What If We Only Use Real Datasets for Scene Text Recognition? Toward Scene Text Recognition With Fewer Labels

Jeonghun Baek Yusuke Matsui Kiyoharu Aizawa
The University of Tokyo
{baek, matsui, aizawa}@hal.t.u-tokyo.ac.jp

Abstract

Scene text recognition (STR) task has a common practice: All state-of-the-art STR models are trained on large synthetic data. In contrast to this practice, training STR models only on fewer real labels (STR with fewer labels) is important when we have to train STR models without synthetic data: for handwritten or artistic texts that are difficult to generate synthetically and for languages other than English for which we do not always have synthetic data. However, there has been implicit common knowledge that training STR models on real data is nearly impossible because real data is insufficient. We consider that this common knowledge has obstructed the study of STR with fewer labels. In this work, we would like to reactivate STR with fewer labels by disproving the common knowledge. We consolidate recently accumulated public real data and show that we can train STR models satisfactorily only with real labeled data. Subsequently, we find simple data augmentation to fully exploit real data. Furthermore, we improve the models by collecting unlabeled data and introducing semi- and self-supervised methods. As a result, we obtain a competitive model to state-of-the-art methods. To the best of our knowledge, this is the first study that 1) shows sufficient performance by only using real labels and 2) introduces semi- and self-supervised methods into STR with fewer labels. Our code and data are available: https://github.com/ku21fan/STR-Fewer-Labels.

1. Introduction

Reading text in natural scenes is generally divided into two tasks: detecting text regions in scene images and recognizing the text in the regions. The former is referred to as scene text detection (STD), and the latter as scene text recognition (STR). Since STR can serve as a substitute for manual typing performed by humans, we frequently employ STR for various purposes: translation by recognizing foreign languages, street sign recognition for autonomous driving, various card recognition to input personal information, etc. Unlike optical character recognition (OCR), which focuses on reading texts in cleaned documents, STR also addresses irregular cases in our lives, such as curved or perspective texts, occluded texts, texts in low-resolution images, and texts written in difficult font.

To address these irregular cases, prior works have developed STR models comprising deep neural networks. For example, to address curved or perspective texts, image transformation modules have been proposed to normalize them into horizontal images [41, 60, 55]. Qiao *et al*. [38] has integrated a pretrained language model into STR models to recognize occluded text. Wang *et al*. [53] and Mou *et al*. [33] have introduced a super-resolution module into STR models to handle low-resolution images.

While prior works have improved STR models, the study of training STR models only on fewer real labels (STR with fewer labels) is insufficient. After emerging large synthetic data [15] in 2014, the study of STR with fewer labels has decreased. All state-of-the-art methods use large synthetic data to train STR models instead of sole real data [40, 41, 23, 60, 24, 1, 55, 52, 50, 56, 26, 38, 57, 33]. Implicit common knowledge has been made; *training STR models only on real data results in low accuracy because the amount of real data is very small.* This common knowledge may have hindered studies on STR with fewer labels.

STR with fewer labels is important when we have to train STR models without synthetic data. In practical applications, generating synthetic data close to real data can be difficult depending on the target domain, such as handwritten text or artistic text. In the other case, when we have to recognize languages other than English, there are not always synthetic data for them. Generating appropriate synthetic data for them is difficult for those who do not know target languages.

In this paper, we would like to reactivate STR with fewer labels for such cases. As a first step, we disprove the common knowledge by showing that we can train STR models satisfactorily only with real labels. This is not previously feasible. Because the real data was small, STR mod-

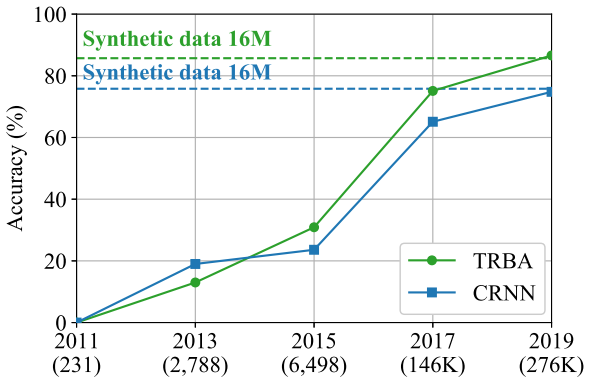


Figure 1: Accuracy vs. number of accumulated real labeled data. Every two years, public real data has been accumulated. In our experiments, we find that accuracy obtained using real data approaches that obtained using synthetic data, along with increment of real data. CRNN [40] and TRBA [1] are VGG-based and ResNet-based STR models, respectively.

els trained on real data had low accuracy, as shown in Figure 1. However, the public real data are accumulated every two years. We consolidate accumulated real data (276K), and find that the accuracy of STR models [40, 1] trained on them is close to that of synthetic data (16M). Namely, we can train STR models only on real data instead of synthetic data. It is high time to change the prevalent perspective from *"We don't have enough real data to train STR models"* to *"We have enough real data to train STR models"*. It is also a good time to study STR with fewer labels.

To improve STR with fewer labels, we find simple yet effective data augmentations to fully exploit real data. In addition, we collect unlabeled data and introduce a semi- and self-supervised framework into the STR. With extensive experiments, we analyze the contribution of them and demonstrate that we can obtain a competitive model to state-of-the-art methods by only using real data. Furthermore, we investigate if our method is also useful when we have both synthetic and real data.

2. Common Practice in STR Dataset

According to a benchmark study [1], obtaining enough real data is difficult because of the high labeling cost. Thus, STR models are generally trained on large synthetic data instead of real data. Real data has been used for evaluation.

2.1. Synthetic Datasets for Training

There are two major synthetic datasets.

MJSynth (MJ) [15] is generated for STR, and it contains 9M word boxes. Each word is generated from a 90K English lexicon and over 1,400 Google Fonts, as shown in Figure 2a.

(a) MJ word boxes (b) ST scene image

Figure 2: Examples of two major synthetic datasets.

SynthText (ST) [11] is originally generated for scene text detection. The texts are rendered onto scene images, as shown in Figure 2b. For STR, we crop the texts in scene images and use them for training. ST has 7M word boxes.

2.2. Real Benchmark Datasets for Evaluation

Six real datasets have been used to evaluate STR models.

Street View Text (SVT) [51] is collected from Google Street View, and contains texts in street images. It contains 257 images for training and 647 images for evaluation.

IIIT5K-Words (IIIT) [31] is crawled from Google image searches with query words such as "billboards" and "movie posters." It contains 2,000 images for training and 3,000 images for evaluation.

ICDAR2013 (IC13) [19] is created for the ICDAR 2013 Robust Reading competition. It contains 848 images for training and 1,015 images for evaluation.

ICDAR2015 (IC15) [18] is collected by people who wear Google Glass, and thus, many of them contain perspective texts and some of them are blurry. It contains 4,468 images for training and 2,077 images for evaluation.

SVT Perspective (SP) [37] is collected from Google Street View, similar to SVT. Unlike SVT, SP contains many perspective texts. It contains 645 images for evaluation.

CUTE80 (CT) [39] is collected for curved text. The images are captured by a digital camera or collected from the Internet. It contains 288 cropped images for evaluation.

They are generally divided into regular (SVT, IIIT, IC13) and irregular (IC15, SP, CT) datasets. The former mainly contains horizontal texts, while the latter mainly contains perspective or curved texts.

3. Consolidating Public Real Datasets

Recently, public real data has been sufficiently accumulated to train STR models, as shown in Figure 1. We consolidate the training set of public real datasets from 2011 to 2019. Table 1 lists datasets. Figure 3 shows the examples of word boxes. Before using the original data directly for training, we conduct some preprocessing on datasets for our task. We summarize the processes in §3.3, and details are in the supplementary materials.

(a) Year 2011 (b) Year 2013 (c) Year 2015 (d) Year 2017 (e) Year 2019

Figure 3: Examples of accumulated real labeled data. More examples are provided in the supplementary materials.

3.1. Real Labeled Datasets Have Increased

Recently, many irregular texts are accumulated as shown in Year 2015, 2017, and 2019 in Figure 3. They can make STR models more robust. Many real labeled datasets are released from ICDAR competitions: IC13, IC15, RCTW, ArT, LSVT, MLT19, and ReCTS (7 of 11 datasets in Table 1). ICDAR competitions are held every two years, and real labeled datasets have also increased in number. We summarize real labeled datasets for every two years.

(a) Year 2011 (SVT) and (b) Year 2013 (IIIT, IC13): Most of images are horizontal texts in the street.

(c) Year 2015 (IC15): Images captured by Google Glass under movement of the wearer, and thus many are perspective texts, blurry, or low-resolution images.

(d) Year 2017 (COCO, RCTW, Uber):

COCO-Text (COCO) [49] is created from the MS COCO dataset [25]. As the MS COCO dataset is not intended to capture text, COCO contains many occluded or low-resolution texts.

RCTW [42] is created for **R**eading **C**hinese **T**ext in the **W**ild competition. Thus many are Chinese text.

Uber-Text (Uber) [62] is collected from Bing Maps Streetside. Many are house number, and some are text on signboards.

(e) Year 2019 (ArT, LSVT, MLT19, ReCTS):

ArT [6] is created to recognize **Ar**bitrary-shaped **T**ext. Many are perspective or curved texts. It also includes Totaltext [7] and CTW1500 [28], which contain many rotated or curved texts.

LSVT [47, 46] is a **L**arge-scale **S**treet **V**iew **T**ext dataset, collected from streets in China, and thus many are Chinese text.

MLT19 [34] is created to recognize **M**ulti-**L**ingual **T**ext. It consists of seven languages: Arabic, Latin, Chinese, Japanese, Korean, Bangla, and Hindi.

ReCTS [61] is created for the **Re**ading **C**hinese **T**ext on **S**ignboard competition. It contains many irregular texts arranged in various layouts or written with unique fonts.

	Dataset	Conf.	Year	# of word boxes: Original	# of word boxes: Processed
Real labeled datasets (Real-L)					
(a)	SVT [51]	ICCV	2011	257	231
(b)	IIIT [31]	BMVC	2012	2,000	1,794
	IC13 [19]	ICDAR	2013	848	763
(c)	IC15 [18]	ICDAR	2015	4,468	3,710
(d)	COCO [49]	arXiv	2016	43K	39K
	RCTW [42]	ICDAR	2017	65K	8,186
	Uber [62]	CVPRW	2017	285K	92K
(e)	ArT [6]	ICDAR	2019	50K	29K
	LSVT [47]	ICDAR	2019	383K	34K
	MLT19 [34]	ICDAR	2019	89K	46K
	ReCTS [61]	ICDAR	2019	147K	23K
	Total	–	–	1.1M	276K
Real unlabeled datasets (Real-U)					
	Book32 [14]	arXiv	2016	3.9M	3.7M
	TextVQA [44]	CVPR	2019	551K	463K
	ST-VQA [3]	ICCV	2019	79K	69K
	Total	–	–	4.6M	4.2M

Table 1: Number of **training set** in public real datasets.

3.2. Real Unlabeled Datasets

We consolidate three unlabeled datasets for semi- and self-supervised learning. They contain scene images and do not have word region annotation. Thus, we use a pretrained text detector to crop words. We use the detector [27], which is not trained on synthetic data and won the ReCTS competition[1]. Details are in the supplementary materials.

Book32 [14] is collected from Amazon Books, and consists of 208K book cover images in 32 categories. It contains many handwritten or curved texts.

TextVQA [44] is created for text-based visual question answering. It consists of 28K OpenImage V3 [21] images from categories such as "billboard" and "traffic sign."

ST-VQA [3] is created for scene text-based visual question answering. It includes IC13, IC15, and COCO, and thus we excluded them.

[1] https://rrc.cvc.uab.es/?ch=12&com=evaluation&task=3

3.3. Preprocessing Real Datasets

We conduct following processes before using real data:

Excluding duplication between datasets Some well-known datasets (ICDAR03 (IC03) [29], MLT17 [35], and TotalText [7]) are excluded because they are included in other datasets: IC13 inherits most of IC03, MLT19 includes MLT17, and ArT includes TotalText. Also, CT and ArT have 122 duplicated word boxes, and we exclude them.

Collecting only English words Some datasets are made for Chinese text recognition (RCTW, ArT, LSVT, ReCTS) or multilingual text recognition (MLT19). Thus they contain languages other than English. We only use words which consist of alphanumeric characters and symbols.

Excluding don't care symbol Some texts have "*" or "#", which denotes "do not care about the text" or "characters hard to read." We exclude the texts containing them.

Excluding vertical or $\pm$ 90 degree rotated texts Some datasets such as Uber-Text [62] contain many vertical texts or $\pm$ 90° rotated texts. We mainly focus on horizontal texts and thus exclude vertical texts. The images whose texts have more than two characters and whose height is greater than the width are excluded. For unlabeled data, the images whose height is greater than the width are excluded.

Splitting training set to make validation set Most real datasets do not have validation set. Thus we split the training set of each dataset into training and validation sets.

In addition, we exclude texts longer than 25 characters following common practice [1].

4. STR With Fewer Labels

In this section, we describe the STR models and semi- and self-supervised learning. Although real data has increased as mentioned in §3.1, real data is still fewer than synthetic data at about 1.7% of synthetic data. To compensate for the low amount of data, we introduce a semi- and self-supervised learning framework to improve the STR with fewer labels. This is inspired by other computer vision tasks with fewer labels (high-fidelity image generation [30] and ImageNet classification [59]).

4.1. STR Model Framework

According to [1], STR is performed in four stages:

1. **Transformation (Trans.)**: normalizes the perspective or curved text into a horizontal text. This is generally done by the Spatial Transformer Network (STN) [16].
2. **Feature extraction (Feat.)**: extracts visual feature representation from the input image. This is generally performed by a module composed of convolutional neural networks (CNNs), such as VGG [43] and ResNet [13].

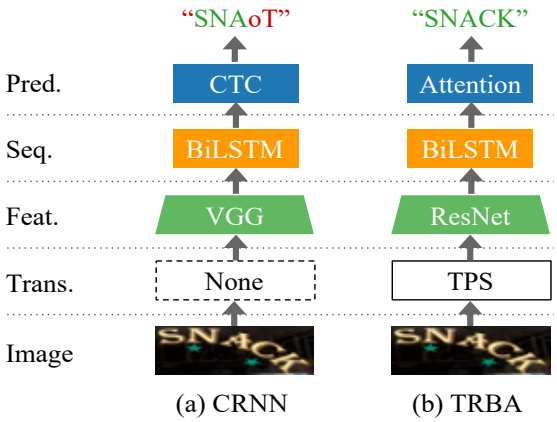


Figure 4: Illustration of CRNN [40] and TRBA [1].

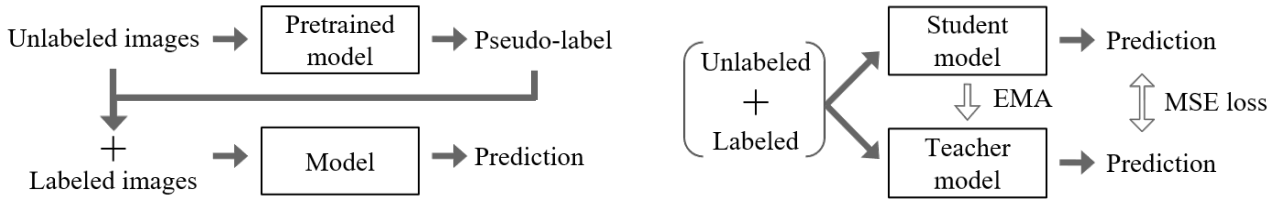
3. **Sequence modeling (Seq.)**: converts visual features to contextual features that capture context in the sequence of characters. This is generally done by BiLSTM [10].
4. **Prediction (Pred.)**: predicts the character sequence from contextual features. This is generally done by a CTC [9] decoder or attention mechanism [2].

For our experiments, we adopt two widely-used models from the STR benchmark [1]: CRNN[40] and TRBA[1], as illustrated in Figure 4. CRNN consists of None, VGG, BiLSTM, and CTC for each stage. CRNN has lower accuracy than state-of-the-art methods, but CRNN is widely chosen for practical usage because it is fast and lightweight. TRBA consists of a thin-plate spline [4] transform-based STN (**T**PS), **R**esNet, **Bi**LSTM, and **A**ttention for each stage. As TRBA uses ResNet and attention mechanism, it is larger and slower than CRNN but has higher accuracy.

4.2. Semi-Supervised Learning

Recently, various semi-supervised methods have been proposed and improved the performance with unlabeled data, particularly in image classification tasks [22, 48, 32, 59]. Since large synthetic data is used for STR to compensate for the lack of data instead of using unlabeled data, studies on training STR with unlabeled data are rare. To the best of our knowledge, there is only one study that uses unlabeled data for the STR benchmark [17]. We introduce two simple yet effective semi-supervised methods for STR.

Pseudo-Label (PL) [22] is a simple approach that uses unlabeled data. The process is as follows: 1) Train the model on labeled data. 2) Using the trained model, make predictions on unlabeled data and use them as pseudolabels. 3) Combine labeled and pseudolabeled data, and retrain the model on them. Figure 5a illustrates PL. Concurrent work [17] also uses PL on the Book32 dataset. The researchers combine pseudolabeled and synthetic data, and use them as a training set.



(a) Pseudo-Label (PL)

(b) Mean Teacher (MT)

Figure 5: Illustration of Pseudo-Label [22] and mean teacher [48]. +, EMA, and MSE denote union of labeled and unlabeled data, exponential moving average, and mean squared error, respectively.

Mean Teacher (MT) [48] is a method that uses consistency regularization. The process is as follows: 1) Prepare a model and a copy of the model. 2) Use the former as a student model and the latter as a teacher model. 3) Apply two random augmentations η and η' on the same mini-batch. 4) Input the former to the student model and the latter to the teacher model. 5) Calculate the mean squared error loss on their outputs. 6) Update the student model. 7) Update the teacher model with an exponential moving average (EMA) of the student model. Figure 5b illustrates MT.

4.3. Self-Supervised Learning

Recently, self-supervised methods have shown promising results in computer vision tasks [8, 12, 5]. Self-supervised learning is generally conducted in two stages: 1) Pretrain the model with a surrogate (pretext) task. 2) Using pretrained weights for initialization, train the model for the main task. The pretext task is generally conducted on unlabeled data, and by learning a pretext task, the model obtains better feature maps for the main task. In this study, we investigated two widely-used methods, RotNet [8] and MoCo [12].

RotNet [8] predicts the rotation of images as a pretext task. The task is simple: rotate input images at 0, 90, 180, and 270 degrees, and the model recognizes the rotation applied to the image.

Momentum Contrast (MoCo) [12] is a contrastive learning method that can be applied to various pretext tasks. Following [12], we use an instance discrimination task [54] as a pretext task. The task consists of the following steps: 1) Prepare a model and a copy of the model. 2) Use the former as a query encoder and the latter as a momentum encoder. 3) Apply two random augmentations η and η' on the same mini-batch. 4) Input the former into a query encoder to make encoded queries $\boldsymbol{q}$. 5) Input the latter into a momentum encoder to make encoded keys $\boldsymbol{k}$. 6) Calculate the contrastive loss, called InfoNCE [36], on pairs of a query q and a key k. For a pair of q and k, if they are derived from the same image, assign a positive, otherwise negative label. 7) Update the query encoder. 8) Update the momentum encoder with an moving average of the query encoder.

5. Experiment and Analysis

In this section, we present the results of our main experiments using STR with fewer real labels with a semi- and self-supervised learning framework.

5.1. Implementation Detail

We summarize our experimental settings. More details of our settings are in our supplementary materials.

Model and training strategy We use the code of the STR benchmark repository[2][1], and use CRNN and TRBA as described in §4.1. We use the Adam [20] optimizer and the one-cycle learning rate scheduler [45] with a maximum learning rate of 0.0005. The number of iterations is 200K, and the batch size is 128. As shown in Table 1, the number of training sets is imbalanced over the datasets. To overcome the data imbalance, we sample the same number of data from each dataset to make a mini-batch.

Dataset and model selection To train models on real data, we use the union of 11 real datasets (Real-L) listed in Table 1. After preprocessing described in §3.3, we have 276K training and 63K validation sets, and use them for training. In all our experiments, we use the 63K validation set for model selection: select the model with the best accuracy on the validation set for evaluation. We validate the model every 2,000 iterations, as in [1]. To train models on synthetic data, we use the union of MJ (9M) and ST (7M). For semi- and self-supervised learning, we use the union of 3 real unlabeled datasets (Real-U, 4.2M) listed in Table 1.

Evaluation metric We use word-level accuracy on six benchmark datasets, as described in §2.2. The accuracy is calculated only on the alphabet and digits, as done in [41]. We calculate the total accuracy for comparison, which is the accuracy of the union of six benchmark datasets (7,672 in total). In our study, accuracy indicates **total accuracy**. For all experiments, we run three trials with different initializations and report averaged accuracies.

[2]https://github.com/clovaai/deep-text-recognition-benchmark

	Method	Year	Train data	Dataset name and # of data						
				IIIT 3000	SVT 647	IC13 1015	IC15 2077	SP 645	CT 288	Total 7672
Reported results	ASTER [41]	2018	MJ+ST	93.4	89.5	91.8	76.1	78.5	79.5	86.4
	ESIR [60]	2019	MJ+ST	93.3	90.2	91.3	76.9	79.6	83.3	86.8
	MaskTextSpotter [24]	2019	MJ+ST	**95.3**	91.8	**95.3**	78.2	83.6	88.5	89.1
	ScRN [55]	2019	MJ+ST	94.4	88.9	93.9	78.7	80.8	87.5	88.2
	DAN [52]	2020	MJ+ST	94.3	89.2	93.9	74.5	80.0	84.4	86.9
	TextScanner [50]	2020	MJ+ST	93.9	90.1	92.9	79.4	**84.3**	83.3	88.3
	SE-ASTER [38]	2020	MJ+ST	93.8	89.6	92.8	80.0	81.4	83.6	88.2
	RobustScanner [57]	2020	MJ+ST	**95.3**	88.1	94.8	77.1	79.5	**90.3**	88.2
	PlugNet [33]	2020	MJ+ST	94.4	**92.3**	95.0	**82.2**	**84.3**	85.0	**89.8**
Our experiment	CRNN-Original [40]	2015	MJ	78.2	80.8	86.7	–	–	–	–
	CRNN-Baseline-synth		MJ+ST	84.3	78.9	88.8	61.5	64.8	61.3	75.8
	CRNN-Baseline-real		Real-L	83.5	75.5	86.3	62.2	60.9	64.7	74.8
	CRNN-PR		Real-L+U	89.8	84.3	90.9	73.1	74.6	82.3	83.4
	TRBA-Original [1]	2019	MJ+ST	87.9	87.5	92.3	71.8	79.2	74.0	82.8
	TRBA-Baseline-synth		MJ+ST	92.1	88.9	93.1	74.7	79.5	78.2	85.7
	TRBA-Baseline-real		Real-L	93.5	87.5	92.6	76.0	78.7	86.1	86.6
	TRBA-PR		Real-L+U	94.8	91.3	94.0	80.6	82.7	88.1	89.3

Table 2: Accuracy of STR models on six benchmark datasets. We show the results reported in original papers. We present our results of CRNN and TRBA: Reproduced models (Baseline-synth), models trained only on real labels (Baseline-real), and our best setting (PR, combination of Pseudo-Label and RotNet). TRBA-PR has a competitive performance with state-of-the-art models. MJ, ST, Real-L, and Real-L+U denote MJSynth, SynthText, union of 11 real labeled datasets, and union of 11 real labeled and 3 unlabeled datasets in Table 1, respectively. In each column, top accuracy is shown in **bold**.

5.2. Comparison to State-of-the-Art Methods

Table 2 lists the results of state-of-the-art methods and our experiments. For a fair comparison, we list the methods that use **only MJ and ST** for training, and evaluate six benchmarks: IIIT, SVT, IC13-1015, IC15-2077, SP, and CT.

Our reproduced models (Baseline-synth) has higher accuracies than in the original paper because we use different settings such as larger datasets (8M to 16M for CRNN and 14.4M to 16M for TRBA), different optimizer (Adam instead of AdaDelta [58]), and learning rate scheduling. Baseline-real is the model only trained on 11 real datasets. CRNN-Baseline-real has an accuracy close to that of CRNN-Synth (74.8% to 75.8%), and TRBA-Baseline-real surpasses TRBA-Synth (86.6% over 85.7%).

TRBA with our best setting (TRBA-PR) trained on only real data has a competitive performance of 89.3% with state-of-the-art methods. PR denotes the combination of Pseudo-Label and RotNet. PR improves Baseline-real by +8.6% for CRNN and +2.7% for TRBA, and results in higher accuracy than Baseline-synth. In the following sections, we analyze our best setting with ablation studies.

5.3. Training Only on Real Labeled Data

In this section, we present the results of training STR models only on real labeled data.

Accuracy depending on dataset increment Table 1 shows the increment of real data, and Figure 1 shows the accuracy improvement. For 2019, when the number of the real training set is 276K, the accuracy of CRNN and TRBA trained only on real labeled data is close to that of synthetic data. This indicates that we have enough real labeled data to train STR models satisfactorily, although the real labeled data is **only 1.7%** of the synthetic data.

These results indicate that we need at least from 146K (Year 2017) to 276K (Year 2019) real data for training STR models. However, according to [1], the diversity of the training set can be more important than the number of training sets. We use 11 datasets, which denotes high diversity, and thus we cannot simply conclude with "276K is enough."

To investigate the significance of real data, we also conduct an experiment that uses only 1.74% of synthetic data (277K images), which is a similar amount to our real data. This results in approximately 10% lower accuracy than Baseline-real (65.1% vs. 74.8% for CRNN and 75.9% vs. 86.6% for TRBA). This indicates that real data is far more significant than synthetic data.

Augmentation	CRNN	TRBA
Baseline-real	74.8	86.6
+ Blur	75.7	86.8
+ Crop	78.8	87.1
+ Rot	79.5	86.2
+ Blur + Crop	79.1	**87.5**
+ Blur + Rot	79.5	86.1
+ Crop + Rot	**80.0**	86.7
+ Blur + Crop + Rot	78.9	86.6
Baseline-synth	75.8	85.7
+ *Aug.*	73.4	85.2

Table 3: Improvement by simple data augmentations. *Aug.* denotes the best augmentation setting in our experiments.

Improvement by simple data augmentations Since our goal is to train an STR model with fewer labels, to compensate for them, we find effective data augmentations. Most STR methods do not use data augmentations [40, 41, 60, 1, 52, 38, 57, 33]. We suppose that they do not use data augmentation because the synthetic data already includes augmented data. In that case, if we apply further data augmentation above on already augmented data, then the results can be worse. For example, applying a 45° rotation on the synthetic text, which was already rotated 45°, makes horizontal text to 90° rotated text. This can produce worse results.

However, if we use real data that do not contain data augmentations, then we can easily make improvements by data augmentation. We investigate simple data augmentations. Specifically, we use the Gaussian blur (Blur) to cope with blurry texts. We use a high ratio cropping (Crop), which slightly cuts the top, bottom, left, and right ends of the text, making STR models robust, and a rotation (Rot) for rotated, perspective, or curved texts. The intensity of each augmentation affects the performance. We find the best intensities for them. Table 3 shows the results of augmentations with best intensity and their combination. The experiments with varying intensities of augmentations are in the supplementary materials.

Combinations of simple augmentations successfully improves the STR models. For CRNN, the best setting (*Aug.*) is the combination of Crop and Rot, which improves the accuracy by 5.2% from Baseline-real. For TRBA, the best setting (*Aug.*) is the combination of Blur and Crop, which improves the accuracy by 0.9% from Baseline-real.

We also apply the *Aug.* to Baseline-synth, and the accuracy decreases. We presume that the combination of already augmented data in synthetic data and *Aug.* can be harmful to the performance. For a similar case in our controlled experiments, the combination of Blur, Crop, and Rot has a lower accuracy than the combination of Crop and Rot. These results indicate that the addition of augmentation can be harmful to the performance depending on the elements.

Method	CRNN	TRBA
Baseline-real + *Aug.*	80.0	87.5
+ PL	82.8 (+2.8)	89.2 (+1.7)
+ MT	79.8 (-0.2)	87.1 (-0.4)
+ RotNet	81.3 (+1.3)	87.5
+ MoCo	80.8 (+0.8)	86.7 (-0.8)
+ PL + RotNet	**83.4 (+3.4)**	**89.3 (+1.8)**

Table 4: Ablation study on semi- and self-supervised methods. Gaps of at least 1.0 points are shown in green.

5.4. Semi- and Self-Supervised Learning

In addition to data augmentations, we further improve STR models by using unlabeled data (Real-U, listed in Table 1) with semi- and self-supervised methods, as described in §4. Table 4 shows the results.

Pseudo-Label (PL) and Mean Teacher (MT): PL boosts the accuracy by 2.8% for CRNN and 1.7% for TRBA. MT decreases the accuracy by -0.2% for CRNN and -0.4% for TRBA.

RotNet and MoCo: Following the common practice that pretrains CNN part of the model [8, 12], we pretrain VGG for CRNN and ResNet for TRBA. We find that when we pretrain both TPS and ResNet in TRBA, the accuracy decreases sharply: -11.8% with RotNet and -5.9% with MoCo. Thus, we only pretrain ResNet in TRBA.

For CRNN, RotNet and MoCo improve the accuracy by 1.3% and 0.8%, respectively. RotNet is slightly more effective than MoCo. For TRBA, RotNet marginally improves the accuracy by +0.09% (87.45% to 87.54%) and MoCo decreases the accuracy by -0.8%.

Combination of semi- and self-supervised methods The semi- and self-supervised methods in our experiments are independent, and thus we can combine them for further improvement. We select PL and RotNet for the combination because they have better accuracy than MT and MoCo, respectively. The PL method can be improved with a more accurate pretrained model to predict pseudolabels. We use RotNet as the more accurate model. Specifically, PL and RotNet are combined as follows: 1) Initialize the models with weights trained by the pretext task of RotNet. 2) Use RotNet to predict pseudolabels. Table 4 shows the results of the combination.

The combination of PL and RotNet has higher accuracy than solely using PL or RotNet. This successfully improves the accuracy by +3.4% from Baseline-real with *Aug.* for CRNN and +1.8% for TRBA. Our best setting (PR) is the combination of *Aug.*, PL, and RotNet.

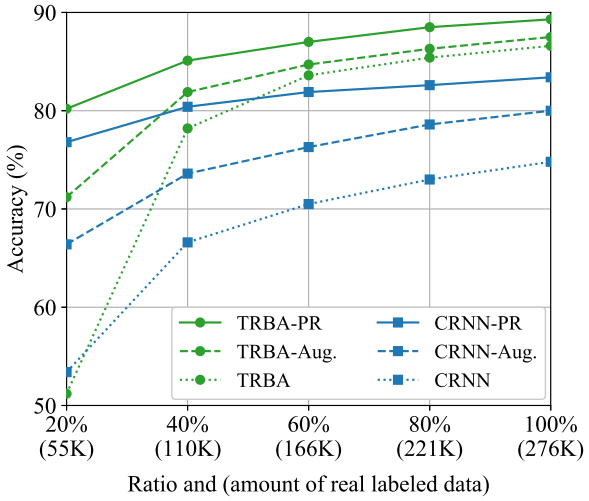


Figure 6: Accuracy vs. amount of real labeled data.

5.5. Varying Amount of Real Labeled Data

Although Figure 1 shows the accuracy depending on dataset increment, the results are entangled with two factors: the amount of labeled data and the diversity of datasets. We investigate the effect of the amount of labeled data by proportionally reducing each labeled dataset while maintaining the diversity of datasets (11 datasets). The amount of unlabeled data is fixed. Figure 6 shows the results.

Baseline-real is drastically dropped -13.2% in accuracy for CRNN and -27.0% for TRBA, with varying data ratios of 40% to 20%. This shows that an accuracy cliff would appear here. When our best augmentation setting (*Aug.*) is applied, the accuracy improves fairly, especially with ratios of 20%, by +13.0% for CRNN and +20.0% for TRBA.

PR with unlabeled data can substitute over 221K labeled data for CRNN and 110K labeled data for TRBA. CRNN-PR with a ratio of 20% exceeds Baseline-real with a ratio of 100% by 2.0%. TRBA-PR with a ratio of 60% exceeds Baseline-real with a ratio of 100% by 0.4%.

The diversity of datasets can be more important than the amount of labeled data. Comparing the Baseline with ratio 40% (110K) to Year 2017 (146K) in Figure 1, while the former has less data than the latter, the former has higher diversity than the latter (11 datasets vs. 7 datasets). The former has higher accuracy than the latter: 66.6% vs. 65.1% for CRNN and 78.2% vs. 75.1% for TRBA.

5.6. Training on Both Synthetic and Real Data

In real scenarios, there is a case in which we have large synthetic data for the general domain and only fewer real data for the target domain. We investigate if our best setting (PR) is also useful for this case by comparing other options.

Fine-tuning on real data Transfer learning with simple fine-tuning is a feasible option for such a case. We conduct training STR models on large synthetic data (MJ and ST, 16M) and then fine-tuning on fewer real data (Real-L, 276K) for 40K iterations.

Method	Train Data	CRNN	TRBA
Fine-tuning			
Baseline-synth	MJ+ST	75.8	85.7
+ FT	Real-L	82.1	90.0
+ FT w/PR	Real-L+U	76.6	87.5
From scratch			
Baseline-real	Real-L	74.8	86.6
Baseline-real	Real-L+MJ+ST	79.8	89.1
PR	Real-L+U	83.4	89.3
PR	Real-L+U+MJ+ST	84.2	90.0

Table 5: Training on both synthetic and real data.

Training from scratch Another option is training STR models on both of them from scratch. We use the union of 11 real labeled and 2 synthetic datasets as a training set.

Table 5 shows the results. Fine-tuning on real labeled data improves the accuracy by +6.3% for CRNN and +4.3% for TRBA. Unexpectedly, fine-tuning with PR increases the accuracy (+0.8% for CRNN and +1.8% for TRBA) but has lower accuracy than fine-tuning only on real labeled data (76.6% vs. 82.1% for CRNN and 87.5% vs. 90.0% for TRBA). This indicates that using semi- and self-supervised methods during fine-tuning can be harmful.

PR has higher accuracy than Baseline-real with synthetic data. This shows that we can substitute the synthetic data with semi- and self-supervised methods that use unlabeled data. For CRNN, PR with synthetic data has higher accuracy than the other settings. This indicates that PR can be useful for training STR models when both large synthetic data and fewer real data are available.

6. Conclusion

Since STR models have been trained on large synthetic data, training STR models on fewer real labels (STR with fewer labels) has not been sufficiently studied. In this paper, we have focused on STR with fewer labels. STR with fewer labels is considered difficult because there are only thousands of real data, resulting in low accuracy. However, this is no longer the case. We have shown that public real data has been accumulated over the years. Although accumulated real data is only 1.7% of the synthetic data, we can train STR models sufficiently by using it. We have further improved the performance by using simple data augmentations and introducing semi- and self-supervised methods with millions of real unlabeled data. This work is a stepping stone toward STR with fewer labels, and we hope this work will facilitate future work on this topic.

References

[1] 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*, 2019. 1, 2, 4, 5, 6, 7

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *ICLR*, 2015. 4

[3] Ali Furkan Biten, Ruben Tito, Andres Mafla, Lluis Gomez, Marçal Rusinol, Ernest Valveny, CV Jawahar, and Dimosthenis Karatzas. Scene text visual question answering. In *ICCV*, 2019. 3

[4] Fred L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *TPAMI*, 1989. 4

[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020. 5

[6] Chee Kheng Chng, Yuliang Liu, Yipeng Sun, Chun Chet Ng, Canjie Luo, Zihan Ni, ChuanMing Fang, Shuaitao Zhang, Junyu Han, Errui Ding, et al. Icdar2019 robust reading challenge on arbitrary-shaped text-rrc-art. In *ICDAR*, 2019. 3

[7] Chee Kheng Ch'ng, Chee Seng Chan, and Chenglin Liu. Total-text: Towards orientation robustness in scene text detection. *IJDAR*, 2020. 3, 4

[8] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *ICLR*, 2018. 5, 7

[9] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *ICML*, 2006. 4

[10] Alex Graves, Marcus Liwicki, Santiago Fernández, Roman Bertolami, Horst Bunke, and Jürgen Schmidhuber. A novel connectionist system for unconstrained handwriting recognition. *TPAMI*, 2009. 4

[11] Ankush Gupta, Andrea Vedaldi, and Andrew Zisserman. Synthetic data for text localisation in natural images. In *CVPR*, 2016. 2

[12] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020. 5, 7

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 4

[14] Brian Kenji Iwana, Syed Tahseen Raza Rizvi, Sheraz Ahmed, Andreas Dengel, and Seiichi Uchida. Judging a book by its cover. *arXiv:1610.09204*, 2016. 3

[15] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Synthetic data and artificial neural networks for natural scene text recognition. In *Workshop on Deep Learning, NeurIPS*, 2014. 1, 2

[16] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *NeurIPS*, 2015. 4

[17] Klára Janoušková, Jiri Matas, Lluis Gomez, and Dimosthenis Karatzas. Text recognition–real world data and where to find them. *arXiv:2007.03098*, 2020. 4

[18] Dimosthenis Karatzas, Lluis Gomez-Bigorda, Anguelos Nicolaou, Suman Ghosh, Andrew Bagdanov, Masakazu Iwamura, Jiri Matas, Lukas Neumann, Vijay Ramaseshan Chandrasekhar, Shijian Lu, et al. Icdar 2015 competition on robust reading. In *ICDAR*, 2015. 2, 3

[19] Dimosthenis Karatzas, Faisal Shafait, Seiichi Uchida, Masakazu Iwamura, Lluis Gomez i Bigorda, Sergi Robles Mestre, Joan Mas, David Fernandez Mota, Jon Almazan Almazan, and Lluis Pere De Las Heras. Icdar 2013 robust reading competition. In *ICDAR*, 2013. 2, 3

[20] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 5

[21] Ivan Krasin, Tom Duerig, Neil Alldrin, Vittorio Ferrari, Sami Abu-El-Haija, Alina Kuznetsova, Hassan Rom, Jasper Uijlings, Stefan Popov, Andreas Veit, Serge Belongie, Victor Gomes, Abhinav Gupta, Chen Sun, Gal Chechik, David Cai, Zheyun Feng, Dhyanesh Narayanan, and Kevin Murphy. Openimages: A public dataset for large-scale multi-label and multi-class image classification. *Dataset available from https://github.com/openimages*, 2017. 3

[22] Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, 2013. 4, 5

[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*, 2019. 1

[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. *TPAMI*, 2019. 1, 6

[25] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 3

[26] Ron Litman, Oron Anschel, Shahar Tsiper, Roee Litman, Shai Mazor, and R Manmatha. Scatter: selective context attentional scene text recognizer. In *CVPR*, 2020. 1

[27] Yuliang Liu, Tong He, Hao Chen, Xinyu Wang, Canjie Luo, Shuaitao Zhang, Chunhua Shen, and Lianwen Jin. Exploring the capacity of sequential-free box discretization network for omnidirectional scene text detection. *arXiv:1912.09629*, 2019. 3

[28] Yuliang Liu, Lianwen Jin, Shuaitao Zhang, Canjie Luo, and Sheng Zhang. Curved scene text detection via transverse and longitudinal sequence connection. *Pattern Recognition*, 2019. 3

[29] Simon M Lucas, Alex Panaretos, Luis Sosa, Anthony Tang, Shirley Wong, and Robert Young. Icdar 2003 robust reading competitions. In *ICDAR*, 2003. 4

[30] Mario Lucic, Michael Tschannen, Marvin Ritter, Xiaohua Zhai, Olivier Bachem, and Sylvain Gelly. High-fidelity image generation with fewer labels. In *ICML*, 2019. 4

[31] Anand Mishra, Karteek Alahari, and CV Jawahar. Scene text recognition using higher order language priors. In *BMVC*, 2012. 2, 3

[32] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *TPAMI*, 2018. 4

[33] 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*, 2020. 1, 6, 7

[34] Nibal Nayef, Yash Patel, Michal Busta, Pinaki Nath Chowdhury, Dimosthenis Karatzas, Wafa Khlif, Jiri Matas, Umapada Pal, Jean-Christophe Burie, Cheng-lin Liu, et al. Icdar2019 robust reading challenge on multi-lingual scene text detection and recognition—rrc-mlt-2019. In *ICDAR*, 2019. 3

[35] Nibal Nayef, Fei Yin, Imen Bizid, Hyunsoo Choi, Yuan Feng, Dimosthenis Karatzas, Zhenbo Luo, Umapada Pal, Christophe Rigaud, Joseph Chazalon, et al. Icdar2017 robust reading challenge on multi-lingual scene text detection and script identification-rrc-mlt. In *ICDAR*, 2017. 4

[36] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv:1807.03748*, 2018. 5

[37] Trung Quy Phan, Palaiahnakote Shivakumara, Shangxuan Tian, and Chew Lim Tan. Recognizing text with perspective distortion in natural scenes. In *ICCV*, 2013. 2

[38] Zhi Qiao, Yu Zhou, Dongbao Yang, Yucan Zhou, and Weiping Wang. Seed: Semantics enhanced encoder-decoder framework for scene text recognition. In *CVPR*, 2020. 1, 6, 7

[39] Anhar Risnumawan, Palaiahankote Shivakumara, Chee Seng Chan, and Chew Lim Tan. A robust arbitrary text detection system for natural scene images. *ESWA*, 2014. 2

[40] 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. *TPAMI*, 2016. 1, 2, 4, 6, 7

[41] Baoguang Shi, Mingkun Yang, Xinggang Wang, Pengyuan Lyu, Cong Yao, and Xiang Bai. Aster: An attentional scene text recognizer with flexible rectification. *TPAMI*, 2018. 1, 5, 6, 7

[42] Baoguang Shi, Cong Yao, Minghui Liao, Mingkun Yang, Pei Xu, Linyan Cui, Serge Belongie, Shijian Lu, and Xiang Bai. Icdar2017 competition on reading chinese text in the wild (rctw-17). In *ICDAR*, 2017. 3

[43] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 4

[44] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *CVPR*, 2019. 3

[45] Leslie N. Smith and Nicholay Topin. Super-convergence: very fast training of neural networks using large learning rates. *AI/ML for MDO*, 2019. 5

[46] Yipeng Sun, Jiaming Liu, Wei Liu, Junyu Han, Errui Ding, and Jingtuo Liu. Chinese street view text: Large-scale chinese text reading with partially supervised learning. In *ICCV*, 2019. 3

[47] Yipeng Sun, Zihan Ni, Chee-Kheng Chng, Yuliang Liu, Canjie Luo, Chun Chet Ng, Junyu Han, Errui Ding, Jingtuo Liu, Dimosthenis Karatzas, et al. Icdar 2019 competition on large-scale street view text with partial labeling-rrc-lsvt. In *ICDAR*, 2019. 3

[48] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *NeurIPS*, 2017. 4, 5

[49] Andreas Veit, Tomas Matera, Lukas Neumann, Jiri Matas, and Serge Belongie. Coco-text: Dataset and benchmark for text detection and recognition in natural images. *arXiv:1601.07140*, 2016. 3

[50] Zhaoyi Wan, Mingling He, Haoran Chen, Xiang Bai, and Cong Yao. Textscanner: Reading characters in order for robust scene text recognition. In *AAAI*, 2020. 1, 6

[51] Kai Wang, Boris Babenko, and Serge Belongie. End-to-end scene text recognition. In *ICCV*, 2011. 2, 3

[52] 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*, 2020. 1, 6, 7

[53] Wenjia Wang, Enze Xie, Xuebo Liu, Wenhai Wang, Ding Liang, Chunhua Shen, and Xiang Bai. Scene text image super-resolution in the wild. In *ECCV*, 2020. 1

[54] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, 2018. 5

[55] 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*, 2019. 1, 6

[56] 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*, 2020. 1

[57] Xiaoyu Yue, Zhanghui Kuang, Chenhao Lin, Hongbin Sun, and Wayne Zhang. Robustscanner: Dynamically enhancing positional clues for robust text recognition. In *ECCV*, 2020. 1, 6, 7

[58] Matthew D Zeiler. Adadelta: an adaptive learning rate method. *arXiv:1212.5701*, 2012. 6

[59] Xiaohua Zhai, Avital Oliver, Alexander Kolesnikov, and Lucas Beyer. S4l: Self-supervised semi-supervised learning. In *ICCV*, 2019. 4

[60] Fangneng Zhan and Shijian Lu. Esir: End-to-end scene text recognition via iterative image rectification. In *CVPR*, 2019. 1, 6, 7

[61] Rui Zhang, Yongsheng Zhou, Qianyi Jiang, Qi Song, Nan Li, Kai Zhou, Lei Wang, Dong Wang, Minghui Liao, Mingkun Yang, et al. Icdar 2019 robust reading challenge on reading chinese text on signboard. In *ICDAR*, 2019. 3

[62] Ying Zhang, Lionel Gueguen, Ilya Zharkov, Peter Zhang, Keith Seifert, and Ben Kadlec. Uber-text: A large-scale dataset for optical character recognition from street-level imagery. In *Scene Understanding Workshop, CVPR*, 2017. 3, 4