Appendix for
Revisiting Scene Text Recognition: A Data Perspective

Qing Jiang , Jiapeng Wang , Dezhi Peng , Chongyu Liu , Lianwen Jin*
South China University of Technology
mountchicken@outlook.com , scutjpwang@foxmail.com, eedzpeng@mail.scut.edu.cn
liuchongyu1996@gmail.com , eelwjin@scut.edu.cn
https://union14m.github.io/

1. Unrecognized Samples in Common Benchmarks

In Fig. 2, we show four types of images in the six common benchmarks that are not correctly recognized by the ensemble of 13 STR models. Specifically, for human unrecognizable images, we adopt the following criteria for adjudication: We recruit five human experts, and each of them submits three possible predictions for each text image. If all five experts failed to recognize a text image (i.e., 15 predictions in total are incorrect), we regard it as a human unrecognizable sample. The majority of these human unrecognizable samples exhibit high levels of blurriness and low resolution. Furthermore, upon further examination of the 16.8% of samples that are classified as "other", we can observe that many of them fall under the categories of the seven challenges that we have discussed before, such as curve text, multi-words text, and artistic text.

2. More Details of Union14M

2.1. Construction of Union14M-U

In order to gather a vast number of high-quality unlabeled text images, we utilize three scene text detectors: DBNet++[1] [14], BDN[2] [15], and EAST[3] [32]. We apply these detectors to three large datasets: Book32[6], OpenImages[11], and Conceptual Captions (CC)[23]. However, directly using the results of these detectors is suboptimal due to the presence of many false positive results produced by different detectors (e.g., in Fig. 1, the rear tire of the police car is detected as a text region by two detectors). While missing detections can be tolerated given a large amount of data, false detections are undesirable as they may introduce noise for subsequent self-supervised learning. To address this issue, we adopt a simple Intersection over Union (IoU) voting strategy to filter out false detections. Specifically, we identify regions where the detected polygons of the three detectors have an IoU larger than 0.7 with respect to each other, and then we use the minimum axis-aligned rectangle of the three detected polygons as the final prediction. Additionally, when selecting images from OpenImages to construct Union14M-U, we exclude images with the same image ID in HierText [16], TextOCR [27], and InterOCR [10] since they have already been used in Union14M-L. Using this strategy, we obtain 10.6 million high-quality text instances in Union14M-U. It is noteworthy that all three detectors are trained on a singular dataset (DBNet++ and EAST are trained on ICDAR2015 [9], BDN is

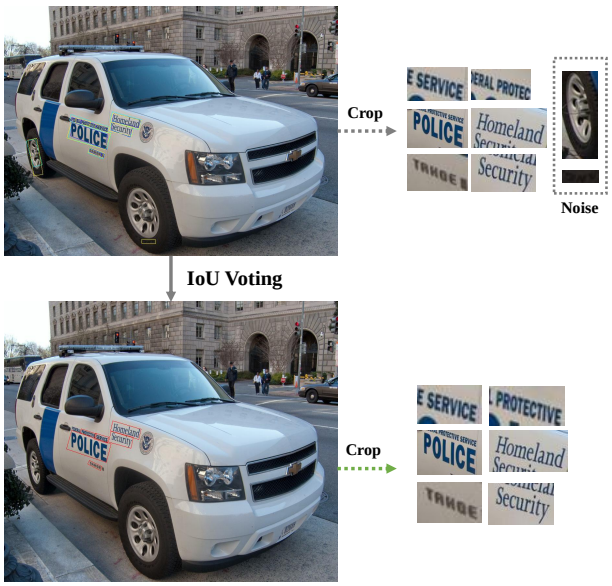

Figure 1. An illustration of our IoU voting strategy for collecting text instances.

*Corresponding author
[1]https://github.com/open-mmlab/mmocr/tree/main/configs/textdet/dbnetpp
[2]https://github.com/Yuliang-Liu/Box_Discretization_Network
[3]https://github.com/SakuraRiven/EAST

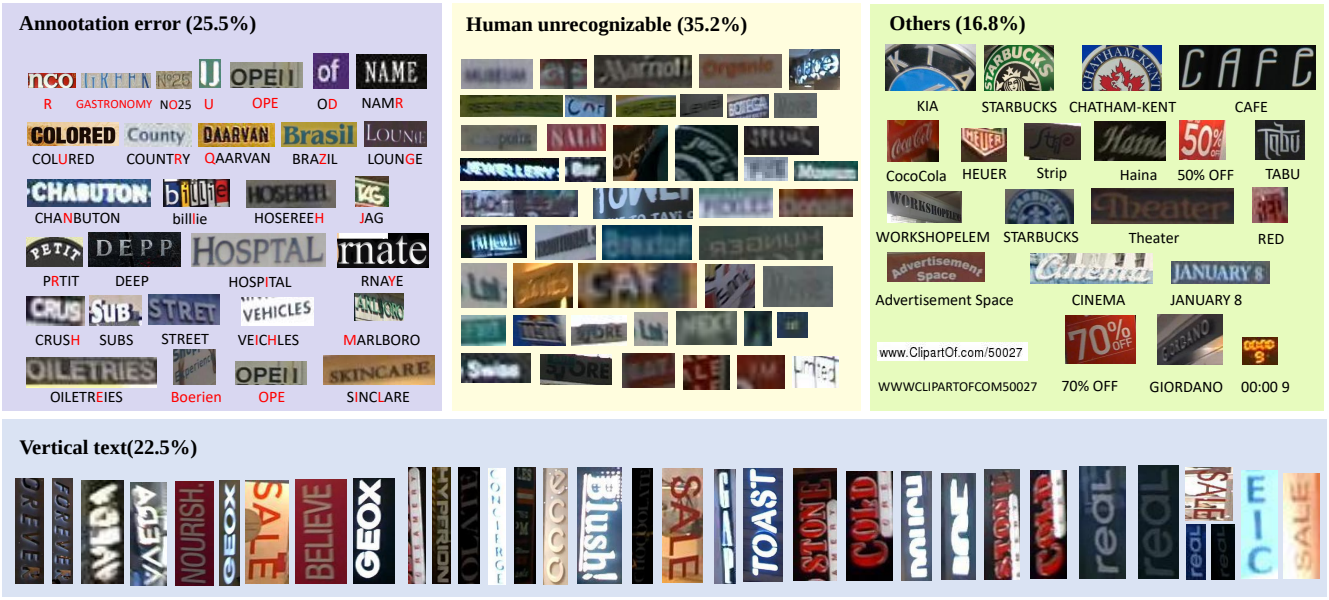

Figure 2. Examples of unrecognized samples in six common benchmarks.

trained on MLT17 [21]), which may contain inherent biases and lead to a lack of diversity in the detected text instances. Therefore, investigating the usage of detectors trained on larger datasets to obtain a larger number of text instances is a potential direction for future research.

Table 1. Comparison of different cropping ways. Settings remain the same as in Tab .3 (paper).

Method	Training Data	Crop method	Acc-UL
SATRN [12]	MJ, ST	axis-aligned	72.09
SATRN [12]	MJ, ST	rotated	73.12
ABINet [3]	MJ, ST	axis-aligned	70.73
ABINet [3]	MJ, ST	rotated	71.19

Table 2. Comparison of different cropping ways. Settings remain the same as in Tab .6 (paper).

Method	Training Data	Crop method	Acc-CB
SATRN [12]	Union14M-L	axis-aligned	91.40
SATRN [12]	Union14M-L	rotated	89.03 (-2.37)
ABINet [3]	Union14M-L	axis-aligned	92.02
ABINet [3]	Union14M-L	rotated	90.13 (-1.89)

2.2. Comparison of Different Cropping Methods

We validate whether the large performance gap in Tab. 3 (paper) is caused by axis-aligned crop. As shown in Tab .1, STR models still perform poorly when using rotated crop, suggesting that the challenges inside Union14M-L are not caused by axis-aligned crops. Moreover, when training with rotated crop images, models exhibit inferior performance as shown in Tab .2, verifying our conjecture in that STR models will gain more robustness when training with a more noised text image. The inconsistency between STR and STD has been a less explored problem (E.g., The STR community used to focus on curve text recognition despite arbitrary shape text detectors being famous).

2.3. Difficulty Assignment in Union14M-L

Our focus is on analyzing the challenges that existing STR models encounter in real-world scenarios. Therefore, we are interested in analyzing the samples that present difficulties. As shown in Fig. 3, we categorize the images in Union14M-L into five difficulty levels using an error voting method. Specifically, given an image I and its corresponding ground truth Y, we conduct forward inference on I using the 13 STR models, and the prediction results are denoted as $[Y_1, Y_2, \cdots, Y_{13}]$. The voting list is defined as $V = [v_1, v_2, \cdots, v_{13}]$, where v_i is defined as:

$$v_i = \begin{cases} 1, & \text{if } Y_i = Y \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

Then each image is empirically assigned to a difficulty level according to the number of correct predictions:

$$\text{level} = \begin{cases} challenging, & \text{if sum(V)} = 0 \\ hard, & \text{if sum(V)} \in [1, 4] \\ medium, & \text{if sum(V)} \in [5, 7] \\ normal, & \text{if sum(V)} \in [8, 10] \\ easy, & \text{if sum(V)} \in [11, 13] \end{cases} \tag{2}$$

The subsets exhibit distinct characteristics based on their respective difficulty levels. For instance, the challenging set comprises a substantial number of images containing curve and vertical text, while the easy set primarily features clear samples and a clear background. The proportion of the images in each difficulty level is illustrated in Fig. 5

Figure 3. Examples of five difficulty levels in Union14M-L.

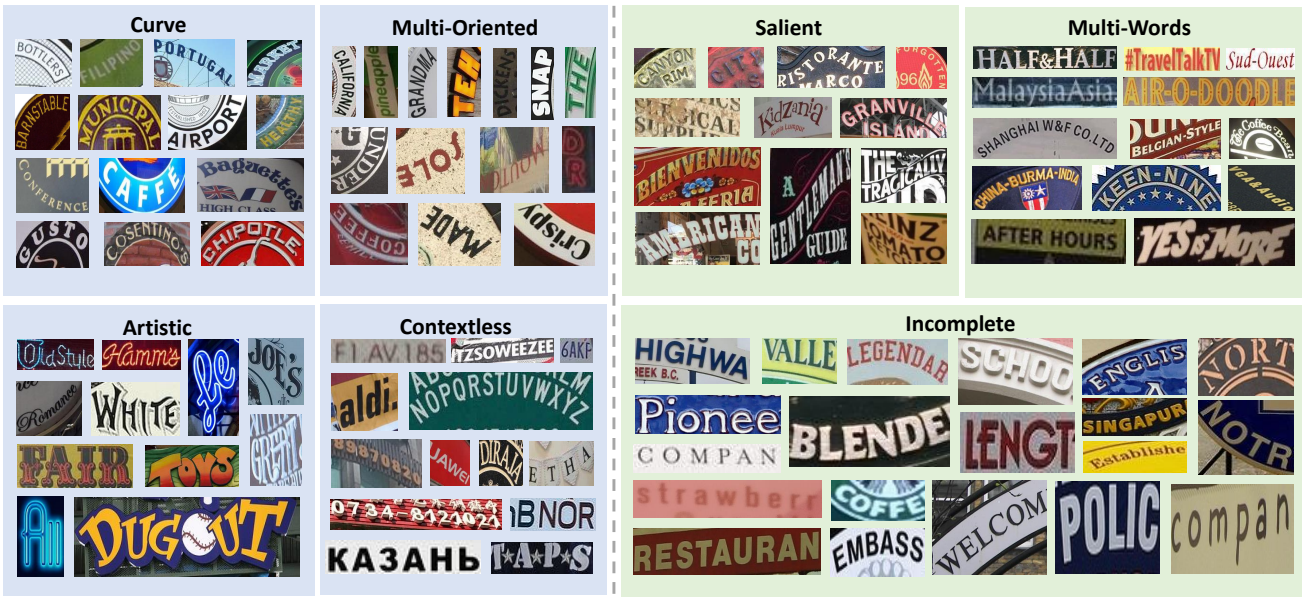

Figure 4. Examples of Union14M-Benchmark.

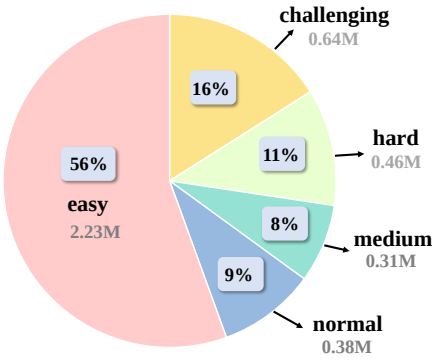

Figure 5. The proportion of samples with different difficulty levels in Union14M-L.

2.4. Consolidation of Union14M-Benchmark

In this section, we provide more information on how we consolidate the Union14M-Benchmark. For each of the seven challenges, excluding incomplete text, we initially collect several reference images from Union14M-L that aligned with the definition of each of the seven challenges. We then recruit five human experts to identify candidate images that shared similarities with the reference images. Subsequently, we manually examined each candidate image and eliminated images that did not meet the specified challenge criteria. Additionally, we also thoroughly recheck the annotations of all images, including digits, cases, and symbols to ensure the quality of the benchmark. For the incomplete text subset, all 1495 images are randomly sampled from the easy set of Union14M-L, and we cropped the first or last letter of each text image.

For the general subset, we sample 20% of the images from each of the five difficulty levels evenly to form the general subset with 400,000 images. With such uniform sampling, the images in the general subset will be more uniformly distributed and more representative. Since the sampling is random, the general subset may have some annotation errors and human unrecognizable samples, as in the six common benchmarks. However, due to a large amount of data, it will take much manual effort to correct these errors, and we also hope that the academic community can work together to correct the errors. In Fig. 4, we show more sam-

Table 3. Vision Transformer variants used in MAERec.

Model	Layers	Hidden size	MLP size	Heads
ViT-Small	12	384	1536	6
ViT-Base	12	768	3072	12

ples of Union14M-Benchmark.

3. Inplementation Details of MAERec

3.1. Vision Transformer

We use vallina Vision Transformer (ViT) [1] as the backbone of MAERec, since it can be easily adapted to masked-image-modeling pre-training. A ViT is composed of a patch embedding layer, position embedding, and a sequence of Transformer blocks.

Patch Embedding: Since a ViT takes a sequence as input, the patch embedding layer is used to convert the input image into a sequence of patches. Specifically, given a text image of size $x \in \mathbb{R}^{H\times W\times C}$, we first resize it to $x_r \in \mathbb{R}^{H_r\times W_r\times C}$, where $H_r = 32$ and $W_r = 128$ following the common practice in STR. We then use a patch embedding layer with a patch size of 4×4 to split the image into non-overlapping patches, in this case, there are 256 patches in total. Each patch is linearly projected to a d-dimensional vector, where d is the embedding dimension of the patch embedding layer.

Position Embedding: To retain positional information in the image, patch embeddings are added with positional embeddings. Specifically, we use sinusoidal positional embeddings in the original ViT [1] as follows:

$$\begin{aligned} \text{PosEnc}(pos, 2i) &= \sin\left(\frac{pos}{10000^{2i/d}}\right) \\ \text{PosEnc}(pos, 2i+1) &= \cos\left(\frac{pos}{10000^{2i/d}}\right) \end{aligned} \tag{3}$$

where $\text{PosEnc}(pos, 2i)$ and $\text{PosEnc}(pos, 2i+1)$ represent the $2i$-th and $(2i+1)$-th dimensions of the positional encoding for a given position pos. d represents the embedding dimension and i ranges from 0 to $\lfloor (d/2) \rfloor - 1$.

Transformer blocks: A Transformer block consists of alternating layers of multi-head self-attention (MHSA) and MLP blocks. Given an input sequence of embeddings $X \in \mathbb{R}^{L\times d}$, where L is the sequence length and d is the embedding dimension, the transformer block can be computed as follows:

$$\text{Block}(X) = \text{LN}(X + \text{LN}(\text{FFN}(\text{LN}(\text{MHSA}(X))))) \tag{4}$$

where LN is the layer normalization layer, FFN is the feed-forward network, and MHSA is the multi-head self-attention layer. We show the configuration of the ViT variants used in MAERec in Tab. 3.

3.2. Masked Image Modeling Pre-training

We adopt MAE [5] framework to pre-train the ViT backbone in MAERec.

Encoder in MAE. We use ViT described in Section 3.1 as the encoder in MAE. Specifically, given patches $x \in \mathbb{R}^{N\times d}$, where N is the number of patches and d is the embedding dimension of the patch embedding layer, we randomly mask 75% of the input patches and only send the remaining 25% visible patches to the ViT encoder. The mask size is set to 4×4 to be consistent with the patch size.

Decoder in MAE. The decoder in MAE is input with the full set of tokens including patch-wise representations from the ViT encoder and learnable mask tokens put in the positions of masked patches. By adding positional embeddings to all the input tokens, the decoder is able to reconstruct the original image from the masked patches. Specifically, we adopt the original decoder used in MAE, which is 8 layers of Transformer blocks and a linear layer to reconstruct the text images from input tokens. The embedding dimension of Transformer blocks is 512 and the number of heads is set to 16. The expanding factor of the MLP layer is set to 4.

Reconstruct target. The decoder in MAE is trained to reconstruct the normalized pixel values of the original image, supervised by MSE loss.

Optimization. We adapt AdamW [17] optimizer to pre-train the model on the 10.6M images of Union14M-U for 20 epochs with an initial learning rate of 1.5e-4. The cosine learning rate scheduler is used with 2 epochs of linear warm-up. The pre-training image size is set to 32×128, and we use no data augmentation. The batch size is set to 256. Pre-training is conducted with 4 NVIDIA A6000 (48GB RAM) GPUs.

3.3. Fine-tuning for Scene Text Recognition

Auto-Regressive Transformer decoder. We use the Transformer decoder in [20] for its superior performance in scene text recognition . Specifically, we use six layers of Transformer decoder to predict text sequence in an auto-regressive manner. The embedding dimension of the Transformer decoder is set to 384 and 768 for the small and base models respectively. The number of heads is set to 8.

Optimization. To be consistent with the pre-training process, we still employ the AdamW optimizer with a weight decay of 0.01, and the cosine learning rate scheduler without warm-up to train the model for 10 epochs. The batch size is set to 64, and the initial learning rate is set to 1e-4. We also adopt the same data augmentation strategy in [3]. Fine-tuning is conducted with 4 NVIDIA 2080Ti (11GB RAM) GPUs.

4. More Experiment Analysis

4.1. Sources of the 13 STR Models

In Tab. 4, we list the sources of the 13 publicly available STR models.

Figure 6. Recognition results on Union14M-Benchmark. GT stands for ground truth. ABINet-S stands for ABINet[3] trained on synthetic datasets (MJ and ST). ABINet-U stands for ABINet trained on Union14M-L. The green text stands for correct recognition and the red text vice versa.

Table 4. The sources of the 13 publicly available STR models.

Method	Link	Official ?
CRNN	https://github.com/Mountchicken/Text-Recognition-on-Cross-Domain-Datasets	No
SVTR	https://github.com/PaddlePaddle/PaddleOCR	Yes
MORAN	https://github.com/Canjie-Luo/MORAN_v2	Yes
ASTER	https://github.com/Mountchicken/Text-Recognition-on-Cross-Domain-Datasets	No
NRTR	https://github.com/open-mmlab/mmocr/tree/main	No
SAR	https://github.com/open-mmlab/mmocr/tree/main	No
DAN	https://github.com/Wang-Tianwei/Decoupled-attention-network	Yes
SATRN	https://github.com/open-mmlab/mmocr/tree/main	No
RobustScanner	https://github.com/open-mmlab/mmocr/tree/main	Yes
SRN	https://github.com/PaddlePaddle/PaddleOCR	No
ABINet	https://github.com/open-mmlab/mmocr/tree/main	No
VisionLAN	https://github.com/wangyuxin87/VisionLAN	Yes
MATRN	https://github.com/byeonghu-na/MATRN	Yes

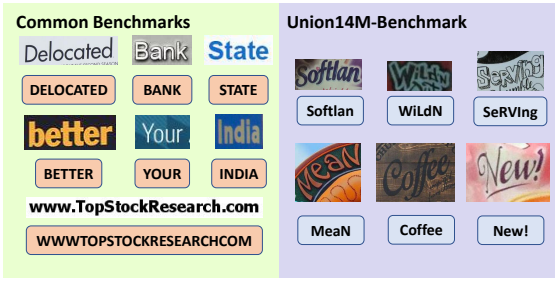

Figure 7. Compare the difference in the annotation of case between common benchmarks and Union14M-Benchmark.

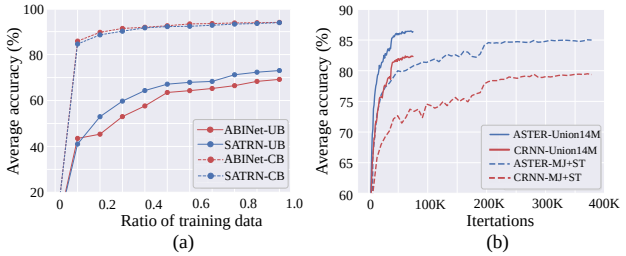

Figure 8. (a) Performance of models trained on increasing fractions of Union14M-L. CB denotes the six common benchmarks; UB denotes Union14M-Benchmark. (b) Performance evolution curves of models trained with Union14M-L or MJ [8, 7] and ST [4] under the same configurations (number of epochs, optimizer, etc.), evaluated on the six common benchmarks.

4.2. WA and WAIC Metrics

In Tab. 5 and Tab. 6, we report the performance of models trained with Union14M-L in terms of WA (word accuracy) and WAIC (word accuracy ignore case) metrics, respectively. While most recent works evaluate STR methods solely on the WAICS (word accuracy ignores case and symbols) metric, which ignores symbols and is case-insensitive, some specific applications require the recognition of symbols and cases, such as captcha recognition and license plate recognition. Compared to models evaluated on the WAICS metric, we can observe a notable decrease in performance when evaluated on both the WA and WAIC metrics. This phenomenon can be attributed to the following reason:

Incorrect case annotation. The performance gap between WA and WAIC is substantial in several common benchmarks, e.g., 50.3% *vs.* 85.89% in IIIT [19] dataset (average accuracy of the 13 STR models). This is primarily due to inconsistent case annotation. As shown in Fig. 7, common benchmarks lack a unified annotation standard for the case. For example, in the IIIT dataset, the letters are all annotated in upper case, whereas in Union14M-Benchmark, we manually check the case annotation of all the 9383 samples in challenge-specific subsets, and correct any case errors. Therefore, the performance gap between WA and WAIC metric in Union14M-Benchmark is much smaller (55.5% *vs.* 57.4%).

Lack of symbols. Additionally, we note that there exists a performance gap between WAIC and WAICS for STR models (88.3% *v.s* 91.2% in common benchmarks; 57.4% *v.s* 62.7% in Union14M-Benchmark). We suggest that this may be due to the infrequent appearance of symbols in the training set in comparison to letters and digits. This can be

Table 5. Performance (WA) of models trained on the training set of **Union14M**. In WA and WAIC metrics, it is impractical to measure the performance of the model on incomplete text set, because the performance is affected by whether the model can correctly predict the case and symbols. For instance, if the model is wrong in case prediction, it will be considered as a false prediction in WA metric, and the error of incomplete text will be ignored.

Type	Method	Common Benchmarks							Union4M Benchmarks							
		IIIT 3000	IC13 1015	SVT 647	IC15 2077	SVTP 645	CUTE 288	Avg	Curve	Multi-Oriented	Artistic	Contextless	Salient	Multi-Words	General	Avg
CTC	CRNN [25]	48.0	44.4	60.9	68.2	70.4	78.5	61.7	18.8	4.2	28.3	37.9	14.4	21.4	56.7	26.0
	SVTR [2]	50.5	46.4	66.3	79.9	61.7	89.9	65.8	69.9	66.2	45.1	61.9	66.4	40.9	73.1	60.5
Attention	MORAN [18]	50.2	45.4	63.5	75.2	59.2	85.8	63.2	41.9	12.0	39.3	49.7	39.4	35.5	41.4	37.0
	ASTER [26]	49.1	45.0	64.8	73.8	58.0	83.7	62.4	36.9	12.1	35.6	46.9	29.0	33.4	62.9	36.7
	NRTR [24]	50.5	47.1	67.7	77.1	60.3	90.3	65.5	47.3	38.6	47.8	64.3	38.7	49.5	71.4	51.1
	SAR [13]	50.5	46.7	67.1	83.5	62.6	90.6	66.8	66.1	53.4	53.3	66.6	55.4	49.8	72.1	59.5
	DAN [12]	49.6	46.3	64.8	74.4	57.7	84.7	62.9	43.9	21.9	43.7	55.1	39.8	38.4	65.1	44.0
	SATRN [28]	50.7	47.3	69.4	83.5	65.0	93.4	68.8	72.0	63.8	58.9	69.5	67.6	45.8	77.2	65.2
	RobustScanner [31]	50.2	46.4	67.4	79.0	61.6	91.0	65.9	63.3	51.0	54.0	72.7	54.7	46.7	71.9	59.2
LM	SRN [30]	50.1	45.5	64.3	74.3	58.8	87.8	63.5	48.0	19.3	43.2	54.9	39.9	27.7	42.9	39.4
	ABINet [3]	50.5	47.0	69.2	83.5	65.6	90.6	67.7	72.2	58.7	57.4	66.0	67.6	41.5	75.6	62.7
	VisionLAN [29]	50.4	45.8	66.0	75.6	60.3	90.6	64.8	68.0	54.7	50.1	58.8	62.5	36.9	70.5	57.4
	MATRN [20]	50.9	47.2	69.6	84.0	65.9	94.1	68.6	78.4	65.0	61.7	69.7	73.0	52.6	76.6	68.1
Ours	MAERec-S w/o PT	51.0	47.7	68.6	82.6	64.7	93.4	68.0	72.7	63.7	57.7	70.4	67.9	48.6	77.1	65.4
	MAERec-S	51.0	47.7	69.4	82.9	66.8	94.1	68.7	78.2	68.8	63.7	76.5	73.2	50.1	78.7	69.9
	MAERec-B w/o PT	50.9	47.6	69.7	83.0	66.1	93.1	68.4	73.7	65.2	57.6	69.7	69.7	48.1	78.1	66.0
	MARec-B	51.3	48.0	70.9	85.2	67.1	95.1	69.6	85.3	81.4	70.9	79.2	80.1	54.6	82.1	76.2

Table 6. Performance (WAIC) of models trained on the training set of **Union14M**.

Type	Method	Common Benchmarks							Union4M Benchmarks							
		IIIT 3000	IC13 1015	SVT 647	IC15 2077	SVTP 645	CUTE 288	Avg	Curve	Multi-Oriented	Artistic	Contextless	Salient	Multi-Words	General	Avg
CTC	CRNN [25]	81.5	91.3	82.4	69.9	69.8	79.2	79.0	18.9	4.3	31.9	39.3	15.1	21.5	58.1	27.0
	SVTR [2]	85.8	94.7	92.4	82.1	85.1	91.0	88.5	70.5	66.6	50.2	63.0	71.4	42.6	74.7	62.7
Attention	MORAN [18]	85.6	93.6	87.3	77.1	82.6	86.1	85.4	42.4	12.4	44.3	51.1	41.0	36.8	42.9	38.7
	ASTER [26]	84.1	92.0	87.6	75.5	79.5	84.0	83.8	37.4	12.5	39.2	47.9	30.2	34.5	64.4	38.0
	NRTR [24]	85.7	96.2	92.3	78.8	83.9	90.3	87.9	47.9	39.1	51.8	65.1	40.1	51.4	72.9	52.6
	SAR [13]	86.5	95.3	90.7	81.6	86.1	91.0	88.5	66.9	54.7	58.0	69.0	57.0	51.2	73.7	61.5
	DAN [12]	84.8	94.6	86.7	76.6	78.5	84.7	84.3	44.6	22.1	47.0	56.6	41.5	39.8	66.7	45.5
	SATRN [28]	86.6	96.2	93.5	85.5	89.9	93.4	90.9	73.0	64.7	64.3	71.1	69.2	47.4	78.8	66.7
	RobustScanner [31]	85.8	95.1	90.4	80.8	85.6	92.0	88.3	64.2	52.8	58.7	72.7	56.9	47.8	73.5	60.9
LM	SRN [30]	85.6	94.2	88.6	76.8	82.9	88.5	86.1	48.7	20.0	47.6	57.9	41.6	27.9	60.7	42.5
	ABINet [3]	86.5	96.8	94.1	85.8	90.9	91.7	91.0	73.0	59.6	62.2	66.3	69.5	43.1	75.6	65.5
	VisionLAN [29]	86.1	94.6	89.3	82.1	84.3	91.3	88.0	68.8	55.2	54.4	60.1	64.7	37.9	72.1	57.4
	MATRN [20]	87.0	97.1	94.4	86.3	92.1	94.4	91.9	79.3	66.0	67.3	71.0	74.9	53.8	78.4	70.0
Ours	MAERec-S w/o PT	86.8	96.9	93.7	84.9	89.6	93.8	91.0	73.7	64.4	62.1	71.5	69.5	49.3	78.7	67.0
	MAERec-S	87.3	97.0	95.1	85.3	92.1	95.1	92.0	79.3	69.5	68.9	77.8	75.1	51.9	80.4	71.8
	MAERec-B w/o PT	86.8	97.2	85.5	95.4	91.6	94.1	91.8	74.8	65.7	62.1	80.0	71.6	50.2	79.7	69.2
	MARec-B	87.9	97.8	96.5	87.7	93.8	95.8	93.2	86.6	82.1	75.9	80.7	82.2	56.2	83.8	78.2

interpreted as a class imbalance issue, which requires further investigation.

4.3. Data Saturation

We conducted a data ablation study to demonstrate the sufficiency of data in Union14M-L. We select ABINet[3] and SATRN[12], and train them on the increasing fractions of the Union14M-L dataset. As depicted in Fig. 8a, the accuracy increases sharply in the beginning and eventually levels out. This indicates that the real data in Union14M-L are sufficient, and adding more real data may not lead to significant performance gain. Moreover, as shown in Fig. 8b, even though the data in Union14M-L are only 1/4 of the synthetic data, training on Union14M-L requires much fewer iterations (four times less) to achieve higher accuracy, which aligns with the Green AI[22] philosophy.

4.4. Data Matters in Self-Supervised Pretraining

In Tab. 7, we compare different dataset combinations used in pre-training and fine-tuning. When pre-training and fine-tuning are both performed on synthetic datasets, MAERec can barely gain a performance boost (89.9 $\rightarrow$ 89.9 for CB, 46.0% $\rightarrow$ 46.1% for UB). However, when fine-tuning is performed on Union14M-L, MAERec can ex-

Table 7. Compare the performance of MAERec-S with different pre-training and fine-tuning datasets. Acc-CB denotes the average accuracy on six common benchmarks. Acc-UB denotes the average accuracy on Union14M-Benchmark (Exclude incomplete text subset).

No.	Pre-train	Fine-tune	Acc-CB	Acc-UB
1	-	MJ, ST	89.9	46.0
2	-	Union14M-L	94.1	73.5
3	MJ, ST	MJ, ST	89.9	46.1
4	MJ, ST	Union14M-L	94.0	75.0
5	Union14M-U	Union14M-L	**95.1**	**78.6**

hibit a performance boost when either pre-trained on synthetic datasets (73.5% $\rightarrow$ 75.0% for UB) or on Union14M-U (73.5% $\rightarrow$ 78.6% for UB). This indicates that fine-tuning on real data is vital for self-supervised learning, and Union14M-U is preferable to synthetic datasets for pre-training (78.6% *vs.* 75.0%).

4.5. Visualize Recognition Results

We show some recognition results on Union14M-Benchmark in Fig. 6. Compared with models trained on synthetic data, training on Union14M can empower STR models to cope with various complex real-world scenarios, thus significantly improving their robustness.

4.6. Why MIM Pre-training Works for STR

When MAERec is pre-trained using MAE on Union14M-U, it shows significant improvement in the STR downstream task. The reason behind this improvement could be attributed to the pre-training process of MIM, where a large portion of the text image (75%) is covered, resulting in only a few patches of each character being visible to the ViT backbone. As a result, if the decoder needs to reconstruct the original image, the ViT backbone must learn to recognize the smallest part of a character to infer the whole character, as shown in Fig. 9. After pre-training, the ViT backbone has learned to differentiate between different characters during pre-training, and the downstream recognition task is essentially a classification task. Hence, the model's performance is significantly enhanced.

References

[1] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An Image is Worth 16x16 Words: Transformers for image recognition at scale. *ICLR*, 2021. 4

[2] Yongkun Du, Zhineng Chen, Caiyan Jia, Xiaoting Yin, Tianlun Zheng, Chenxia Li, Yuning Du, and Yu-Gang Jiang. SVTR: Scene text recognition with a single visual model. *IJCAI*, pages 884–890, 2022. 6

[3] 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, 4, 5, 6

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

[5] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, pages 16000–16009, 2022. 4

[6] Brian Kenji Iwana, Syed Tahseen Raza Rizvi, Sheraz Ahmed, Andreas Dengel, and Seiichi Uchida. Judging a book by its cover. *arXiv preprint arXiv:1610.09204*, 2016. 1

[7] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Synthetic data and artificial neural networks for natural scene text recognition. In *NIPS Deep Learning Workshop*, 2014. 5

[8] 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. 5

[9] 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*, pages 1156–1160. IEEE, 2015. 1

[10] Ilya Krylov, Sergei Nosov, and Vladislav Sovrasov. Open images v5 text annotation and yet another mask text spotter. In *ACML*, pages 379–389. PMLR, 2021. 1

[11] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The open images dataset v4. *Int. J. Comput. Vis.*, 128(7):1956–1981, 2020. 1

[12] Junyeop Lee, Sungrae Park, Jeonghun Baek, Seong Joon Oh, Seonghyeon Kim, and Hwalsuk Lee. On recognizing texts of arbitrary shapes with 2d self-attention. In *CVPR Workshops*, pages 546–547, 2020. 2, 6

[13] Hui Li, Peng Wang, Chunhua Shen, and Guyu Zhang. Show, Attend and Read: A simple and strong baseline for irregular text recognition. In *AAAI*, volume 33, pages 8610–8617, 2019. 6

[14] Minghui Liao, Zhisheng Zou, Zhaoyi Wan, Cong Yao, and Xiang Bai. Real-time scene text detection with differentiable binarization and adaptive scale fusion. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2022. 1

[15] Yuliang Liu, Sheng Zhang, Lianwen Jin, Lele Xie, Yaqiang Wu, and Zhepeng Wang. Omnidirectional scene text detection with sequential-free box discretization. In *IJCAI*, pages 3052–3058, 2019. 1

[16] Shangbang Long, Siyang Qin, Dmitry Panteleev, Alessandro Bissacco, Yasuhisa Fujii, and Michalis Raptis. Towards end-to-end unified scene text detection and layout analysis. In *CVPR*, pages 1049–1059, 2022. 1

[17] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 4

[18] Canjie Luo, Lianwen Jin, and Zenghui Sun. MORAN: A multi-object rectified attention network for scene text recognition. *Pattern Recognition*, 90:109–118, 2019. 6

Figure 9. More reconstruction samples. For each triplet, we show the ground truth (top), the masked image (middle), and the reconstructed image (bottom). Images are from *artistic text*, *multi-words text*, and *contextless text* in Union14M-Benchmark.

[19] Anand Mishra, Karteek Alahari, and CV Jawahar. Scene text recognition using higher order language priors. In *BMVC*. BMVA, 2012. 5

[20] Byeonghu Na, Yoonsik Kim, and Sungrae Park. Multi-modal Text Recognition Networks: Interactive enhancements between visual and semantic features. In *ECCV*, pages 446–463, 2022. 4, 6

[21] 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. ICDAR 2019 robust reading challenge on multi-lingual scene text detection and recognition—rrc-mlt-2019. In *ICDAR*, pages 1582–1587. IEEE, 2019. 2

[22] Roy Schwartz, Jesse Dodge, Noah A Smith, and Oren Etzioni. Green AI. *Communications of the ACM*, 63(12):54–63, 2020. 6

[23] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual Captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *ACL*, pages 2556–2565, 2018. 1

[24] Fenfen Sheng, Zhineng Chen, and Bo Xu. NRTR: A no-recurrence sequence-to-sequence model for scene text recognition. In *ICDAR*, pages 781–786. IEEE, 2019. 6

[25] 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. 6

[26] 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, 2018. 6

[27] Amanpreet Singh, Guan Pang, Mandy Toh, Jing Huang, Wojciech Galuba, and Tal Hassner. TextOCR: Towards large-scale end-to-end reasoning for arbitrary-shaped scene text. In *CVPR*, pages 8802–8812, 2021. 1

[28] 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*, volume 34, pages 12216–12224, 2020. 6

[29] Yuxin Wang, Hongtao Xie, Shancheng Fang, Jing Wang, Shenggao Zhu, and Yongdong Zhang. From Two to One: A new scene text recognizer with visual language modeling network. In *CVPR*, pages 14194–14203, 2021. 6

[30] 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 12113–12122, 2020. 6

[31] Xiaoyu Yue, Zhanghui Kuang, Chenhao Lin, Hongbin Sun, and Wayne Zhang. RobustScanner: Dynamically enhancing positional clues for robust text recognition. In *ECCV*, pages 135–151. Springer, 2020. 6

[32] Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, and Jiajun Liang. EAST: an efficient and accurate scene text detector. In *CVPR*, pages 5551–5560, 2017. 1