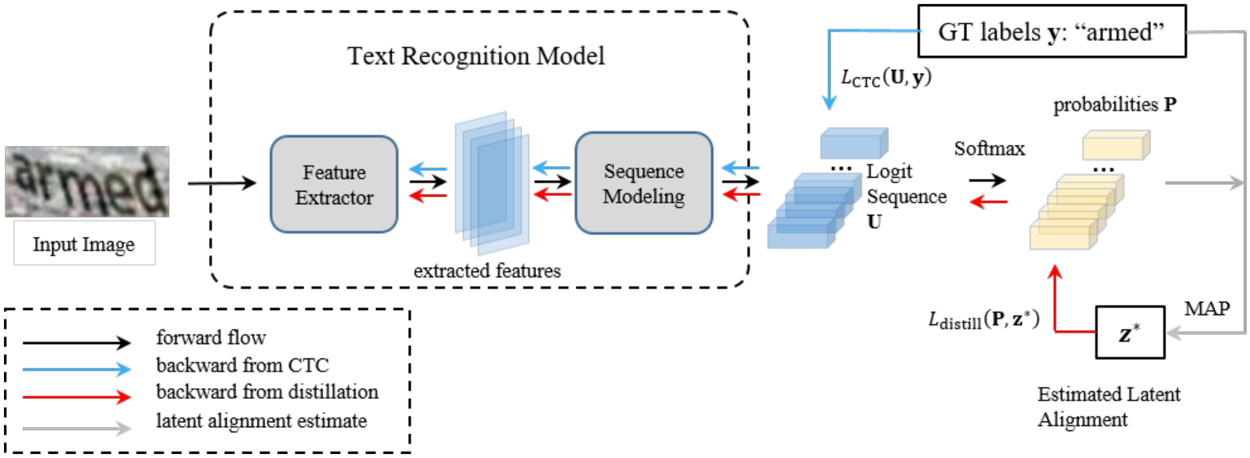

Figure 2: The Architecture of DCTC in Self-distillation Scheme

Derivation of our generation method Given input (image) $\mathbf{X}$ and its corresponding true label sequence $\mathbf{y}$, at time t, the certain z_t that is most likely to decoded into $\mathbf{y}$ can be formulated as a MAP estimate:

$$\begin{aligned} z_t^* &= \arg\max_{z_t \in V'} p(\mathbf{y}|z_t, \mathbf{X}) \\ &= \arg\max_{z_t \in V'} \frac{p(z_t|\mathbf{y},\mathbf{X})p(\mathbf{y}|\mathbf{X})}{p(z_t|\mathbf{X})} \\ &= \arg\max_{z_t \in V'} \frac{p(z_t|\mathbf{y},\mathbf{X})}{p(z_t|\mathbf{X})} \end{aligned} \tag{5}$$

In Eq. (5), $p(z_t|\mathbf{X})$ is the probability that is directly output by the model. $p(z_t|\mathbf{X},\mathbf{y})$ is the probability that character $z_t \in V'$ appears at time step t when $\mathbf{y}$ and $\mathbf{X}$ are given. We now model $p(z_t|\mathbf{X},\mathbf{y})$ in the CTC setting. Using (Graves et al. 2006)'s notation, let $\alpha(\cdot,\cdot)$ and $\beta(\cdot,\cdot)$ be the forward and the backward table respectively. $\alpha(\cdot,\cdot),\ \beta(\cdot,\cdot) \in \mathbb{R}^{l',T}$, where $l' = 2L+1$ is the length of the augmented true label sequence $\mathbf{y}' \in V'^{l'}$ used in computing α and β. Forward table element $\alpha(i,t)$ means the probability that the cumulative paths go through $\mathbf{y}'_i \in V'$ at time step t from the start of $\mathbf{y}'$. Backward table element $\beta(i,t)$ means the probability that the cumulative paths go through $\mathbf{y}'_i$ at time step t from the end of $\mathbf{y}'$. As such, $\alpha(i,t)\beta(i,t)/\mathbf{P}(\mathbf{y}'_i,t)$ is the probability that the total paths go through $\mathbf{y}'_i$ at time step t over the whole time sequence. Denote $S(i,t) = \alpha(i,t)\beta(i,t)/\mathbf{P}(\mathbf{y}'_i,t)$ for simplicity. Then, for a specific class c, in the CTC setting, we have:

$$p(z_t = c|\mathbf{X},\mathbf{y}) \propto \sum_{i,\mathbf{y}'_i=c}^{l'} S(i,t) \tag{6}$$

We need a way to connect Eq. (6) to a value that we can easily compute. Observe that the gradients of CTC loss with respect to $\mathbf{U}$ is given by (Graves et al. 2006):

$$\frac{\partial \mathcal{L}_{\text{CTC}}}{\partial \mathbf{U}(c,t)} = \mathbf{G}(c,t) = \mathbf{P}(c,t) - \frac{\sum_{i,\mathbf{y}'_i=c}^{l'} S(i,t)}{p(\mathbf{y}|\mathbf{X})} \tag{7}$$

So we have:

$$\begin{aligned} p(z_t = c|\mathbf{X},\mathbf{y}) &\propto (\mathbf{P}(c,t) - \mathbf{G}(c,t))p(\mathbf{y}|\mathbf{X}) \\ &\propto \mathbf{P}(c,t) - \mathbf{G}(c,t) \end{aligned} \tag{8}$$

Note the fact that $\mathbf{P}(c,t) = p(z_t = c|\mathbf{X})$. Now, comparing Eq. (5) and Eq. (8), Eq. (5) becomes:

$$\begin{aligned} z_t^* &= \arg\max_{c \in V'} \frac{\mathbf{P}(c,t) - \mathbf{G}(c,t)}{\mathbf{P}(c,t)} \\ &= \arg\max_{c \in V'} \left(1 - \frac{\mathbf{G}(c,t)}{\mathbf{P}(c,t)}\right) \\ &= \arg\min_{c \in V'} \frac{\mathbf{G}(c,t)}{\mathbf{P}(c,t)} \end{aligned} \tag{9}$$

The ultimate form Eq. (3) is simply the vectorized version of Eq. (9), which is easy to implement. **We use Eq. (3) to generate latent alignment in practice.**

There might be a concern that Eq. (3) seems to have singularities when $\mathbf{P}$ has zeros, which impedes the calculation of $\mathbf{z}^*$. However, it is not the case. There is NO singularity at all. We can prove that $\mathbf{G}(c,t)/\mathbf{P}(c,t)$ is bounded within $[0,1]$ along $\mathbf{P}(c,t)$ changing from 0 to 1. The proof, however, is cumbersome. Readers who are interested in it can refer to supplementary materials.

Our proposed estimation method can generate incredibly high-quality latent alignment. We empirically show that in experiments.

Summary of DCTC Loss

DCTC loss works in a self-distillation scheme, as such, $\mathbf{z}^*$ is directly estimated from the CTC loss that supervises the student model. No other teacher models participated. So, we substitute Eq. (3) into Eq. (2) and get:

$$\mathcal{L}_{\text{DCTC}}(\mathbf{U},\mathbf{P},\mathbf{y},\mathbf{z}^*) = \mathcal{L}_{\text{CTC}}(\mathbf{U},\mathbf{y}) + \lambda\mathcal{L}_{\text{distill}}(\mathbf{P},\mathbf{z}^*) \tag{10}$$

We show a pseudo code of DCTC loss in Algorithm 1 for a clear understanding. Meanwhile, the architecture of our method is shown in Fig. 2.

Algorithm 1: Calculation of DCTC loss in self-distillation scheme

```
Input: the input logits $\mathbf{U}$, ground truth label sequence $\mathbf{y}$, weighting factor $\lambda$
Calculate probabilities $\mathbf{P} = \mathrm{softmax}_{V'}\mathbf{U}$.
Calculate CTC loss $\mathcal{L}_1 = \mathcal{L}_{\mathrm{CTC}}(\mathbf{U}, \mathbf{y})$.
Without tracing gradients, copy $\mathbf{U}$ as $\mathbf{U}'$
Calculate CTC loss $\mathcal{L}_2 = \mathcal{L}_{\mathrm{CTC}}(\mathbf{U}', \mathbf{y})$.
Calculate gradients $\mathbf{G} = \partial\mathcal{L}_2/\partial\mathbf{U}'$
Take argmin over vocabulary: $\mathbf{z}^* = \arg\min_{V'} \mathbf{G}/\mathbf{P}$
Compute $\mathcal{L}_{\mathrm{DCTC}} = \mathcal{L}_1 + \lambda\mathcal{L}_{\mathrm{distill}}(\mathbf{P}, \mathbf{z}^*)$
Output: $\mathcal{L}_{\mathrm{DCTC}}$
```

Experiments

Datasets

All datasets used in our experiments are publicly available. Our experiments are conducted on English and Chinese scenarios. For English text recognition task, we train all models on two commonly used synthetic scene text recognition datasets: ST (Gupta, Vedaldi, and Zisserman 2016) and MJ (Jaderberg et al. 2014). We evaluate all models on six English benchmark datasets: IC13 (Karatzas et al. 2013), IC15 (Karatzas et al. 2015), SVT (Wang, Babenko, and Belongie 2011), SVTP (Phan et al. 2013), IIIT (Mishra, Karteek, and Jawahar 2012) and CT (Risnumawan et al. 2014). Each of these six contains 857, 647, 3000, 1811, 645 and 288 test samples, respectively. For Chinese text recognition task, we use the Chinese Benchmark datasets (Chen et al. 2021a). It contains four subsets: Scene, Web, Document(Doc) and Handwritten(Hand). Each of these four contains 63646, 14059, 50000 and 18651 test samples, respectively. We train all models on their own training set and evaluate them on their own test set. The license of academically using Handwritten subset, aka SCUT-HCCDoc (Zhang, Liang, and Jin 2020), has been issued by its owner as per our request.

Implementation Details

Base Models We choose six base models as the student models. They are CRNN (Shi, Bai, and Yao 2017), TRBA (Baek, Matsui, and Aizawa 2021), SVTR-T, SVTR-S and SVTR-B (Du et al. 2022). We use them trained with CTC loss as the baseline models and compare them to models trained with DCTC loss in a self-distillation scheme (directly replacing CTC loss with DCTC loss), meaning the teacher is the student itself. All models are implemented with PaddleOCR[1]. We implemented DCTC as a CUDA-CPP extension for computation efficiency.

Hyperparameters Hyperparameters for the number of training epochs, batch size, data augmentation strategy, optimizer, learning rate, and decay policy are different as per the base models and follow the base models' own original settings described in their source. The image size used for English task is (h,w)=(32,100), and for Chinese task is (32,256). The distillation coefficient λ in $\mathcal{L}_{\mathrm{DCTC}}$ is set to 0.025 for English tasks, and 0.01 for Chinese tasks. All experiments are conducted on Nvidia Tesla V100 GPUs.

Metrics and Evaluation Protocols We use accuracy to evaluate all models' performance. **Accuracy (ACC)** is the ratio of the number of totally correct predictions over the number of test samples. Certain protocols applied when evaluating. For English tasks, only numbers and letters (case-insensitive) are evaluated. For Chinese tasks, we follow (Chen et al. 2021a)'s conventions: 1) convert full-width characters to half-width characters; 2) convert traditional Chinese characters to simplified Chinese characters; 3) all letters to lowercase, and 4) discard all spaces.

In addition, we propose a new metric "**Alignment Accuracy (AACC)**" to measure the quality of the latent alignment estimate. It is defined as the ACC of the **decoded** latent alignments and the ground truth labels. The difference from evaluating model performance is that we do not apply any protocol when evaluating AACC. We decode latent alignments in a CTC-greedy way, meaning collapsing repeating characters and removing all blanks.

A Model-Wise Comparison

We compare DCTC loss with CTC loss on six models mentioned and collect all results in Tab. 1. Each model is compared to its baseline, which is the one trained by CTC loss. We can clearly see that all models achieve accuracy improvement over almost all benchmark datasets, which profoundly verifies the effectiveness of our method at the model level. CRNN, the most classical, representative, and widely-used industrial CTC-based text recognition model, obtains a 2.6% average accuracy increment on English and 2.1% on Chinese benchmarks. The advanced CTC-based single-visual-model text recognition method, SVTR series, can also gain accuracy improvement by our method. Up to 0.9% and 1.1% average accuracy improvement in English and Chinese are observed when trained with DCTC loss. Besides, as our method does not change the structure of the models, the inference speed remains the same.

A Loss-Wise Comparison

Our method can be regarded as a variant of CTC loss when working in a self-distillation scheme. Many variants of CTC have been proposed but have yet to be experimented with advanced models or on Chinese benchmarks. In this part, we compare our method with other variants of CTC loss. The chosen variants are FocalCTC[2] (Feng, Yao, and Zhang 2019) and EnCTC[3] (Liu, Jin, and Zhang 2018) We choose them because they 1) have been peer-reviewed and 2) have public code bases (in footnotes).

We align the hyperparameters with their original settings to make the comparison fair. For FocalCTC loss, $\alpha = 1$ and $\gamma = 2$; for EnCTC loss, the regularization coefficient $\beta =$

[1] https://github.com/PaddlePaddle/PaddleOCR

[2] https://github.com/PaddlePaddle/PaddleOCR

[3] https://github.com/liuhu-bigeye/enctc.crnn

Base Model	Methods	English Benchmarks							Chinese Benchmarks				
		IC13	SVT	IIIT	IC15	SVTP	CT	Avg	Scene	Web	Doc	Hand	Avg
CRNN	CTC	90.3	78.9	84.3	65.9	64.8	61.3	77.3	54.9	56.2	97.5	48.0	68.7
	DCTC	**90.7**	**82.4**	**88.9**	**66.1**	**65.4**	**68.1**	**79.9**	**58.6**	**57.0**	**98.0**	**49.7**	**70.8**
TRBA	CTC	94.0*	88.9*	93.6*	76.5*	79.8*	84.0*	87.3	59.6*	57.8*	98.2*	48.9*	71.3
	DCTC	**94.2**	**90.4**	**93.9**	**78.1**	**81.3**	**85.8**	**88.2**	**61.1**	**58.6**	**99.2**	**49.5**	**72.4**
SVTR-T	CTC	96.3	91.6	94.4	84.1	85.4	88.2	90.8	67.9	61.8*	99.1*	47.2*	75.3
	DCTC	**96.4**	**92.3**	**95.4**	**85.3**	**86.1**	**89.9**	**91.7**	**68.3**	**63.9**	**99.2**	**48.1**	**75.9**
SVTR-S	CTC	95.7	**93.0**	95.0	84.7	87.9	92.0	91.6	69.0	63.9*	99.2*	49.5*	76.3
	DCTC	**96.4**	92.5	**96.2**	**86.2**	**88.1**	**92.4**	**92.5**	**70.3**	**65.8**	**99.4**	**50.3**	**77.3**
SVTR-B	CTC	97.1	91.5	96.0	85.2	**89.9**	91.7	92.3	71.4	64.1*	99.3*	50.0*	77.5
	DCTC	**97.1**	**92.9**	**96.3**	**87.2**	89.6	**92.1**	**93.1**	**72.2**	**67.0**	**99.4**	**50.4**	**78.2**
SVTR-L	CTC	97.2	91.7	96.3	86.6	88.4	**95.1**	92.8	72.1	66.3*	99.3*	50.3*	78.1
	DCTC	**97.4**	**93.7**	**96.9**	**87.3**	**88.5**	92.3	**93.3**	**73.9**	**68.5**	**99.4**	**51.0**	**79.2**

Table 1: Results of Model-wise Comparison. Bold ACCs are the model-wise better results. ACC marked by * means those data are not reported and thus reproduced by us. Results on English benchmarks of the baseline models of CRNN and TRBA are reported by (Baek, Matsui, and Aizawa 2021). Results on Chinese benchmarks of the baseline model of CRNN are reported by (Chen et al. 2021a). Results of the baseline model of SVTR series are reported by (Du et al. 2022).

Base Model	Variants	English Benchmarks							Chinese Benchmarks				
		IC13	SVT	IIIT	IC15	SVTP	CT	Avg	Scene	Web	Doc	Hand	Avg
CRNN	CTC	90.3	78.9	84.3	65.9	64.8	61.3	77.3	54.9	56.2	97.5	48.0	68.7
	FocalCTC	89.6	80.1	81.2	65.2	63.0	60.2	75.6	54.8	56.0	97.5	48.3	68.7
	EnCTC	90.1	81.5	85.6	64.7	62.9	59.0	77.1	49.0	50.7	97.5	36.6	64.2
	DCTC	**90.7**	**82.4**	**88.9**	**66.1**	**65.4**	**68.1**	**79.9**	**58.6**	**57.0**	**98.0**	**49.7**	**70.8**
SVTR-T	CTC	96.3	91.6	94.4	84.1	85.4	88.2	90.8	67.9	61.8*	99.1*	47.2*	75.3
	FocalCTC	96.0	91.0	94.3	84.1	85.1	87.9	90.6	67.1	60.2	99.2	46.5	74.8
	EnCTC	94.9	90.8	94.5	84.3	85.4	88.2	90.6	65.9	63.7	97.9	47.1	74.2
	DCTC	**96.4**	**92.3**	**95.4**	**85.3**	**86.1**	**89.9**	**91.7**	**68.3**	**63.9**	**99.2**	**48.1**	**75.9**

Table 2: Results of Loss-wise Comparison. ACC marked by * means those data are not reported and thus reproduced by us; Results of CTC loss and DCTC are the same as in Tab. 1; Results of FocalCTC and EnCTC are all reproduced by us.

0.2. The experiment results are collected in Tab. 2. We use CRNN and SVTR-T as base models for efficiency. We can see that our method consistently achieves improvements on all benchmarks, further proving our method's effectiveness.

Comparison of Latent Alignment Estimate

Much previous research on distillation for CTC-based models has been working on finding reasonable estimates of the latent alignment. They used various means to directly to utilize $p(\mathbf{z}|\mathbf{X})$ to estimate the latent alignment. The most naive utilization way is to take the hard prediction of $p(\mathbf{z}|\mathbf{X})$, i.e., $\arg\max_{V'} \mathbf{P}$. In this section, we compare our estimate method with two other sources of estimate: one is to take $\arg\max_{V'} \mathbf{P}$ directly from the model itself, denoted as "Self". Another is to take $\arg\max_{V'} \mathbf{P}$ from a three-layer Transformer encoder branch additionally added to the model, denoted as "Teacher". This branch is trained with a CTC loss during the training process. We use CRNN and SVTR-T as the experiment models. We record AACC in training on English, Chinese Scene and, Chinese Hand tasks under different estimate methods, respectively. AACCs are computed by the average over ten consecutive batches at certain progress points in training. Results of AACC are visually shown in Fig. 3. We also record the model accuracy under different estimate methods and collect the results in Tab. 3.

We can see from Fig. 3 that our method (DCTC) can yield high-quality latent alignment (High AACC) even at the beginning of training. "Teacher" takes second place, while "Self" only generates moderate estimates after a period of training. Besides, our estimate method gets quickly saturated to nearly 100% AACC. In contrast, "Teacher" and "Self" ling for a long time at a low-to-middle level AACC and can hardly get close to 100%, not to mention that "Teacher" used an extra Transformer encoder. Tab. 3 shows that both "Teacher" and "Self" cause accuracy degradation on almost all benchmarks, suggesting that simply taking the output of distilled module as the latent alignments is harmful in the CTC setting, no matter whether using a teacher. The only exception is that "Teacher" boosts CRNN on English benchmarks. We explain that the added Transformer branch fundamentally increases the model capability of CRNN, overcoming the harm from the low-quality estimate of the "Teacher" method. In conclusion, our method can draw high-quality

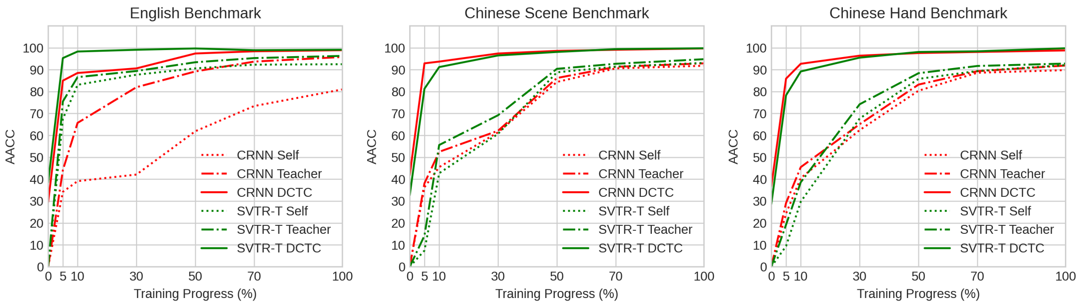

Figure 3: Curves of AACC of Estimated Latent Alignment

Base Model	Estimate Method	Extra Module	English Benchmarks							Chinese Benchmarks				
			IC13	SVT	IIIT	IC15	SVTP	CT	Avg	Scene	Web	Doc	Hand	Avg
CRNN	-	-	90.3	78.9	84.3	65.9	64.8	61.3	77.3	54.9	56.2	97.5	48.0	68.7
	Self	N	88.6	75.1	82.4	63.8	60.5	59.4	75.0	46.1	50.7	92.4	40.2	61.6
	Teacher	Y	90.2	81.0	88.8	64.5	63.7	65.3	79.0	48.9	52.3	94.9	42.1	64.1
	DCTC	N	**90.7**	**82.4**	**88.9**	**66.1**	**65.4**	**68.1**	**79.9**	**58.6**	**57.0**	**98.0**	**49.7**	**70.8**
SVTR-T	-	-	96.3	91.6	94.4	84.1	85.4	88.2	90.8	67.9	61.8*	99.1*	47.2*	75.3
	Self	N	95.0	90.3	93.2	84.8	85.1	86.5	90.1	67.3	60.0	99.1	46.6	74.8
	Teacher	Y	95.9	91.1	94.0	83.4	85.7	86.4	90.3	66.4	61.1	99.0	46.5	74.5
	DCTC	N	**96.4**	**92.3**	**95.4**	**85.3**	**86.1**	**89.9**	**91.7**	**68.3**	**63.9**	**99.2**	**48.1**	**75.9**

Table 3: Results of latent alignment method estimate method comparison. ACC marked by * means those data are not reported and thus reproduced by us.

distillation dark knowledge during the whole training time. This phenomenon explains why DCTC loss can still benefit the student model even under a self-distillation scheme, where no extra teacher model participates.

Visual Show of the Effectiveness of Distillation Supervision

Eq. (1) suggests that DCTC adds frame-wise and character-level supervision to original CTC supervision. Unlike CTC, who more emphasizes sequence supervision, this distillation supervision will make the character features more discriminative, which contributes to the overall performance improvement. We select several hard example clusters from test sets and fetch their features from an SVTR-T model trained with DCTC loss. A hard example cluster is a group of characters more prone to be wrongly recognized as each other. We make a feature visualization study with t-SNE (van der Maaten and Hinton 2008). Fig. 4 illustrates two hard example clusters of feature projections. Different characters are marked with different colors. Our method drive the model to extract more discriminative features which are more cohesive than those extracted by the baselines. For more clusters, please refer to the supplementary materials.

Conclusion

In this paper, we base on a self-distillation framework through MAP estimate to formulate DCTC, as a variant of CTC loss. The way we estimate the latent alignments can distill high-quality dark knowledge from the student model itself and well address the alignment inconsistency problem, which is supported by our quantitative analysis. Our proposed DCTC loss is concise yet quite effective. It boasts various text recognition models' performance on both English and Chinese benchmarks. Furthermore, visual analysis shows that DCTC loss can yield more cohesive features, which explains performance improvement. Besides, our method barely incurs additional computational complexity, training data, and training phase.

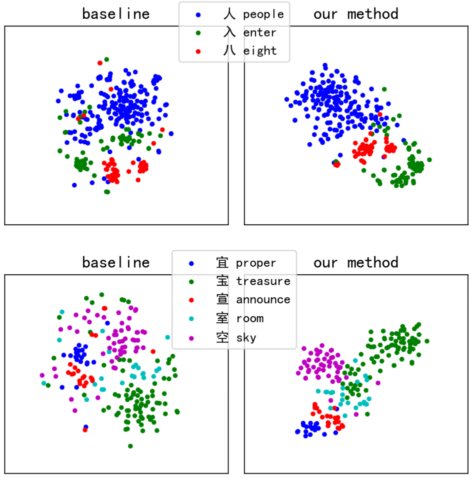

Figure 4: Feature visualization. Each row represents a hard sample cluster

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 *Proc. CVPR*, 15302–15312.

Baek, J.; Kim, G.; Lee, J.; Park, S.; Han, D.; Yun, S.; Oh, S. J.; and Lee, H. 2019. What Is Wrong With Scene Text Recognition Model Comparisons? Dataset and Model Analysis. In *Proc. ICCV*, 4714–4722.

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. CoRR:2103.04400.

Bautista, D.; and Atienza, R. 2022. Scene Text Recognition with Permuted Autoregressive Sequence Models. In *Proc. ECCV*, volume 13688, 178–196.

Bhunia, A. K.; Sain, A.; Chowdhury, P. N.; and Song, Y. 2021a. Text is Text, No Matter What: Unifying Text Recognition using Knowledge Distillation. In *Proc. ICCV*, 963–972.

Bhunia, A. K.; Sain, A.; Kumar, A.; Ghose, S.; Chowdhury, P. N.; and Song, Y. 2021b. Joint Visual Semantic Reasoning: Multi-Stage Decoder for Text Recognition. In *Proc. ICCV*, 14920–14929.

Chao, L.; Chen, J.; and Chu, W. 2020. Variational Connectionist Temporal Classification. In *Proc. ECCV*, volume 12373, 460–476.

Chen, J.; Yu, H.; Ma, J.; Guan, M.; Xu, X.; Wang, X.; Qu, S.; Li, B.; and Xue, X. 2021a. Benchmarking Chinese Text Recognition: Datasets, Baselines, and an Empirical Study. *CoRR*, abs/2112.15093.

Chen, X.; Jin, L.; Zhu, Y.; Luo, C.; and Wang, T. 2021b. Text Recognition in the Wild: A Survey. *ACM Comput. Surv.*, 54(2).

Cheng, Z.; Bai, F.; Xu, Y.; Zheng, G.; Pu, S.; and Zhou, S. 2017. Focusing Attention: Towards Accurate Text Recognition in Natural Images. In *Proc. ICCV*.

Cong, F.; Hu, W.; Huo, Q.; and Guo, L. 2019. A Comparative Study of Attention-Based Encoder-Decoder Approaches to Natural Scene Text Recognition. In *Proc. ICDAR*, 916–921.

Da, C.; Wang, P.; and Yao, C. 2022. Levenshtein OCR. In Avidan, S.; Brostow, G. J.; Cissé, M.; Farinella, G. M.; and Hassner, T., eds., *Proc. ECCV*, volume 13688, 322–338.

Ding, H.; Chen, K.; and Huo, Q. 2019. Compression of CTC-Trained Acoustic Models by Dynamic Frame-Wise Distillation or Segment-Wise N-Best Hypotheses Imitation. In *Interspeech*, 3218–3222.

Ding, H.; Chen, K.; and Huo, Q. 2020. Improving Knowledge Distillation of CTC-Trained Acoustic Models With Alignment-Consistent Ensemble and Target Delay. *IEEE/ACM transactions on audio, speech, and language processing*.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; Uszkoreit, J.; and Houlsby, N. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *ICLR*.

Du, Y.; Chen, Z.; Jia, C.; Yin, X.; Zheng, T.; Li, C.; Du, Y.; and Jiang, Y.-G. 2022. SVTR: Scene Text Recognition with a Single Visual Model. *Proc. IJCAI*.

Du, Y.; Li, C.; Guo, R.; Cui, C.; Liu, W.; Zhou, J.; Lu, B.; Yang, Y.; Liu, Q.; Hu, X.; Yu, D.; and Ma, Y. 2021. PP-OCRv2: Bag of Tricks for Ultra Lightweight OCR System. *CoRR*, abs/2109.03144.

Fang, S.; Xie, H.; Wang, Y.; Mao, Z.; and Zhang, Y. 2021. Read Like Humans: Autonomous, Bidirectional and Iterative Language Modeling for Scene Text Recognition. *Proc. CVPR*, 7094–7103.

Feng, X.; Yao, H.; and Zhang, S. 2019. Focal CTC Loss for Chinese Optical Character Recognition on Unbalanced Datasets. *Complexity*, 2019: 1–11.

Gao, L.; Zhang, H.; and Liu, C. 2021. Regularizing CTC in Expectation-Maximization Framework with Application to Handwritten Text Recognition. In *IJCNN*, 1–7.

Graves, A.; Fernández, S.; Gomez, F.; and Schmidhuber, J. 2006. Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks. In *ICML*, ICML '06, 369–376. New York, NY, USA. ISBN 1595933832.

Guan, T.; Gu, C.; Tu, J.; Yang, X.; Feng, Q.; Zhao, Y.; and Shen, W. 2022a. Self-supervised Implicit Glyph Attention for Text Recognition.

Guan, T.; Shen, W.; Yang, X.; Feng, Q.; and Jiang, Z. 2022b. Self-supervised Character-to-Character Distillation for Text Recognition.

Gupta, A.; Vedaldi, A.; and Zisserman, A. 2016. Synthetic Data for Text Localisation in Natural Images. *Proc. CVPR*, 2315–2324.

Hinton, G. E.; Vinyals, O.; and Dean, J. 2015. Distilling the Knowledge in a Neural Network. *CoRR*.

Huang, M.; You, Y.; Chen, Z.; Qian, Y.; and Yu, K. 2018. Knowledge Distillation for Sequence Model. In *Interspeech*, 3703–3707.

Jaderberg, M.; Simonyan, K.; Vedaldi, A.; and Zisserman, A. 2014. Synthetic Data and Artificial Neural Networks for Natural Scene Text Recognition. *CoRR*, abs/1406.2227.

Jaynes, E. T. 1957. Information Theory and Statistical Mechanics. II. *Physical Review*.

Karatzas, D.; i Bigorda, L. G.; Nicolaou, A.; Ghosh, S. K.; Bagdanov, A. D.; Iwamura, M.; Matas, J.; Neumann, L.; Chandrasekhar, V. R.; Lu, S.; Shafait, F.; Uchida, S.; and Valveny, E. 2015. ICDAR 2015 competition on Robust Reading. *Proc. ICDAR*, 1156–1160.

Karatzas, D.; Shafait, F.; Uchida, S.; Iwamura, M.; i Bigorda, L. G.; Mestre, S. R.; Romeu, J. M.; Mota, D. F.; Almazán, J.; and de las Heras, L.-P. 2013. ICDAR 2013 Robust Reading Competition. *2013 12th International Conference on Document Analysis and Recognition*, 1484–1493.

Kim, Y.; and Rush, A. M. 2016. Sequence-Level Knowledge Distillation. In *EMNLP*, 1317–1327.

Kuang, Z.; Sun, H.; Li, Z.; Yue, X.; Lin, T. H.; Chen, J.; Wei, H.; Zhu, Y.; Gao, T.; Zhang, W.; Chen, K.; Zhang, W.; and Lin, D. 2021. MMOCR: A Comprehensive Toolbox for Text Detection, Recognition and Understanding. In *ACMMM*, 3791–3794.

Kurata, G.; and Audhkhasi, K. 2018. Improved Knowledge Distillation from Bi-Directional to Uni-Directional LSTM CTC for End-to-End Speech Recognition. In *SLT*, 411–417.

Li, C.; Liu, W.; Guo, R.; Yin, X.; Jiang, K.; Du, Y.; Du, Y.; Zhu, L.; Lai, B.; Hu, X.; Yu, D.; and Ma, Y. 2022. PP-OCRv3: More Attempts for the Improvement of Ultra Lightweight OCR System.

Li, M.; Lv, T.; Cui, L.; Lu, Y.; Florencio, D.; Zhang, C.; Li, Z.; and Wei, F. 2021. TrOCR: Transformer-based Optical Character Recognition with Pre-trained Models. *CoRR*.

Liao, M.; Zhang, J.; Wan, Z.; Xie, F.; Liang, J.; Lyu, P.; Yao, C.; and Bai, X. 2019. Scene Text Recognition from Two-Dimensional Perspective. In *Proc. AAAI*, 8714–8721.

Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; and Dollár, P. 2020. Focal Loss for Dense Object Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42: 318–327.

Liu, H.; Jin, S.; and Zhang, C. 2018. Connectionist Temporal Classification with Maximum Entropy Regularization. In *NeurIPS*, 839–849.

Long, S.; He, X.; and Yao, C. 2021. Scene Text Detection and Recognition: The Deep Learning Era. *Int. J. Comput. Vis.*, 129(1): 161–184.

Lu, N.; Yu, W.; Qi, X.; Chen, Y.; Gong, P.; Xiao, R.; and Bai, X. 2021. MASTER: Multi-aspect non-local network for scene text recognition. *Pattern Recognition*, 117: 107980.

Mishra, A.; Karteek, A.; and Jawahar, C. V. 2012. Top-down and bottom-up cues for scene text recognition. *Proc. CVPR*, 2687–2694.

Miyato, T.; ichi Maeda, S.; Koyama, M.; and Ishii, S. 2017. Virtual Adversarial Training: A Regularization Method for Supervised and Semi-Supervised Learning. *CoRR*.

Moriya, T.; Ochiai, T.; Karita, S.; Sato, H.; Tanaka, T.; Ashihara, T.; Masumura, R.; Shinohara, Y.; and Delcroix, M. 2020. Self-Distillation for Improving CTC-Transformer-Based ASR Systems. In *ISCA*, 546–550.

Patel, G.; Allebach, J. P.; and Qiu, Q. 2023. Seq-UPS: Sequential Uncertainty-aware Pseudo-label Selection for Semi-Supervised Text Recognition. In *WACV*, 6169–6179.

Phan, T. Q.; Shivakumara, P.; Tian, S.; and Tan, C. L. 2013. Recognizing Text with Perspective Distortion in Natural Scenes. *Proc. ICCV*, 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 *Proc. CVPR*, 13525–13534.

Risnumawan, A.; Shivakumara, P.; Chan, C. S.; and Tan, C. L. 2014. A robust arbitrary text detection system for natural scene images. *Expert Syst. Appl.*, 41: 8027–8048.

Shi, B.; Bai, X.; and Yao, C. 2017. An End-to-End Trainable Neural Network for Image-Based Sequence Recognition and Its Application to Scene Text Recognition. *TPAMI*, 39: 2298–2304.

Shi, B.; Wang, X.; Lyu, P.; Yao, C.; and Bai, X. 2016. Robust Scene Text Recognition with Automatic Rectification. In *Proc. CVPR*, 4168–4176.

Takashima, R.; Li, S.; and Kawai, H. 2018. An Investigation of a Knowledge Distillation Method for CTC Acoustic Models. In *ICASSP*, 5809–5813.

Tanaka, R.; Ono, S.; and Furuhata, A. 2019. Fast Distributional Smoothing for Regularization in CTC Applied to Text Recognition. In *Proc. ICDAR*, 302–308.

van der Maaten, L.; and Hinton, G. E. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9: 2579–2605.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All you Need. In *NeurIPS*.

Wan, Z.; Xie, F.; Liu, Y.; Bai, X.; and Yao, C. 2019. 2D-CTC for Scene Text Recognition. *CoRR*, abs/1907.09705.

Wang, H.; Liao, J.; Cheng, T.; Gao, Z.; Liu, H.; Ren, B.; Bai, X.; and Liu, W. 2022a. Knowledge Mining with Scene Text for Fine-Grained Recognition. In *Proc. CVPR*, 4614–4623.

Wang, K.; Babenko, B.; and Belongie, S. J. 2011. End-to-end scene text recognition. *Proc. ICCV*, 1457–1464.

Wang, Y.; Xie, H.; Fang, S.; Xing, M.; Wang, J.; Zhu, S.; and Zhang, Y. 2022b. PETR: Rethinking the Capability of Transformer-Based Language Model in Scene Text Recognition. *IEEE Trans. Image Processing*.

Wen, Y.; Zhang, K.; Li, Z.; and Qiao, Y. 2016. A Discriminative Feature Learning Approach for Deep Face Recognition. In *Proc. ECCV*.

Xie, Z.; Huang, Y.; Zhu, Y.; Jin, L.; Liu, Y.; and Xie, L. 2019. Aggregation Cross-Entropy for Sequence Recognition. In *Proc. CVPR*, 6538–6547.

Yang, M.; Liao, M.; Lu, P.; Wang, J.; Zhu, S.; Luo, H.; Tian, Q.; and Bai, X. 2022. Reading and Writing: Discriminative and Generative Modeling for Self-Supervised Text Recognition. 4214–4223.

Yu, D.; Li, X.; Zhang, C.; Han, J.; Liu, J.; and Ding, E. 2020. Towards Accurate Scene Text Recognition with Semantic Reasoning Networks. CoRR:2003.12294.

Zhang, H.; Liang, L.; and Jin, L. 2020. SCUT-HCCDoc: A new benchmark dataset of handwritten Chinese text in unconstrained camera-captured documents. *Pattern Recognit.*, 108: 107559.

Zhu, Y.; Liao, M.; Yang, M.; and Liu, W. 2018. Cascaded Segmentation-Detection Networks for Text-Based Traffic Sign Detection. *IEEE Trans. Intell. Transp. Syst.*, 19(1): 209–219.