Pushing the Performance Limit of Scene Text Recognizer without Human Annotation

Caiyuan Zheng[1,2]*, Hui Li[3], Seon-Min Rhee[4], Seungju Han[4], Jae-Joon Han[4], Peng Wang[1,2]†

[1]School of Computer Science and Ningbo Institute, Northwestern Polytechnical University, China,
[2]National Engineering Laboratory for Integrated Aero-Space-Ground-Ocean
Big Data Application Technology, China,
[3]Samsung R&D Institute China Xi'an (SRCX),
[4]Samsung Advanced Institute of Technology (SAIT), South Korea

2020202704@mail.nwpu.edu.cn, {hui01.li, s.rhee, sj75.han, jae-joon.han}@samsung.com
peng.wang@nwpu.edu.cn

Abstract

Scene text recognition (STR) attracts much attention over the years because of its wide application. Most methods train STR model in a fully supervised manner which requires large amounts of labeled data. Although synthetic data contributes a lot to STR, it suffers from the real-to-synthetic domain gap the restricts model performance. In this work, we aim to boost STR models by leveraging both synthetic data and the numerous real unlabeled images, exempting human annotation cost thoroughly. A robust consistency regularization based semi-supervised framework is proposed for STR, which can effectively solve the instability issue due to domain inconsistency between synthetic and real images. A character-level consistency regularization is designed to mitigate the misalignment between characters in sequence recognition. Extensive experiments on standard text recognition benchmarks demonstrate the effectiveness of the proposed method. It can steadily improve existing STR models, and boost an STR model to achieve new state-of-the-art results. To our best knowledge, this is the first consistency regularization based framework that applies successfully to STR.

1. Introduction

Scene text recognition (STR) is to recognize text in natural scenes and is widely used in many applications such as image retrieval, robot navigation and instant translation. Compared to traditional OCR, STR is more challenging because of multiple variations from the environment, various

---

*Part of the work was done when C.Zheng was an intern at SRCX.
†P. Wang is the corresponding author.

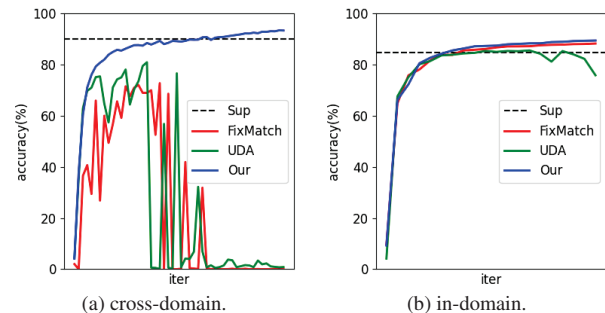(a) cross-domain.   (b) in-domain.

Figure 1. Scene text recognition test accuracy by using supervised training, existing consistency regularization SSL (UDA [50] and FixMatch [43]) and our method. Cross-domain means the labeled and unlabeled training data are from different domains (*e.g.* synthetic labeled *vs.* real unlabeled in our setting), while in-domain means they are from similar condition. UDA and FixMatch are feasible in in-domain condition but fail in cross-domain setting. It is observed that the test accuracy drops drastically during the training process, and the highest accuracy is even lower than that obtained by supervised training. By contrast, our method is able to stabilize the training process and improve test performance in both in-domain and cross-domain conditions.

font styles and complicated layouts.

Although STR has made great success, it is mainly researched in a fully supervised manner. Real labeled datasets in STR are usually small because the annotation work is expensive and time-consuming. Hence, two large synthetic datasets MJSynth [16, 17] and SynthText [15] are commonly used to train STR models and produce competitive results. However, there exists domain gap between synthetic and real data which restricts the effect of synthetic data. Briefly speaking, synthetic dataset can improve STR performance, but STR model is still hungry for real data.

Considering that it is easy to obtain a large scale of unla-

beled data in real world, many researchers intend to leverage unlabeled data and train models in a Semi-Supervised Learning (SSL) manner. Baek *et al*. [3] and Fang *et al*. [9] introduced self-training methods to train STR models and receive improved performance. Nevertheless, self-training requires a pre-trained model to predict pseudo-labels for unlabeled data and then re-trains the model, which affects the training efficiency. By contrast, Consistency Regularization (CR), another important component of state-of-the-art (SOTA) SSL algorithms, has not been well exploited in STR.

In this paper, we would like to explore a CR-based SSL approach to improve STR models, where only synthetic data and unlabeled real data are used for training, exempting human annotation cost thoroughly. CR assumes that the model should output similar predictions when fed perturbed versions of the same image [38]. It tends to outperform self-training on several SSL benchmarks [1,36]. Nevertheless, it is non-trivial to utilize existing CR methods to STR directly. We attempt to two representative CR approaches, UDA [50] and FixMatch [43]. Neither of them is feasible in our setting. As shown in Figure 1a, the models are quite unstable during the training process. Compared with experiments on image classification where they show big superiority, we assume the reasons lie in the following two aspects.

1) Our labeled images are synthetic while unlabeled images are from real scenarios. The domain gap between synthetic and real images affects the training stability. Actually, it is found that the collapsed models recognize synthetic inputs with a reasonable accuracy, but generate nearly identical outputs for all real inputs. We conjecture that they incorrectly utilize the domain gap to minimize the overall loss: they learn to distinguish between synthetic and real data, and learn reasonable representations for synthetic data to minimize the supervised loss, but simply project real data to identical outputs such that the consistency loss is zero. To validate this conjecture, we perform another experiment by using training images all from real. As shown in Figure 1b, the training processes of UDA and FixMatch become stable in such a setting. However, we aim to relieve human labeling cost. The introduced domain gap becomes an issue.

2) Different from image classification, STR is a kind of sequence prediction task. The alignment between character sequences brings another difficulty to consistency training.

To address the aforementioned problems, we propose a robust character-level consistency regularization based framework for STR. Firstly, inspired by BYOL [14] that prevents model collapse without using negative samples in contrastive learning, we propose an asymmetric consistency training structure for STR. Secondly, a character-level CR unit is proposed to ensure the character-level consistency during training process. Thirdly, some techniques are subtly adopted in training process, such as weight decay and domain adaption, which improve STR model furthermore.

The main contributions are summarized as follows:

1) We propose a robust consistency regularization based semi-supervised framework for STR. It is capable of tackling the cross-domain setting, thus more easily benefitting from labelled synthetic data and unlabeled real data. Compared with self-training approaches, our method is more efficient, without iteratively predicting and re-training.

2) Considering the sequential property of text, we propose a character-level consistency regularization (CCR) unit to ensure better sequence alignment between the outputs of two siamese models.

3) Extensive experiments are performed to analyze the effectiveness of the proposed framework. It boosts the performance of a variety of existing STR models. Despite free of human annotation, our method achieves new SOTA performance on several standard text recognition benchmarks for both regular and irregular text.

2. Related Work

2.1. Scene Text Recognition

Researches usually treat text recognition as a sequence prediction task and employ RNNs to model the sequences for recognition without character separation. Connectionist temporal classification (CTC) model [39,47] and attention-based encoder-decoder model [21, 40] are two commonly used frameworks for STR. The success of regular text recognition leads researchers to turn their attentions to irregular text recognition. [25, 27, 40, 41, 53, 56] rectified irregular text into regular ones to alleviate the difficulty in recognition. [23] and [52] employed 2D attention to handle the complicated layout of irregular text. [7, 49, 55] attempted to improve recognition accuracy by mitigating the alignment drift in attention. [10,34,54] tried to integrate semantic information from language model to enhance word recognition. All those methods needs to be trained in a fully supervised manner.

2.2. Semi-Supervised Learning

Semi-Supervised Learning (SSL) aims to use labeled data and additional unlabeled data to boost model performance. There are mainly two types of SSL methods that relate to our work, self-training [5, 13, 22, 51] and consistency regularization (CR) [20,30,43,44,50]. Self-training is simple and effective. It first employs labeled data to train a teacher model, then predicts pseudo labels for unlabeled data, and finally trains a student model using both labeled and pseudo-labeled data. Pseudo Label [22] and Noisy Student [51] are two popular variants. CR is based on the manifold assumption that model outputs should be consistent when fed different augmentation views of the same image. For example, Temporal Ensembling [20] encourages a consensus prediction of the unknown labels using the outputs of
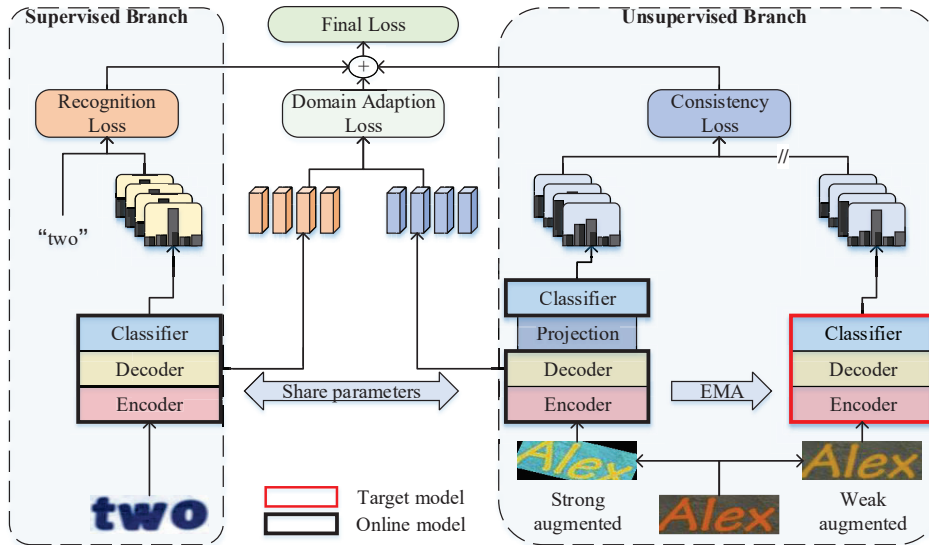
Figure 2. Overall framework of our proposed consistency regularization method for STR. Our model takes advantage of labeled synthetic data and unlabeled real data, exempting human annotation cost thoroughly. An asymmetric structure is designed with EMA and domain adaption to encourage a stable model training.

the network-in-training on different epochs. Mean Teacher [44] requires the outputs from teacher model and student model to be consistent, and updates teacher model by averaging student model weights. FixMatch [43] combines CR and pseudo-labeling for better performance. UDA [50] argues the importance of noise injection in consistency training, and achieves SOTA performance on a wide variety of language and vision SSL tasks.

2.3. Semi-Supervised Text Recognition

Some work has been proposed to train STR model with SSL. For instance, Gao *et al*. [11] adopted reinforcement learning techniques to exploit unlabeled data for STR performance improving. However, both labeled and unlabeled data are divided from synthetic data, without domain gap issue. [58] and [18] utilized domain adaption techniques to mitigate the domain shift between source and target data, so as to improve recognition results on target domain. Baek *et al*. [3] attempted to train STR model by using real data only, and tried both Pseudo Label and Mean Teacher to enhance STR performance. Fang *et al*. [9] proposed an autonomous, bidirectional and iterative language modeling for STR. A self-training strategy was applied with the ensemble of iterative prediction to increase STR performance further.

3. Proposed Method

3.1. Overview

As shown in Figure 2, our framework consists of an STR model for text recognition and a CR architecture to inte-

grate information from both labeled and unlabeled data. We adopt the attention based encoder-decoder STR model here for illustration. However, our framework is not restricted to autoregressive STR models. The encoder extracts discriminative features from input images, while the decoder generates character-level features. The classifier maps features into probabilities over character space via a linear transformation and Softmax.

We define two modes for STR model, named training mode and inference mode, according to whether the "ground-truth" character sequence is provided. In training mode, "ground-truth" characters are sent to the decoder for next character prediction. By contrast, in inference mode, the output of previous step is fed into decoder to infer next character. Both modes receive a special "BOS" token at the first step which means the start of decoding. Training mode ends when all ground-truth characters are input, while inference mode ends when generating an "EOS" token.

The CR architecture is inspired by UDA [50], which consists of two branches, namely supervised and unsupervised branch, as demonstrated in Figure 2. The supervised branch is trained on labeled data, while the unsupervised branch takes two augmented views of an unlabeled image as input, and requests the outputs to be similar with each other. Motivated by BYOL [14], we employ STR models with the same architecture but different parameters in unsupervised branch for the two views of inputs, denoted as online model and target model separately. The online model shares parameters with the one used in supervised branch. To overcome the instability during model training and improve STR perfor-

mance, an additional projection layer is introduced before classifier in online model of the unsupervised branch.

3.2. Supervised Branch

Supervised branch adopts the online STR model and runs in training mode, using the labeled synthetic data. Specially, denote the weight of online STR model as θ_o, which is comprised of parameters from three modules, *i.e.*, encoder, decoder and classifier, referring to Figure 2. Given the input image $\mathbf{X}^L$ and the ground-truth character sequence $\mathbf{Y}^{gt} = \{y_1^{gt}, y_2^{gt}, \ldots, y_t^{gt}\}$, the supervised branch outputs a sequence of vector $\mathbf{P}^L = \{\mathbf{p}_1^L, \mathbf{p}_2^L, \ldots, \mathbf{p}_T^L\}$. Cross-entropy loss is employed to train the model, *i.e.*,

$$\mathcal{L}_{reg} = \frac{1}{T} \sum_{t=1}^{T} \log p_t^L(y_t^{gt} \mid \mathbf{X}^L) \qquad (1)$$

where $p_t^L(y_t^{gt})$ represents the predicted probability of the output being y_t^{gt} at time step t. T is the sequence length.

3.3. Unsupervised Branch

Different from [50] and inspired by [14], unsupervised branch in our framework relies on two models, referred to as online STR model (with model parameter θ_o) and target STR model (with model parameter θ_t) respectively. The two models interact and learn from each other.

Given the input image without label $\mathbf{X}^U$, two different augmentation approaches are adopted which produce two augmented views of the image, denoted as $\mathbf{X}^{U_w}$ and $\mathbf{X}^{U_s}$ respectively. The online STR model takes $\mathbf{X}^{U_s}$ as input and runs in training mode. Motivated by the collapse preventing solution in [14], an additional projection layer is introduced between the decoder and classifier, as shown in Figure 2, and the parameters are denoted as θ_p independently. It is composed by 2 layers of perceptron with ReLU activation. The added projection layer makes the architecture asymmetric between the online and target model, which contributes to a stable training process. The classifier is then followed to transform the output vector into probabilities over character space, denoted as $\mathbf{P}^{U_s} = \{\mathbf{p}_1^{U_s}, \mathbf{p}_2^{U_s}, \ldots, \mathbf{p}_T^{U_s}\}$.

The target STR model takes $\mathbf{X}^{U_w}$ as input and runs in inference mode, which generates a sequence of probabilities $\mathbf{P}^{U_w} = \{\mathbf{p}_1^{U_w}, \mathbf{p}_2^{U_w}, \ldots, \mathbf{p}_T^{U_w}\}$. The output sequence is used as the reference target to train the online model. A stop-gradient operation is acted on the target model, and its parameters θ_t are an exponential moving average (EMA) of the online model parameter θ_o, *i.e.*,

$$\theta_t = \alpha \theta_t + (1 - \alpha)\theta_o \qquad (2)$$

where $\alpha \in [0, 1]$ is the target decay rate. EMA makes the target model produce relatively stable targets for online model, which helps to keep the projection layer in near optimal and benefits the model training as well.
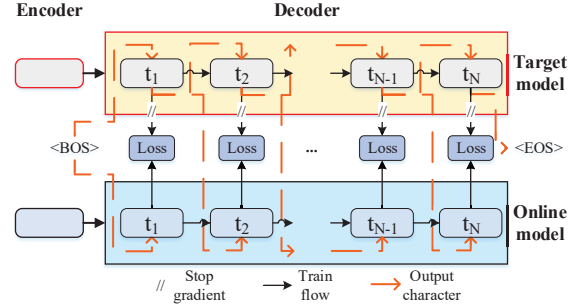

Figure 3. Character-level consistency regularization (CCR). In each time step t, target decoder and online decoder share the same output character produced by target decoder in previous time step so as to keep good character alignment. Consistency loss is computed between the outputs in each time step.

As indicated in [13, 50], regularizing predictions with low entropy would be beneficial to SSL. We sharpen the output from target STR model $\mathbf{P}^{U_w}$ by using a low Softmax temperature τ. Denote the output vector at step t before Softmax as $\mathbf{z}_t^{U_w} = \{z_1^{U_w}, z_2^{U_w}, \ldots, z_C^{U_w}\}$, C is the number of character classes, then

$$p_t^{U_w}(y_t) = \frac{\exp(z_{y_t}^{U_w}/\tau)}{\sum_{y_t'} \exp(z_{y_t'}^{U_w}/\tau)} \qquad (3)$$

We set $\tau = 0.4$ following [50].

The consistency training regularizes the outputs of $\mathbf{P}^{U_w}$ and $\mathbf{P}^{U_s}$ to be invariant. However, given that STR is a sequence recognition task, a character-level consistency regularization (CCR) unit is proposed for autoregressive decoder, so as to keep a good sequence alignment. As shown in Figure 3, in decoding time step t, a pseudo label is generated from target model by taking the class that has the highest probability in $\mathbf{p}_t^{U_w}$. The pseudo label will be used as the input for both online and target decoder in next time step. The design enforces online decoder and target decoder share the same context information, benefits character level alignment, and thus ensures a stable consistency training.

To alleviate the influence caused by noise samples in training process, we filter out noise samples based on their confidence scores in recognition. The confidence score is the cumulative product of the maximum output probability from target model in each decoding step, *i.e.*,

$$\mathbf{s}^{U_w} = \prod_{t=1}^{T} p_t^{U_w}(y_t \mid \mathbf{X}^{U_w}) \qquad (4)$$

The consistency loss used in unsupervised branch is then defined as:

$$\mathcal{L}_{cons} = \mathbb{I}(\mathbf{s}^{U_w} > \beta_U) \frac{1}{T} \sum_{t=1}^{T} Dist(\mathbf{p}_t^{U_w}, \mathbf{p}_t^{U_s}) \qquad (5)$$

where $\mathbb{I}(\mathbf{s}^{U_w} > \beta_U)$ is an indicator, β_U is a threshold for filtering out noises and $Dist(\cdot)$ is a function to measure the

character-level distance between $\mathbf{P}^{U_w}$ and $\mathbf{P}^{U_s}$. There are several choices for $Dist$, such as Cross Entropy (CE), KL-divergence or Mean Squared Error (MSE). KL-divergence is adopted in our framework by default.

3.4. Additional Training Techniques

Weight Decay. Weight decay is an important component in contrastive learning [6, 14] and SSL [43]. It is claimed that [45] weight decay in BYOL can help balance weights between predictor and online model dynamically, and improve the representation ability of online model. Here we also adopt it into our model training so as to improve the feature learning capability of online model.

Domain Adaption. To mitigate the domain shift in training data, a character-level domain adaptation unit is employed between the supervised and unsupervised branches, referring to [58]. Specially, in each decoding step, decoder of the online model extracts vision feature for the character to be decoded, denoted as $\mathbf{H}^L = \{\mathbf{h}_1^L, \mathbf{h}_2^L, \cdots, \mathbf{h}_T^L\}$ and $\mathbf{H}^{U_s} = \{\mathbf{h}_1^{U_s}, \mathbf{h}_2^{U_s}, \cdots, \mathbf{h}_T^{U_s}\}$ for features extracted in supervised and unsupervised branch respectively. Domain adaption loss is defined as

$$\mathcal{L}_{da} = \frac{1}{4d^2}\|(cov(\mathbf{H}^L) - cov(\mathbf{H}^{U_s})\|_F^2 \qquad (6)$$

where $\| \cdot \|_F^2$ denotes the squared matrix Frobenius norm, $cov(\mathbf{H})$ is covariance matrix, d is the feature dimension.

3.5. Overall Objective Function

We sum the three loss functions defined above. The overall objective function for training our proposed model is:

$$\mathcal{L}_{overall} = \mathcal{L}_{reg} + \lambda_{cons}\mathcal{L}_{cons} + \lambda_{da}\mathcal{L}_{da} \qquad (7)$$

where λ_{cons} and λ_{da} are hyper-parameters to balance three terms. We set $\lambda_{cons} = 1$ and $\lambda_{da} = 0.01$ empirically.

4. Experiment

4.1. Datasets

Two types of data are used here for training, *i.e.*, synthetic data with annotations and real data without label.

Two widely used synthetic datasets are adopted including **SynthText (ST)** [15] and **MJSynth (MJ)** [17], which results in 14.5M samples in total, referring to as **synthetic labeled data (SL)**.

For real unlabeled scene text data, we collected from three public available datasets, Places2 [59], OpenImages[1] and ImageNet ILSVRC 2012 [37]. CRAFT [4] was employed to detect text from these images. Then we cropped text images with detection scores larger than 0.7. Images with low resolution (width times height is less than 1000)

---

were also discarded. There are finally 10.5M images, denoted as **real unlabeled data (RU)**.

In addition, during ablation study, to demonstrate the superiority of the proposed framework, we also conduct experiments by using real labeled data collected by [3]. It has 278K images totally, named as **real labeled data (RL)**.

Six commonly used scene text recognition benchmarks are adopted to evaluate our method.

ICDAR 2013 (IC13) contains 1095 cropped word images. Following [54], we remove images that contain non-alphanumeric characters, which results in 857 test patches.

IIIT5K-Words (IIIT) [29] has 3000 nearly horizontal word patches for test.

Street View Text (SVT) [48] consists of 647 word images collected from Google Street View for test.

SVT-Perspective (SVTP) [35] contains 645 images for test, which are cropped from side-view snapshots in Google Street View.

CUTE80 (CUTE) [32] has 288 curved text images.

ICDAR 2015 (IC15) [19] contains 2077 word images cropped from incidental scene images. After removing images with non-alphanumeric characters, there are 1811 word patches left for test.

4.2. Evaluation Metric

Following common practice, we report word-level accuracy for each dataset. Moreover, in order to comprehensively evaluate models for their recognition performance on both regular and irregular text, following [3], we introduce an average score (Avg) which is the accuracy over the union of samples in all six datasets.

4.3. Implementation Details

The whole model is trained end-to-end without pre-training. We use a batch size of 384 for labeled data and 288 for unlabeled data. By default, we set the target decay rate $\alpha = 0.999$ and confidence threshold $\beta_U = 0.5$ respectively. Both supervised branch and unsupervised branch are jointly trained, while we only use the model in supervised branch in inference time.

Four STR models are adopted to validate the effectiveness of the proposed framework, with their default model configurations, including CRNN [39], MORAN [28], HGA [52] and TRBA [2]. Note that CRNN uses CTC for character decoding, which is non-autoregressive. Hence, C-CR is not adopted when training model with CRNN.

We adopt Adadelta when training MORAN or HGA, following their original optimization method. The learning rate is 1.0 initially and decreases during training process. AdamW [26] optimizer is adopted when using CRNN or TRBA model. Following [3], we use the one-cycle learning rate scheduler [42] with a maximum learning rate of 0.001. The weight decay rate is aligned with the used STR model.

---

[1] https://storage.googleapis.com/openimages/web/index.html

The unsupervised branch takes two augmented views of an image as input. Here we define two types of augmentations, *i.e.*, StrongAug and WeakAug. StrongAug is borrowed from RandAugment [8] which includes multiple augmentation strategies on both geometry transformations and color jitter. Considering Cutout may crop some characters from the image which will corrupt the semantic information of text, we remove "Cutout" operation from RandAugment. WeakAug only has color jitter, including brightness, contrast, saturation and hue. In our framework, we use WeakAug for target model and StrongAug for online models of both supervised and unsupervised branches.

4.4. Comparison with SOTA

We perform experiments by using different STR models. For fair comparison, we also reproduce those models under supervised setting using the same data augmentation strategy as that used in our semi-supervised training. As presented in Table 1, our reproduced models have comparable or even higher accuracies than that reported in the original paper. Those results provide an even fair baseline to show the advantage of our method. Experiments with their original settings can be found in Supplementary.

By training with the proposed framework using additional unlabeled real images, all models gain improvement. To be specific, CRNN improves by 3.1% (from 82.8% to 85.9%) on average, MORAN increases from 88.5% to 90.2% (+1.7%). HGA has an accuracy increase of 2.5% (from 88.7% to 91.2%) and TRBA has an increase of 3.0% (from 91.5% to 94.5%). The consistent enhancement over different STR models shows the effectiveness and universality of our proposed method. Specially, the performance gain over irregular text (IC15, SVTP and CUTE) is more obvious, since irregular text has more variance on appearance which is hard to be generated by synthetic engine.

Note that although TRBA is worse than ABINet [9] in supervised setting (91.5% *vs.* 92.7%), our framework helps TRBA outperform ABINet that adopts self-training in semi-supervised setting (94.5% *vs.* 93.5%), which proves the superiority of our proposed CR method again. Compared with other SOTA work, our proposed framework with TRBA achieves the highest accuracies on vast majority of test datasets (only except IIIT), which demonstrates its robustness for both regular and irregular text recognition.

In addition, to accelerate training process, we perform an experiment with TRBA using only 10% synthetic labeled data (denoted as "SL_{sm}" that contains only 1.45M images) and 10% real unlabeled data (denoted as "RU_{sm}" which has 1.05M images). Surprisingly, experimental results is fairly good with the average score of 93.2%, even higher than that obtained by $TRBA_{pr}$ (91.5%) and ABINet [9] (92.7%). It should be noted that $TRBA_{pr}$ and ABINet are trained in a fully supervised manner using all synthetic data
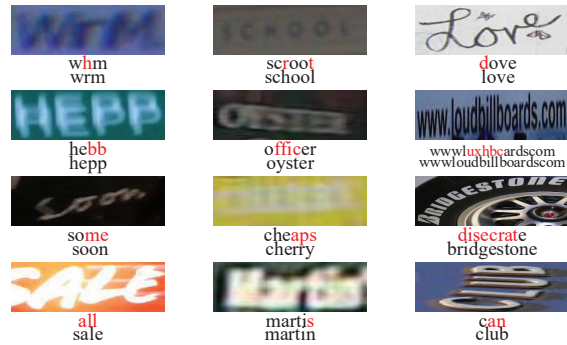

Figure 4. Hard examples that can be successfully recognized by using our method. The first line shows the recognition results by $TRBA_{pr}$, which include mistakes (red characters), while the second line are results by $TRBA_{cr}$. Our method enables TRBA address even tough samples like dark, blur, or severe distortion.

(14.5M). The training data is 5.8 times more than that used in $TRBA_{sm}$. The excellent results suggest the necessary of using real images in training STR models and the advantage of our semi-supervised training framework.

In Figure 4, we present several examples that can be correctly recognized by $TRBA_{cr}$ but encounter failure when using $TRBA_{pr}$. Although the employed real images are unlabeled, STR models can still get benefit from our method, particularly for recognizing text that are severely blurred, distorted, or with artistic font.

4.5. Ablation Study

In order to analyze the proposed model, we conduct a series of ablation experiments in this section. All ablation experiments are performed using TRBA because of its good performance. SL_{sm} and RU_{sm} are employed for fast training. More experiments with different data size can be found in Supplementary.

4.5.1 Effect of domain gap on model stability

In this work, we propose a stable CR based SSL framework for STR. As stated in Section 1, we guess it is the domain inconsistency among training data used in STR that causes the instability or even failure by previous CR methods.

To prove this conjecture, we perform experiments using domain consistent training data (in-domain data). Specially, we split the real labeled training data RL into RL_{20p} and RL_{80p} with a ratio of 1:4. RL_{20p} is adopted with labels while RL_{80p} is employed without annotations. SOTA CR methods are tested, including FixMatch [43] and UDA [50]. As presented in Table 2, when training data is from the same domain, they work well. The test accuracy increase by 3.6% using FixMatch and 2.6% using UDA. However, when the training data is from different domains, *e.g.*, SL_{sm} and RU_{sm}, their training processes become unstable. We test the models before collapse. The recognition

	Methods	Labeled Dataset	Unlabeled Dataset	Regular Text			Irregular Text			Avg
				IC13	SVT	IIIT	IC15	SVTP	CUTE	
SOTA Methods	Shi *et al.* [39] (CRNN)	MJ	-	-	80.8	78.2	-	-	-	-
	Luo *et al.* [28](MORAN)	SL	-	-	88.3	93.4	77.8	79.7	81.9	-
	Yang *et al.* [52](HGA)	SL	-	-	88.9	94.7	79.5	80.9	85.4	-
	Baek *et al.* [2](TRBA)	SL	-	-	87.5	87.9	-	79.2	74.0	-
	Liao *et al.* [24](Mask TextSpotter)	SL	-	95.3	91.8	93.9	77.3	82.2	87.8	88.3
	Wan *et al.* [46](TextScanner)	SL	-	92.9	90.1	93.9	79.4	84.3	83.3	88.5
	Wang *et al.* [49](DAN)	SL	-	93.9	89.2	94.3	74.5	80.0	84.4	87.2
	Yue *et al.* [55](RobustScanner)	SL	-	94.8	88.1	95.3	77.1	79.5	90.3	88.4
	Qiao *et al.* [34](SRN)	SL	-	95.5	91.5	94.8	82.7	85.1	87.8	90.4
	Zhang *et al.* [57](SPIN)	SL	-	-	90.9	95.2	82.8	84.3	83.2	-
	Mou *et al.* [31](PlugNet)	SL	-	-	92.3	94.4	-	84.3	84.3	-
	Qiao *et al.* [33](PIMNet)	SL	-	95.2	91.2	95.2	83.5	84.3	84.4	90.5
	Fang *et al.* [9](ABINet)	SL	-	<u>97.4</u>	<u>93.5</u>	96.2	<u>86.0</u>	89.3	89.2	92.7
	Gao *et al.* [12]	10% SL	90% SL	-	78.1	74.8	-	-	-	-
	Baek *et al.* [3](CRNN)	RL	Book32 et al.	-	84.3	89.8	-	74.6	82.3	-
	Baek *et al.* [3](TRBA)	RL	Book32 et al.	-	91.3	94.8	-	82.7	88.1	-
	Fang *et al.* [9](ABINet)	SL	Uber-Text	97.3	94.9	**96.8**	87.4	90.1	93.4	93.5
Ours	CRNN-pr	SL	-	91.0	82.2	90.2	71.6	70.7	81.3	82.8
	CRNN-cr	SL	RU	92.4	87.9	92.0	75.8	75.7	85.8	85.9
	MORAN-pr	SL	-	95.1	90.4	93.4	79.7	80.6	85.4	88.5
	MORAN-cr	SL	RU	96.5	93.0	94.1	82.6	82.9	88.5	90.2
	HGA-pr	SL	-	95.0	89.5	93.6	79.8	81.1	87.8	88.7
	HGA-cr	SL	RU	95.4	93.2	94.9	84.0	86.8	92.0	91.2
	TRBA-pr	SL	-	97.3	91.2	95.3	84.2	86.4	92.0	91.5
	TRBA-cr	10% SL	10% RU	97.3	94.7	96.2	87.0	89.6	**94.4**	93.2
	TRBA-cr	SL	RU	**98.3**	**96.3**	96.5	**89.3**	**93.3**	93.4	**94.5**

Table 1. Comparison with SOTA methods on STR test accuracy. In each column, the best result is shown in bold, and the best result in supervised setting is shown with underline. "-pr" means our reproduced results and "-cr" means using our **c**onsistency **r**egularization method. Our method improves STR models firmly, and propels TRBA towards new SOTA performance on test benchmarks.

accuracies are even lower than that obtained by only using SL_{sm}, with performance degradation of 11.0% (FixMatch) and 4.6% (UDA) separately.

By contrast, our method is able to improve the recognition accuracy no matter the training data is from similar domain or not. In comparison to the results by fully supervised training, our method improves STR model accuracy steadily by 4.5% (84.8% to 89.3%) using in-domain data and 3.3% (89.9% to 93.2%) in cross-domain setting. The performance gain in in-domain setting is even larger than that brought by FixMatch and UDA.

4.5.2 Ablation on model units

The techniques used in our method include an additional projection layer for asymmetric structure, EMA, domain adaption and weight decay. Here we analyze the effect of each unit in detail. The experiments are performed with C-CR added to benefit character-level consistency.

As presented in Table 3, the use of additional projection layer can improve the final average score by 0.7%. However, the performance is still lower than that obtained under fully supervised setting (87.7% *vs.* 89.9%). As indicated in [45], without weight decay, the consistency between online and target outputs is dependent mainly on the projection layer, rendering the online model weights inferior. Weight decay helps balance weights between online model

	Labeled/ Unlabeled Data	Methods	IC13 / IC15	SVT / SVTP	IIIT / CUTE	Avg
In-domain	RL$_{20p}$(55.7K)/ -	Sup	90.1 / 77.6	87.5 / 78.0	88.8 / 83.0	84.8
	RL$_{20p}$(55.7K)/ RL$_{80p}$(223K)	FixMatch	93.0 / 82.3	88.6 / 82.5	92.0 / 88.5	88.4
		UDA	92.5 / 80.7	88.6 / 80.9	91.4 / 88.5	87.4
		Ours	93.8 / 82.5	91.5 / 83.6	92.9 / 88.5	89.3
Cross-domain	SL$_{sm}$(1.45M)/ -	Sup	96.0 / 82.4	90.0 / 82.6	94.4 / 88.9	89.9
	SL$_{sm}$(1.45M)/ RU$_{sm}$(1.06M)	FixMatch	90.0 / 72.6	86.2 / 77.2	79.2 / 69.1	78.9
		UDA	94.2 / 75.7	85.3 / 79.5	90.0 / 82.3	85.3
		Ours	97.3 / 87.0	94.7 / 89.6	96.2 / 94.4	**93.2**

Table 2. Experiments with CR methods on in-domain and cross-domain data settings. Our method can consistently improve recognition accuracy. The results of FixMatch and UDA in cross-domain setting are obtained by the models before collapse.

and projection layer dynamically. The use of weight decay, with projection layer, increases the average score on test data by another 3.5%, surpassing the supervised results. EMA mechanism brings an accuracy gain of 1.6% furthermore as it helps keep projection layer in near-optimal and improves training stability. Lastly, the adding of domain adaption im-

Projection	WD	EMA	DA	IC13 IC15	SVT SVTP	IIIT CUTE	Avg
				94.2 80.5	91.5 84.0	88.7 84.0	87.0
✔				94.5 81.6	90.1 86.1	89.5 85.4	87.7
✔	✔			97.2 85.9	93.0 87.0	93.5 91.3	91.2
✔	✔	✔		96.7 86.7	94.6 89.3	95.9 92.7	92.8
✔	✔	✔	✔	97.3 87.0	94.7 89.6	96.2 94.4	**93.2**

Table 3. Ablation on model units. "Projection" means using additional projection layer before classifier. "WD" means weight decay, "EMA" means using EMA for target model. "DA" means domain adaption.

Method	IC13 IC15	SVT SVTP	IIIT CUTE	Avg
SCR	96.6 84.9	93.0 85.9	96.4 93.1	92.2
CCR	97.3 87.0	94.7 89.6	96.2 94.4	**93.2**

Table 4. Effect of our proposed CCR. Compared to using standard consistency regularization, training with CCR conduces to 1% average score increase for TRBA.

Consistency Loss	IC13 IC15	SVT SVTP	IIIT CUTE	Avg
MSE	96.3 84.0	92.0 86.8	94.2 92.0	91.0
CE	97.4 86.9	94.3 89.8	96.3 92.7	**93.2**
KL-divergence	97.3 87.0	94.7 89.6	96.2 94.4	**93.2**

Table 5. Ablation on different distance functions used in consistency loss. CE and KL-divergence leads to similar performance, better than MSE.

Method	IC13 IC15	SVT SVTP	IIIT CUTE	Avg
Pseudo Label (PL)	95.9 82.9	91.2 85.7	95.4 90.6	90.9
Noisy Student (NS)	96.3 85.5	94.4 86.7	96.1 94.1	92.4
Ours	97.3 87.0	94.7 89.6	96.2 94.4	**93.2**

Table 6. Comparison with other semi-supervised methods. Our method brings more benefit to STR model and outperforms the other approaches.

empirically, CE leads to similar recognition performance with KL-divergence, while MSE results in lower accuracies (93.2% *vs*. 91.0%).

4.6. Comparison with Other Semi-supervised Methods

We compare our method with other SSL approaches that have been successfully used in STR, including Pseudo Label (PL) [22] and Noisy Student (NS) [51]. TRBA is used as the basic model. PL based SSL is performed following the practice in [3], while NS based SSL is following [51], with the threshold $\beta_U = 0.5$ and 3 iterations of re-training.

The results are shown in Table 6. Our CR based method outperforms all the others, with the resulted average score 2.3% higher than PL and 0.8% higher than NS. Note that compared to NS, our training process is more efficient, without time consuming iterations.

5. Conclusion

In this paper, we propose a robust character-level consistency regularization method for STR. Our framework consists of a supervised branch trained with synthetic labeled data, and an unsupervised branch trained by two augmented views of real unlabeled images. An asymmetric structure is designed with EMA, weight decay and domain adaption to encourage a stable model training and overcome the domain gap issue caused by synthetic and real images. Moreover, a character-level consistency regularization unit is proposed to ensure better character alignment. Without using any human annotated data, our method is able to improve existing STR models by a large margin, and achieves new SOTA performance on STR benchmarks.

Acknowledgements

proves the average test accuracy up to 93.2%.

4.5.3 Effect of CCR

Another contribution of this work is a character-level consistency regularization (CCR) unit to handle the specially sequential property of STR task. Instead of letting online model and target model run separately in unsupervised branch (standard consistency regularization, SCR), and only restricting their final outputs by consistency loss, we proposed CCR to enforce the same context information for both online and target model. Experimental results in Table 4 prove the effectiveness of CCR. It helps TRBA improve 1% more on the final test accuracy.

4.5.4 Ablation on distance measure functions

By default, we use KL-divergence to measure the consistency in loss function (5). Here we test other distance measure functions, such as CE and MSE. As presented in Table 5,

References

[1] Abulikemu Abuduweili, Xingjian Li, Humphrey Shi, Cheng-Zhong Xu, and Dejing Dou. Adaptive consistency regularization for semi-supervised transfer learning. In *CVPR*, 2021. 2

[2] Jeonghun Baek, Geewook Kim, Junyeop Lee, Sungrae Park, Dongyoon Han, Sangdoo Yun, Seong Joon Oh, and Hwalsuk Lee. What is wrong with scene text recognition model comparisons? dataset and model analysis. In *ICCV*, pages 4714–4722, 2019. 5, 7

[3] Jeonghun Baek, Yusuke Matsui, and Kiyoharu Aizawa. What if we only use real datasets for scene text recognition? toward scene text recognition with fewer labels. In *CVPR*, pages 3113–3122, 2021. 2, 3, 5, 7, 8

[4] Youngmin Baek, Bado Lee, Dongyoon Han, Sangdoo Yun, and Hwalsuk Lee. Character region awareness for text detection. In *CVPR*, pages 9365–9374, 2019. 5

[5] Lucas Beyer, Xiaohua Zhai, Avital Oliver, and Alexander Kolesnikov. S4L: self-supervised semi-supervised learning. In *ICCV*, pages 1476–1485, 2019. 2

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607, 2020. 5

[7] Zhanzhan Cheng, Fan Bai, Yunlu Xu, Gang Zheng, Shiliang Pu, and Shuigeng Zhou. Focusing attention: Towards accurate text recognition in natural images. In *ICCV*, pages 5086–5094, 2017. 2

[8] Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. Randaugment: Practical automated data augmentation with a reduced search space. In *CVPR Workshops*, pages 3008–3017, 2020. 6

[9] Shancheng Fang, Hongtao Xie, Yuxin Wang, Zhendong Mao, and Yongdong Zhang. Read like humans: Autonomous, bidirectional and iterative language modeling for scene text recognition. In *CVPR*, pages 7098–7107, 2021. 2, 3, 6, 7

[10] Shancheng Fang, Hongtao Xie, Zheng-Jun Zha, Nannan Sun, Jianlong Tan, and Yongdong Zhang. Attention and language ensemble for scene text recognition with convolutional sequence modeling. In *ACM Multimedia*, pages 248–256, 2018. 2

[11] Yunze Gao, Yingying Chen, Jinqiao Wang, and Hanqing Lu. Semi-supervised scene text recognition. *IEEE Trans. Image Process.*, 30:3005–3016, 2021. 3

[12] Yunze Gao, Yingying Chen, Jinqiao Wang, and Hanqing Lu. Semi-supervised scene text recognition. *IEEE Trans. Image Process.*, 30:3005–3016, 2021. 7

[13] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *NIPS*, pages 529–536, 2004. 2, 4

[14] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Ávila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent - A new approach to self-supervised learning. In *NeurIPS*, 2020. 2, 3, 4, 5

[15] Ankush Gupta, Andrea Vedaldi, and Andrew Zisserman. Synthetic data for text localisation in natural images. In *CVPR*, pages 2315–2324, 2016. 1, 5

[16] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Synthetic data and artificial neural networks for natural scene text recognition. *NIPS Deep Learning Workshop*, 2014. 1

[17] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Reading text in the wild with convolutional neural networks. *Int. J. Comput. Vis.*, 116(1):1–20, 2016. 1, 5

[18] Lei Kang, Marçal Rusiñol, Alicia Fornés, Pau Riba, and Mauricio Villegas. Unsupervised adaptation for synthetic-to-real handwritten word recognition. In *WACV*, pages 3491–3500. IEEE, 2020. 3

[19] Dimosthenis Karatzas, Lluis Gomez-Bigorda, Anguelos Nicolaou, Suman K. Ghosh, Andrew D. Bagdanov, Masakazu Iwamura, Jiri Matas, Lukas Neumann, Vijay Ramaseshan Chandrasekhar, Shijian Lu, Faisal Shafait, Seiichi Uchida, and Ernest Valveny. ICDAR 2015 competition on robust reading. In *ICDAR*, pages 1156–1160, 2015. 5

[20] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. In *ICLR*, 2017. 2

[21] Chen-Yu Lee and Simon Osindero. Recursive recurrent nets with attention modeling for OCR in the wild. In *CVPR*, pages 2231–2239, 2016. 2

[22] Dong-Hyun Lee. Pseudo-label : The simple and efficient semi-supervised learning method for deep neural networks. *ICML 2013 Workshop : Challenges in Representation Learning (WREPL)*, 07 2013. 2, 8

[23] Hui Li, Peng Wang, Chunhua Shen, and Guyu Zhang. Show, attend and read: A simple and strong baseline for irregular text recognition. In *AAAI*, pages 8610–8617, 2019. 2

[24] Minghui Liao, Pengyuan Lyu, Minghang He, Cong Yao, Wenhao Wu, and Xiang Bai. Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 43(2):532–548, 2021. 7

[25] Wei Liu, Chaofeng Chen, and Kwan-Yee K. Wong. Charnet: A character-aware neural network for distorted scene text recognition. In Sheila A. McIlraith and Kilian Q. Weinberger, editors, *AAAI*, pages 7154–7161, 2018. 2

[26] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 5

[27] Canjie Luo, Lianwen Jin, and Zenghui Sun. MORAN: A multi-object rectified attention network for scene text recognition. *Pattern Recognit.*, 90:109–118, 2019. 2

[28] Canjie Luo, Lianwen Jin, and Zenghui Sun. MORAN: A multi-object rectified attention network for scene text recognition. *Pattern Recognit.*, 90:109–118, 2019. 5, 7

[29] Anand Mishra, Karteek Alahari, and C. V. Jawahar. Scene text recognition using higher order language priors. In *BMVC*, pages 1–11, 2012. 5

[30] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: A regularization

method for supervised and semi-supervised learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(8):1979–1993, 2019. 2

[31] Yongqiang Mou, Lei Tan, Hui Yang, Jingying Chen, Leyuan Liu, Rui Yan, and Yaohong Huang. Plugnet: Degradation aware scene text recognition supervised by a pluggable super-resolution unit. In *ECCV (15)*, volume 12360 of *Lecture Notes in Computer Science*, pages 158–174, 2020. 7

[32] Trung Quy Phan, Palaiahnakote Shivakumara, Shangxuan Tian, and Chew Lim Tan. Recognizing text with perspective distortion in natural scenes. In *ICCV*, pages 569–576, 2013. 5

[33] Zhi Qiao, Yu Zhou, Jin Wei, Wei Wang, Yuan Zhang, Ning Jiang, Hongbin Wang, and Weiping Wang. Pimnet: A parallel, iterative and mimicking network for scene text recognition. In *ACM Multimedia*, pages 2046–2055, 2021. 7

[34] Zhi Qiao, Yu Zhou, Dongbao Yang, Yucan Zhou, and Weiping Wang. SEED: semantics enhanced encoder-decoder framework for scene text recognition. In *CVPR*, pages 13525–13534, 2020. 2, 7

[35] Anhar Risnumawan, Palaiahnakote Shivakumara, Chee Seng Chan, and Chew Lim Tan. A robust arbitrary text detection system for natural scene images. *Expert Syst. Appl.*, 41(18):8027–8048, 2014. 5

[36] Mamshad Nayeem Rizve, Kevin Duarte, Yogesh S Rawat, and Mubarak Shah. In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning. In *ICLR*, 2021. 2

[37] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.*, 115(3):211–252, 2015. 5

[38] M. Sajjadi, M. Javanmardi, and T. Tasdizen. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In *NIPS*, 2016. 2

[39] Baoguang Shi, Xiang Bai, and Cong Yao. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(11):2298–2304, 2017. 2, 5, 7

[40] Baoguang Shi, Xinggang Wang, Pengyuan Lyu, Cong Yao, and Xiang Bai. Robust scene text recognition with automatic rectification. In *CVPR*, pages 4168–4176, 2016. 2

[41] Baoguang Shi, Mingkun Yang, Xinggang Wang, Pengyuan Lyu, Cong Yao, and Xiang Bai. ASTER: an attentional scene text recognizer with flexible rectification. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(9):2035–2048, 2019. 2

[42] Leslie N. Smith and Nicholay Topin. Super-convergence: very fast training of neural networks using large learning rates. In *Defense + Commercial Sensing*, 2019. 5

[43] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *NeurIPS*, 2020. 1, 2, 3, 5, 6

[44] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *NIPS*, pages 1195–1204, 2017. 2, 3

[45] Yuandong Tian, Xinlei Chen, and Surya Ganguli. Understanding self-supervised learning dynamics without contrastive pairs. In *ICML*, volume 139 of *Proceedings of Machine Learning Research*, pages 10268–10278. PMLR, 2021. 5, 7

[46] Zhaoyi Wan, Minghang He, Haoran Chen, Xiang Bai, and Cong Yao. Textscanner: Reading characters in order for robust scene text recognition. In *AAAI*, pages 12120–12127, 2020. 7

[47] Jianfeng Wang and Xiaolin Hu. Gated recurrent convolution neural network for ocr. In *NIPS*, 2017. 2

[48] Kai Wang, Boris Babenko, and Serge J. Belongie. End-to-end scene text recognition. In *ICCV*, pages 1457–1464, 2011. 5

[49] Tianwei Wang, Yuanzhi Zhu, Lianwen Jin, Canjie Luo, Xiaoxue Chen, Yaqiang Wu, Qianying Wang, and Mingxiang Cai. Decoupled attention network for text recognition. In *AAAI*, pages 12216–12224, 2020. 2, 7

[50] Qizhe Xie, Zihang Dai, Eduard H. Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. In *NeurIPS*, 2020. 1, 2, 3, 4, 6

[51] Qizhe Xie, Minh-Thang Luong, Eduard H. Hovy, and Quoc V. Le. Self-training with noisy student improves imagenet classification. In *CVPR*, pages 10684–10695, 2020. 2, 8

[52] Lu Yang, Peng Wang, Hui Li, Zhen Li, and Yanning Zhang. A holistic representation guided attention network for scene text recognition. *Neurocomputing*, 414:67–75, 2020. 2, 5, 7

[53] Mingkun Yang, Yushuo Guan, Minghui Liao, Xin He, Kaigui Bian, Song Bai, Cong Yao, and Xiang Bai. Symmetry-constrained rectification network for scene text recognition. In *ICCV*, pages 9146–9155, 2019. 2

[54] Deli Yu, Xuan Li, Chengquan Zhang, Tao Liu, Junyu Han, Jingtuo Liu, and Errui Ding. Towards accurate scene text recognition with semantic reasoning networks. In *CVPR*, pages 12110–12119, 2020. 2, 5

[55] Xiaoyu Yue, Zhanghui Kuang, Chenhao Lin, Hongbin Sun, and Wayne Zhang. Robustscanner: Dynamically enhancing positional clues for robust text recognition. In *ECCV (19)*, volume 12364 of *Lecture Notes in Computer Science*, pages 135–151, 2020. 2, 7

[56] Fangneng Zhan and Shijian Lu. ESIR: end-to-end scene text recognition via iterative image rectification. In *CVPR*, pages 2059–2068, 2019. 2

[57] Chengwei Zhang, Yunlu Xu, Zhanzhan Cheng, Shiliang Pu, Yi Niu, Fei Wu, and Futai Zou. SPIN: structure-preserving inner offset network for scene text recognition. In *AAAI*, pages 3305–3314, 2021. 7

[58] Yaping Zhang, Shuai Nie, Wenju Liu, Xing Xu, Dongxiang Zhang, and Heng Tao Shen. Sequence-to-sequence domain adaptation network for robust text image recognition. In *CVPR*, pages 2740–2749, 2019. 3, 5

[59] Bolei Zhou, Àgata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(6):1452–1464, 2018. 5