Leveraging Text Localization for Scene Text Removal via Text-aware Masked Image Modeling

Zixiao Wang[1], Hongtao Xie[1]*, YuXin Wang[1], Yadong Qu[1], Fengjun Guo[2], and Pengwei Liu[2]

[1] University of Science and Technology of China
[2] IntSig Information Co., Ltd
{wzx99,wangyx58,qqqyd}@mail.ustc.edu.cn, htxie@ustc.edu.cn
{fengjun_guo,pengwei_liu}@intsig.net

Abstract. Existing scene text removal (STR) task suffers from insufficient training data due to the expensive pixel-level labeling. In this paper, we aim to address this issue by introducing a Text-aware Masked Image Modeling algorithm (TMIM), which can pretrain STR models with low-cost text detection labels (e.g., text bounding box). Different from previous pretraining methods that use indirect auxiliary tasks only to enhance the implicit feature extraction ability, our TMIM first enables the STR task to be directly trained in a weakly supervised manner, which explores the STR knowledge explicitly and efficiently. In TMIM, first, a Background Modeling stream is built to learn background generation rules by recovering the masked non-text region. Meanwhile, it provides pseudo STR labels on the masked text region. Second, a Text Erasing stream is proposed to learn from the pseudo labels and equip the model with end-to-end STR ability. Benefiting from the two collaborative streams, our STR model can achieve impressive performance only with the public text detection datasets, which greatly alleviates the limitation of the high-cost STR labels. Experiments demonstrate that our method outperforms other pretrain methods and achieves state-of-the-art performance (37.35 PSNR on SCUT-EnsText). Code will be available at https://github.com/wzx99/TMIM.

Keywords: Scene text removal · Pretraining · Masked image modeling

1 Introduction

Scene text removal (STR) plays an important role in many real-world applications, such as privacy protection and picture editing [21,38,45]. It aims to delete the text regions in the given image and fill them with proper background content. Recently, with the success of the EnsNet [45], deep learning based methods have been widely applied in STR tasks [21,31,34]. These data-driven algorithms demonstrate promising text removal ability under the supervision of pixel-level

*Corresponding Author

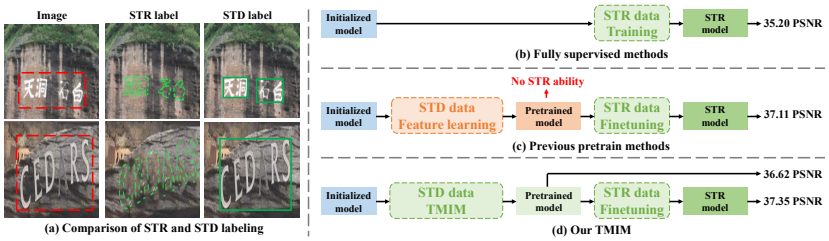


Fig. 1: (a)Comparison of the annotation in scene text removal (STR) task and scene text detection (STD) task. It is clear that STR requires fine-grained manual modification. While STD only requires box-level text detection labeling which can be obtained by advanced OCR systems. (b-c) previous fully supervised [20] and pretraining [27] methods. (d)our TMIM.

annotations. However, compared with labels in other fundamental OCR tasks such as scene text detection (STD) task (Fig. 1(a)), annotating a large number of pixel-level text removal labels is expensive and time-consuming (5 - 10 minutes per image [21]). As a result, the size of annotated STR datasets is relatively small (e.g., 2.7k in [21] vs 111k in existing STD datasets as explained in Sec. 4.1), which is inadequate to fully exploit the potential of advanced STR models.

Aiming at addressing the issue of limited training data, some previous works utilize synthetic datasets for training [10,45]. To enhance the quality of synthetic data, Tang et al. [31] and Jiang et al. [13] develop the Text Synthesis Engine to synthesize more challenging texts. But the domain gap between synthetic and real data can not be perfectly eliminated, resulting in limited performance when applied to real data. Other works propose to pretrain models on additional data. Zdenek et al. [43] follow image inpainting approaches [26] and pretrain the model on general inpainting datasets. To fit the data distribution of scene text images, Peng et al. [27] further introduce STD datasets for feature learning. They combine masked image modeling (MIM) and STD as pretraining tasks to enhance the feature representation. However, when the texts are masked, the MIM pretraining tends to recover the masked text rather than remove it, which is in conflict with the STR target. Thus, although the STD datasets can provide additional scene text images at a low cost, the lack of STR labels forces existing methods to design indirect pretraining tasks for implicit feature learning that lacks specificity to STR (Fig. 1(c)), resulting in redundancy and inefficiency.

To further enhance the STR pretraining efficiency, we focus on addressing the conflict between MIM and STR and propose a Text-aware Masked Image Modeling (TMIM) pretraining framework, which can directly employ STR training on STD datasets through a novel utilization of text detection labels. The TMIM consists of two parallel streams: a Background Modeling (BM) stream for learning non-text background generation rules and a Text Erasing (TE) stream for end-to-end STR learning. Specifically, in the BM stream, we first put masks on all text regions and random non-text regions, and then train the model to recover them. In contrast to general MIM methods that aim to reconstruct all masked

regions, we solely supervise the results on non-text regions. Thus, the model is forced to concentrate on generating non-text content, which greatly eliminates the harmful ambiguity of whether to generate text or background in the pretraining stage. Meanwhile, the prediction results on masked text regions can further serve as pseudo labels for STR learning. Hence, we present the TE stream to utilize these labels for end-to-end text removal training, which explicitly enhances the STR capability. Besides, we propose a Prompt-based Multi-task Learning strategy to enable the two streams to share the same model weights and train the two streams simultaneously, which efficiently transfers background generation knowledge in the BM stream to the TE stream for STR, thereby simplifying the training process. Compared with previous methods in Fig. 1(c-d), our TMIM first enables end-to-end STR training with only text detection labels, thereby alleviating the constraints imposed by the high-cost small-scale STR dataset.

It is worth noting that we focus on a data perspective for low-labeling-cost pretraining rather than designing a specific model architecture. Since there already exists many large-scale labeled STD datasets(111k in STD vs 2.7k in STR as listed in Sec. 4.1), we believe it is valuable for our TMIM to further explore the potential in them to alleviate the data limitation problem. The proposed TMIM offers a straightforward and effective framework that can be simply applied to existing STR model architectures for stability performance improvement. We investigate the results of our method on 5 advanced models, covering both CNN and Transformer architectures, which results in an over 0.34 PSNR improvement across all models. Especially, our TMIM enhances Uformer-B [39] to a new state-of-the-art PSNR (37.35) on the SCUT-EnsText dataset, surpassing other pretraining methods by 0.24. Moreover, our weakly supervised pretrained models with only STD labels also achieve remarkable STR performance with 36.62 PSNR, outperforming all compared fully supervised no-finetuning models with 0.3 improvements. These results validate that our TMIM effectively captures valuable STR knowledge from low-cost STD datasets. Overall, our approach presents a novel solution for label-limited STR training, which can serve as a plug-and-play method to support complex STR models from a new data perspective. The contributions of our method are summarized as follows:

- We demonstrate that the STR ability can be obtained using text detection labels, which significantly alleviates the limitation of insufficient STR annotations and first enables STD datasets to train end-to-end STR models.
- We propose a novel weakly supervised Text-aware Masked Image Modeling framework (TMIM) for STR pretraining. By applying non-text generation rules on text regions, TMIM can transfer background modeling knowledge for text removal efficiently.
- Our method outperforms other pretrain methods and achieves state-of-the-art performance on the SCUT-EnsText dataset (37.35 PSNR). And our pretrained model with only STD dataset also surpasses most existing STR methods (36.62 PSNR).

2 Related Work

Scene text removal. STR is an important technique for many applications and has become a research hotspot. In the early years, traditional text removal methods [17, 36] follow a two-stage method that first detected the text localization and then use the manually designed inpainting algorithm to erase the texts. With the development of deep learning, researchers tend to introduce deep neural networks for end-to-end text removal. EnsNet [45] combines a fully convolutional network and an adversarial network to reduce the artifacts. As text region often takes up a small part of the whole image, some works [5, 18, 20, 21, 33, 38] propose to add a text region mask to help the network focus on text region. For example, MTRNet [34] and MTRNet++ [33] introduce an additional mask to guide the network to erase texts in the masked region. Erasenet [21] adds text detection as an auxiliary task to enhance the perception of text regions. PERT [38] combines text detection and text removal results together and designs a region-based modification strategy to explicitly prevent erasing on non-text regions. However, the data-driven deep learning algorithms heavily rely on training labels [7, 8, 42], which is insufficient in existing small-scale STR datasets.

To address this problem, Jiang et al. [13] propose a self-supervised synthesis algorithm to obtain high-quality synthetic data. Since synthetic data still has a domain gap with real data, ViTEraser [27] introduces extra real text detection datasets and presents indirect self-supervised pretraining tasks for implicit representation learning. Compared with them, this paper aims at using text detection data to directly train the model for explicit STR learning, which significantly enhances the training efficiency.

Masked Image Modeling. MIM aims to predict the masked region of the given image, which is a general pretraining method for various visual tasks. BEiT [2] proposes to randomly replace some visual tokens with the mask token. It trains the model to recover both the masked visual tokens and the original image. MAE [11] discards the masked tokens and trains the model to only focus on recovering the image. SimMIM [41] further uses larger masks and a lighter decoder to enhance the feature learning on the encoder. Besides, SparK [32] entends the MIM framework for convolutional networks. Recently, MIM-based pretraining methods have been widely applied to text analysis and understanding fields, such as MTM [37] for text detection, MARec [15] for text recognition, and SegMIM [27] for text removal. In contrast to them, we further develop MIM with a weakly supervised framework for explicit STR learning rather than implicit feature learning, which greatly enhances the pretraining efficiency.

3 Methodology

3.1 Pipeline

For the pretraining stage, our TMIM employs weakly supervised STR learning on scene text images with text detection labels, which can be easily obtained from existing large-scale STD datasets or using an OCR system for automatic labeling.

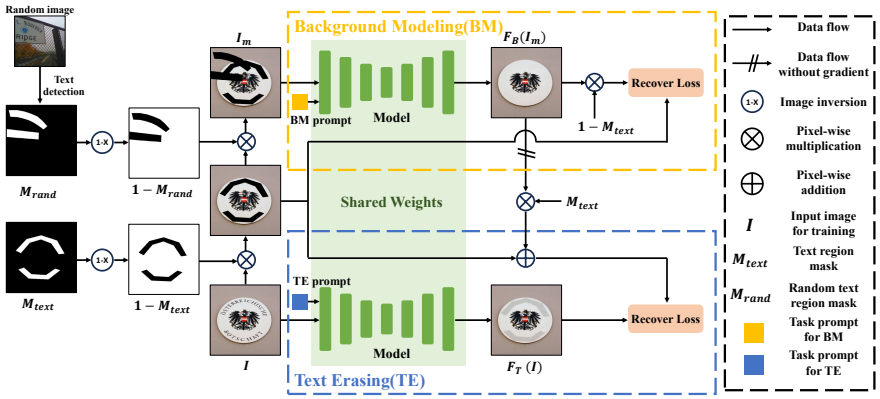


Fig. 2: The overall framework of our TMIM, which consists of two streams: Background Modeling (BM) and Text Erasing (TE)stream. First, the BM stream uses the masked image as the input and trains the model to recover the background regions. Meanwhile, the TE stream adopts the recovery results from the BM stream to build the pseudo labels and train the model for STR.

The framework of TMIM is illustrated in Fig. 2. The TMIM contains two parallel streams: Background Modeling (BM) stream and Text Erasing (TE) stream. In the BM stream, we train the model to recover the masked regions with non-text content, granting the model background generation ability. In the TE stream, we first combine the input image and the background predictions from the BM stream to obtain the pseudo STR label. Then, we send the original input image into the model and utilize the pseudo label for end-to-end STR training. During the optimization, the two streams are trained simultaneously and share the same model weights. To enhance the stability of training two streams in a single model, we further introduce a Prompt-based Multi-task Learning strategy, which injects different task prompts into the model to handle each stream respectively.

For the finetuning stage, as the TE stream follows the general end-to-end STR training framework, we remove the BM stream and solely train the TE stream using the annotated STR dataset. Besides, benefiting from the weakly supervised STR learning in the TE stream, our pretrained model can also achieve satisfactory STR performance without finetuning.

3.2 Background Modeling stream

Existing MIM methods typically apply a random mask to the input image, leading the model to reconstruct the masked regions and thereby learning for content generalization. However, in STR, they tend to predict unfavorable text-like textures on the masked text regions (Fig. 3), which learns the opposite knowledge for text removal. We believe this problem is caused by MIM trying to recover all contents in the masked regions which may also contain texts in it. Thus, such general MIM training blindly encourages the model to rewrite those texts rather

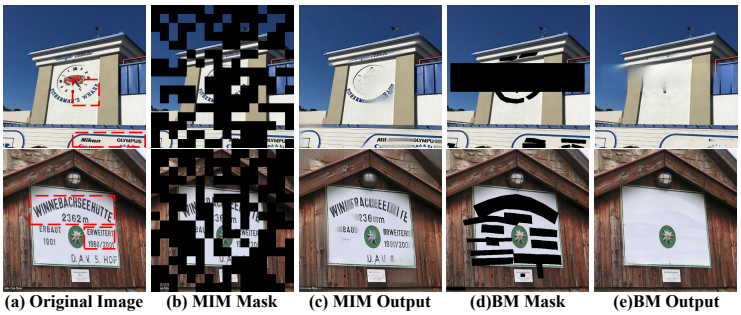


Fig. 3: Comparison with MIM and our BM during the training process. MIM tends to generate text-like content and our BM learns to erase texts.

than remove them, resulting in conflict with the STR target. To address this conflict, we design a Background Modeling (BM) stream to focus on background content generation, which develops MIM in two aspects. Firstly, we introduce text detection labels to prevent optimization from reconstructing text regions. Secondly, we replace the random mask with a text region mask from another image, aligning the shape and size of recovery regions with the text regions.

Specifically, given an input image I and the corresponding binary text region mask M_{text} (1 means the text region). We first use M_{text} to mask all text regions and get the background image I_{bg}. Then we random sample a new text region mask M_{rand} from the dataset to delete the non-text regions in I_{bg} for MIM training. After preprocessing by M_{text} and M_{rand}, the masked image I_m is sent into the BM stream F_B and we force the output $F_B(I_m)$ to recover the non-text regions in I_{bg}. The total optimization process can be formulated as follows,

$$I_{bg} = I * (1 - M_{text}), \tag{1}$$

$$I_m = I * (1 - M_{text}) * (1 - M_{rand}), \tag{2}$$

$$L_{BM} = l(F_B(I_m) * (1 - M_{text}), I_{bg}), \tag{3}$$

where l can be any recovery loss (e.g., L1 loss) and $*$ denotes the Hadamard product. Compared with the general MIM method, the model in the BM stream always learns to predict non-text results during the whole pretraining stage, avoiding the learning of text textures. Thus, the model is biased to generate non-text backgrounds, resulting in three advantages: (1) Benefiting from the model not predicting text-like textures, the conflict between MIM and STR can be addressed, resulting in a consistent optimization target. (2) By utilizing M_{rand} collected from real-world text region masks, our BM stream enables the model to adapt to the shapes of text regions, enhancing its ability to remove texts. (3) Since we also add masks on text regions using M_{text}, the BM predictions on M_{text} can be seen as the coarse text removal labels for the TE stream.

3.3 Text Eraseing stream

Although the BM stream can capture strong background generation knowledge for text removal, it still needs the guidance of text region mask M_{text} to localize the regions for generation. To achieve end-to-end text removal, we further propose a Text Eraseing (TE) stream to simultaneously learn text localization perception and text texture removal.

Concretely, to ensure the flexibility for various STR model structures, the TE stream just adopts the standard end-to-end STR process: the text image I is directly fed into the TE stream F_T to generate final STR predictions. During optimization, we utilize the text region masks (M_{text}) to combine the background regions of the input image I and the text regions of BM results ($F_B(I_m)$) as the pseudo label. The loss is calculated as follows,

$$I_{pseudo} = I * (1 - M_{text}) + F_B(I_m) * M_{text}, \tag{4}$$

$$L_{TE} = l(F_T(I), I_{pseudo}). \tag{5}$$

By replacing all texts in the image with BM predictions as the label, we achieve end-to-end weakly supervised STR learning that leads the model to automatically overwrite non-text backgrounds on text regions. With explicitly optimizing the STR target during pretraining, our TE stream further improves the learning efficiency on STD datasets, resulting in high-quality pretrained models.

3.4 Prompt-based Multi-task Learning

Since both streams utilize the same model but take different inputs and labels, we design a Prompt-based Multi-task Learning (PML) strategy to lead the model performing the required stream. Concretely, we define two 1D vectors as prompts for each stream. In the forward process, we first use the encoder to extract the image feature. Then, the vector for the corresponding task will be duplicated and added to the feature map element-wise, providing a bias to distinguish different tasks. Finally, the biased feature is sent to the decoder for prediction. Thus, the PML does not need to modify the model structure and can be used in most of the end-to-end STR models seamlessly. With PML, the two streams can be trained in parallel without introducing many additional parameters or computations. As a result, the background generation knowledge learned from the BM stream can be exhaustively utilized for text removal. This enhances the efficiency of our framework as well streamlines the pretraining process into one model and one stage. Results in Sec. 4 confirm the effectiveness of this strategy in training models with multiple tasks in TMIM.

3.5 Optimization

For the pretraining stage, we combine $F_B(I_m)$ and $F_T(I)$ for joint optimization, both BM and TE streams use the same recovery loss function and the total loss

is the sum of them. Inspired by [38], the loss can be divided into the following parts: content loss, SSIM loss, and feature loss. To enhance clarity, we denote both $F_B(I_m)$ and $F_T(I)$ as O. And their label I_{bg} and I_{pseudo} are denoted as Y.
Content loss. This loss assumes the prediction result to be the same as the label in both RGB (L1 loss) and frequency (FFL loss [14]) domains:

$$L_C = \lambda_1 L1(O, Y) + \lambda_2 FFL(O, Y), \tag{6}$$

where both λ_1 and λ_2 are set as 15.
SSIM loss. Based on [40], We further introduce SSIM as a training target to optimize the structure similarity between O and its label.

$$L_{SSIM} = 1 - SSIM(O, Y). \tag{7}$$

Feature loss. This loss is commonly used in image-to-image tasks to ensure the content of prediction and label have the same image style. It uses a pretrained VGG net to extract feature maps and calculate their similarity as the optimization target. We sum the GS loss and VGG loss defined in [38] as our feature loss L_F to optimize the style of O.

Overall, the combined pretraining loss for the BM and TE streams is defined as:

$$L_{BM/TE} = L_C + \alpha L_{SSIM} + \beta L_F, \tag{8}$$

where both α and β are set to 1. And the total loss is calculated by:

$$L = L_{BM} + L_{TE}. \tag{9}$$

Finally, as the finetuning stage also follows the TE stream, we simply use L_{TE} as the finetuning loss.

4 Experiments

4.1 Datasets and Implementation Details

Scene text detection datasets. We combine 8 public STD datasets for pretraining, including Total Text [3], ICDAR2015 [16], MLT19 [25], ReCTS [44], ArT [4], LSVT [29], COCO Text [35], and Text OCR [28]. We remove the overlapped test images in SCUT-EnsText [21] and finally get 111066 images.
Scene text removal datasets. We utilize two commonly used STR datasets for finetuing and testing, including SCUT-EnsText [21]and SCUT-Syn [45]. Concretely, SCUT-EnsText is a real-world dataset. It contains 2749 images for training and 813 images for testing. SCUT-Syn is a synthetic dataset that contains 8000 images for training and 800 images for testing.
Model architectures. We use Uformer [39] as our baseline STR model. Besides, to evaluate the generalization of our TMIM, we further introduce other advanced open-source STR architectures for evaluation, including Erasenet [21], PERT

Table 1: The influence of different modules in TMIM. Bold values denote the best results in each column. The baseline model is Uformer-T.

Methods	PSNR↑	MSSIM↑	MSE↓	AGE↓	pEPs↓	pCEPs↓
TE(baseline)	34.07	97.10	0.12	2.04	0.0130	0.0085
+MIM	34.53	97.10	0.11	1.98	0.0121	0.0076
+BM	36.00	**97.54**	**0.08**	1.76	0.0095	0.0060
+BM/PML(TMIM)	**36.18**	97.48	**0.08**	**1.75**	**0.0092**	**0.0056**

Table 2: The STR ability of pretrained models. "FS" denotes fully supervised training on the STR dataset. "WS" denotes weakly supervised pretraining on the STD dataset.

Methods	Strategy	PSNR↑	MSSIM↑	MSE↓	AGE↓
Uformer-T	FS	34.07	97.10	0.12	2.04
Uformer-T+TMIM	WS	**35.59(1.52↑)**	**97.22(0.12↑)**	**0.08(0.04↓)**	**1.91(0.13↓)**
Uformer-S	FS	34.81	97.36	0.11	1.90
Uformer-S+TMIM	WS	**36.05(1.24↑)**	**97.26(0.10↑)**	**0.08(0.03↓)**	**1.85(0.05↓)**
Uformer-B	FS	35.17	97.35	0.11	1.90
Uformer-B+TMIM	WS	**36.62(1.45↑)**	**97.36(0.01↑)**	**0.06(0.05↓)**	**1.71(0.19↓)**

[38], CTRNet [20], and FETNet [23]. Especially, for CTRNet, we remove the text perception module to avoid needing the additional STD model.

Optimization details. In the training stage, we use the AdamW optimizer [22] with the momentum of (0.9, 0.999) for optimization. Following [39], the learning rate and weight decay are set to 2e-4 and 0.02. For augmentation, we employ random flip and random jitter on color, brightness. We pretrain the model on the combined STD dataset for 10 epochs with the batch size of 8. The pretrained models are finetuned on the target STR dataset for 200 epochs with the batch size of 16. All images are resized to 512×512. The experiments are employed with the PyTorch framework on 8 NVIDIA RTX 3090 GPUs.

Evaluation Metrics. Following the previous STR method [21], we evaluate our framework with both Image-Eval and Detection-Eval metrics. For Image-Eval metrics, we utilize PSNR, MSSIM, MSE, AGE, pEPs and pCEPs. We follow [27] to multiply MSE with 100%. For Detection-Eval metrics, we use the pretrained CRAFT model [1] as the text detector to test the precision (P), recall (R), and f-measure (F) on the erased images. If not specified, we use SCUT-EnsText as the test dataset.

4.2 Ablation Study

The effectiveness of each module in TMIM. As shown in Tab. 1, the Uformer-T without pretraining can be seen as the TMIM with only TE stream, resulting in 34.07 PSNR. And with general MIM pretraining, it obtains 34.53

Table 3: The generalization with different STR architectures.

Methods	PSNR↑	MSSIM↑	MSE↓	AGE↓
Erasenet	32.30	95.42	0.15	3.02
+TMIM	**33.63(1.33↑)**	**97.20(1.78↑)**	**0.11(0.04↓)**	**2.45(0.57↓)**
PERT	33.62	97.00	0.13	2.19
+TMIM	**35.21(1.59↑)**	**97.15(0.15↑)**	**0.09(0.04↓)**	**1.88(0.31↓)**
CTRNet	35.20	97.36	0.09	2.20
+TMIM	**35.54(0.34↑)**	**97.48(0.12↑)**	**0.07(0.02↓)**	**2.15(0.05↓)**
FETNet	34.53	97.01	0.13	**1.75**
+TMIM	**35.16(0.63↑)**	**97.22(0.21↑)**	**0.11(0.02↓)**	1.96(-0.21↓)
Uformer-B	35.17	97.35	0.11	1.90
+TMIM	**37.35(2.18↑)**	**97.78(0.43↑)**	**0.05(0.06↓)**	**1.61(0.29↓)**

PSNR. After introducing our BM, we get an obvious improvement of 1.47 PSNR. This benefits from the explicit STR learning and the pseudo labels guiding. With the PML, the model finally achieves 36.18 PSNR which surpasses the baseline with and without pretraining by 2.11 and 1.65. Overall, the improvements demonstrate that: (1) Our method can explore more valuable knowledge from existing STD data to better alleviate the limitation of the small-scale STR dataset. (2) Introducing text localization information to learn non-text content generation and generating pseudo labels for explicit STR learning is more suitable than implicit feature learning in MIM. (3) The task prompt can well balance the two streams in one model for further improvement.

The STR ability of pretrained model. As the pretrained models in our TMIM also have end-to-end STR ability, we list their performance in Tab. 2. We compare our weakly supervised pretrained models trained on STD datasets with the fully supervised models trained on STR datasets. Tab. 2 shows that our pretrained Uformer surpasses the fully supervised baseline on all 3 sizes. Especially, our Uformer-B+TMIM obtains 36.62 PSNR, which also outperforms all non-finetuning STR models in Tab. 5 (36.62 vs 36.32). This proves that our TMIM can explicitly learn valuable STR knowledge from STD datasets without finetuning, providing new insight for label-constrained STR learning.

The generalization of TMIM. As a model-agnostic framework, to evaluate the generalization of our TMIM, we test our TMIM on 5 advanced STR models including both CNN and Transformer architectures: Erasenet [21], PERT [38], CTRNet [20], FETNet [23], and Uformer [39]. As illustrated in Tab. 3, our TMIM can enhance the overall performance on multiple STR architectures. All results on PSNR, MSSIM, and MSE metrics are improved. This demonstrates that our pretraining method is a general framework for further improving the performance of different STR model architectures.

Comparison with other pretraining methods. In Tab. 4, we reimplement other pretraining methods for comparison, including MIM and SegMIM. We

Table 4: Comparison with other pretraining methods.

Method	Pretrain	PSNR↑	MSSIM↑	MSE↓	AGE ↓	pEPs ↓	pCEPs ↓
PERT	×	33.62	97.00	0.13	2.19	0.0135	0.0088
	MIM	33.81	97.10	0.1203	2.07	0.0146	0.0094
	SegMIM	34.92	97.13	0.0963	1.89	0.0121	0.0076
	TMIM(ours)	**35.21**	**97.15**	**0.0942**	**1.88**	**0.0119**	**0.0074**
Uformer-S	×	34.81	97.36	0.1128	1.90	0.0119	0.0077
	MIM	36.03	97.45	0.0832	1.74	0.0093	0.0057
	SegMIM	36.68	97.56	0.0684	1.69	0.0086	0.0053
	TMIM(ours)	**36.87**	**97.74**	**0.0614**	**1.62**	**0.0078**	**0.0048**

Table 5: Comparison with SOTA STR models on SCUT-EnsText. Bold and underlined values denote the 1st and 2nd results in each column. † denotes our reimplement. ‡ denotes pretrained on STD datasets.

Methods	Image-Eval						Detection-Eval			Param
	PSNR↑	MSSIM↑	MSE↓	AGE↓	pEPs↓	pCEPs↓	R↓	P↓	F↓	
Original	-	-	-	-	-	-	69.5	79.4	74.1	-
EraseNet [21]	32.30	95.42	0.15	3.02	0.0160	0.0090	4.6	53.2	8.5	17.8
PERT [38]	33.62	97.00	0.13	2.19	0.0135	0.0088	4.1	50.5	7.6	14
Tang et al. [31]	35.54	96.24	0.09	-	-	-	3.6	-	-	30.7
PSSTRNet [24]	34.65	96.75	0.14	1.72	0.0135	0.0074	5.1	47.7	9.3	4.9
CTRNet [20]	35.20	97.36	0.09	2.20	0.0106	0.0068	1.4	38.4	2.7	159.8
MBE [12]	35.03	97.31	-	2.06	0.0128	0.0088	-	-	-	-
SAEN [5]	34.75	96.53	0.07	1.98	0.0125	0.0073	-	-	-	19.8
FETNet [23]	34.53	97.01	0.13	1.75	0.0137	0.0080	5.8	51.3	10.5	8.5
Uformer-B† [39]	35.17	97.35	0.11	1.90	0.0107	0.0069	2.9	51.6	5.5	50.88
ViTEraser [27]	36.32	97.51	0.0565	1.86	0.0074	<u>0.0041</u>	0.635	37.8	1.248	191.9
ViTEraser‡ [27]	<u>37.11</u>	97.61	<u>0.0474</u>	1.70	<u>0.0066</u>	**0.0035**	**0.389**	<u>29.7</u>	**0.768**	191.9
Uformer-T+TMIM‡	36.18	97.48	0.0813	1.75	0.0092	0.0056	1.457	45.2	2.824	5.29
Uformer-S+TMIM‡	36.87	<u>97.74</u>	0.0614	<u>1.62</u>	0.0078	0.0048	0.595	**12.4**	1.113	20.75
Uformer-B+TMIM‡	**37.35**	**97.78**	**0.0470**	**1.61**	**0.0065**	<u>0.0041</u>	<u>0.533</u>	34.7	<u>1.051</u>	50.88

choose a CNN-based (PERT [38]) and a Transofmer-based (Uformer-S [39]) architecture as the baseline model. For a fair comparison, we also pretrain the baseline model with the compared methods on our 111k STD images and finetune them on the SCUT-EnsText dataset. It is shown that our TMIM surpasses other pretrain methods on all metrics. These results demonstrate that our method is a more suitable framework for the STR task, which can exhaustively explore the upper bound of multiple models.

4.3 Comparison with State-of-the-Art Methods

In Tab. 5, we first apply our TMIM on Uformer and compare it with other advanced STR methods on the real SCUT-EnsText dataset. It can be seen that our Uformer-B+TMIM outperforms other methods, achieving state-of-the-art

Table 6: Comparison with SOTA STR models on SCUT-Syn. Bold values denote the best results in each column. † denotes our reimplement. ‡ denotes pretrained on STD datasets.

Methods	PSNR↑	MSSIM↑	MSE↓	AGE↓	pEPs↓	pCEPs↓
EraseNet [21]	38.32	97.67	0.02	1.60	0.0048	0.0004
PERT [38]	39.40	97.87	0.02	1.41	0.0046	0.0007
Tang et al. [31]	38.60	97.55	0.02	-	-	-
MBE [12]	43.85	98.64	-	**0.94**	0.0013	0.00004
SAEN [5]	38.63	98.27	0.03	1.39	0.0043	0.0004
FETNet [23]	39.14	97.97	0.02	1.26	0.0046	0.0008
Uformer-B† [39]	40.60	98.37	0.0156	1.29	0.0032	0.0004
ViTEraser [27]	42.53	98.45	0.0102	1.19	0.0018	0.000016
ViTEraser‡ [27]	42.97	98.55	0.0092	1.11	0.0015	0.000011
Uformer-T+TMIM‡	41.81	98.62	0.0125	1.16	0.0023	0.000149
Uformer-S+TMIM‡	42.59	98.73	0.0111	1.11	0.0019	0.000087
Uformer-B+TMIM‡	**44.24**	**98.86**	**0.0084**	0.95	**0.0013**	**0.000011**

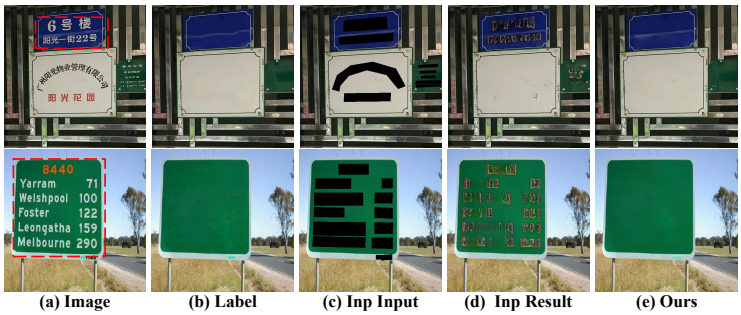


Fig. 4: The qualitative results on SCUT-EnsText compared with Erasenet [21], Tang et al. [31], and CTRNet [20].

PSNR↑ (37.35 vs 37.11), SSIM↑ (97.78 vs 97.61), MSE↓ (0.0470 vs 0.0474), and pEPS↓ (0.0065 vs 0.0066). Especially, compared with the recently proposed ViTEraser [27] which also uses large-scale STD data for pretraining, our Uformer-B+TMIM gets better results on most image-eval metrics with a lighter pretraining process (10 epochs with 111k images vs 100 epochs with 98K images) and model parameters (50.88M vs 191.9M). For detection-eval metrics, [27] performs better while it is sensitive to the detection model, which may be jitter under other detectors. This verifies that our weakly supervised STR learning can cap-

Table 7: Comparison with inpainting methods. † denotes using the ground truth of text region masks for testing.

Methods	PSNR↑	MSSIM↑	MSE↓	AGE↓
CTSDG† [9]	33.10	95.55	0.14	-
SPL† [46]	35.41	97.39	0.07	-
LaMa† [30]	35.25	97.36	0.07	1.97
MAT† [19]	34.22	97.13	0.09	2.19
Uformer-B+TMIM(ours)	**37.35**	**97.78**	**0.05**	**1.61**



(a) Image (b) Label (c) Inp Input (d) Inp Result (e) Ours

Fig. 5: The qualitative results compared with the inpainting method Lama [30]. (c) is the input for inpainting method masked by the text detection ground truth. (d) is the result of LaMa [30].

ture more STR knowledge efficiently. Besides, we illustrated some representative results in Fig. 4.

In Tab. 6, we also test our methods on the synthetic SCUT-Syn dataset. Results show that our method reaches state-of-the-art performance except for MBE [12] in the AGE metric, which combines multiple networks for prediction. And the overall results again verify the superiority of our method.

4.4 Discussion

Comparison with inpainting methods. Existing inpainting methods also can be used for text removal when providing additional text detection results on the test set. However, similar to MIM, the training target of the inpainting task is also to reconstruct the original contents, which results in unexpected text-like textures rather than removing the texts on scenes where text frequently appears (as illustrated in Fig. 5). [21] also shows that inpainting models perform poorly on recovering text regions as there is a domain gap between STR and inpainting tasks. Moreover, we introduce the ground truth of the text region for inpainting models to test the STR ability in Tab. 7. The results of CTSDG [9] and SPL [46] are collected from [20]. And we further reimplement LaMa [30] and MAT [19] using the STD and STR dataset in Sec. 4.1. It is obvious that our TMIM obtains

Table 8: Comparison with detection-based STR methods. † denotes using the text region masks from [1]. ‡ denotes only concern the results on text region follow [18].

Methods	PSNR↑	MSSIM↑	MSE↓	AGE↓
Tang et al.† [31]	35.54	97.00	0.09	-
CTRNet† [20]	35.20	97.36	0.09	2.20
DeepEraser† [6]	35.84	97.48	0.08	1.71
Uformer-B+TMIM(ours)	**37.35**	**97.78**	**0.05**	**1.61**
CTRNet‡ [20]	37.20	97.66	0.07	-
GaRNet‡ [18]	41.37	98.46	-	0.59
DeepEraser‡ [6]	42.47	98.65	-	0.59
Uformer-B+TMIM‡(ours)	**43.88**	**98.88**	**0.04**	**0.54**

better performance without the need for text detection ground truth (37.35 vs 35.41 PSNR), verifying our method is more suitable for the STR task.

Comparison with Detection-based STR methods. In contrast to using the STD data for pretraining, there are also many two stage STR methods [6, 18, 20, 31] that introduce additional pretrained STD models to predict text region masks as the auxiliary information. The main difference between our TMIM and them lies in: (1) We only need text region masks during the training stage. In the inference stage, both the pretrained and finetuned models in our TMIM can directly remove the texts in a simple one-stage process without using text detection results. (2) As listed in Tab. 8, our method without text detection results also outperforms these two stage STR methods (37.35 vs 35.84 PSNR). After following [18] to only focus on the results in text regions, our model finally achieves state-of-the-art 43.88 PSNR, demonstrating our method is more efficient than existing detection-based STR methods.

5 Conclusions

In this paper, we propose a weakly supervised STR pretraining framework to address the limitation of high-cost small-scale STR datasets, named Text-aware Masked Image Modeling (TMIM). The TMIM introduces low-cost text localization for pretraining, which enables end-to-end STR training under weak supervision. We build a BM stream to learn background generation rules from non-text regions and predict pseudo STR labels on text regions. Meanwhile, we design a TE stream to learn text removal using pseudo labels. Experiments show that our pretrained model already obtains impressive performance and finally achieves state-of-the-art performance after finetuning. In the future, we will explore a more efficient STR learning method under low-cost text detection labeling, further enhancing the final STR performance.

Acknowledgements. This work is supported by the National Key Research and Development Program of China (2022YFB3104700), the National Nature

Science Foundation of China (62121002, U23B2028, 62102384). We acknowledge the support of GPU cluster built by MCC Lab of Information Science and Technology Institution, USTC. We also thank the USTC supercomputing center for providing computational resources for this project.

References

1. Baek, Y., Lee, B., Han, D., Yun, S., Lee, H.: Character region awareness for text detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9365–9374 (2019)
2. Bao, H., Dong, L., Piao, S., Wei, F.: Beit: Bert pre-training of image transformers. arXiv preprint arXiv:2106.08254 (2021)
3. Ch'ng, C.K., Chan, C.S.: Total-text: A comprehensive dataset for scene text detection and recognition. In: 2017 14th IAPR international conference on document analysis and recognition (ICDAR). vol. 1, pp. 935–942. IEEE (2017)
4. Chng, C.K., Liu, Y., Sun, Y., Ng, C.C., Luo, C., Ni, Z., Fang, C., Zhang, S., Han, J., Ding, E., et al.: Icdar2019 robust reading challenge on arbitrary-shaped text-rrc-art. In: 2019 International Conference on Document Analysis and Recognition (ICDAR). pp. 1571–1576. IEEE (2019)
5. Du, X., Zhou, Z., Zheng, Y., Ma, T., Wu, X., Jin, C.: Modeling stroke mask for end-to-end text erasing. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 6151–6159 (2023)
6. Feng, H., Wang, W., Liu, S., Deng, J., Zhou, W., Li, H.: Deeperaser: Deep iterative context mining for generic text eraser. arXiv preprint arXiv:2402.19108 (2024)
7. Ge, J., Xie, H., Min, S., Li, P., Zhang, Y.: Dual part discovery network for zero-shot learning. In: Proceedings of the 30th ACM International Conference on Multimedia. pp. 3244–3252 (2022)
8. Ge, J., Xie, H., Min, S., Zhang, Y.: Semantic-guided reinforced region embedding for generalized zero-shot learning. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35, pp. 1406–1414 (2021)
9. Guo, X., Yang, H., Huang, D.: Image inpainting via conditional texture and structure dual generation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 14134–14143 (2021)
10. Gupta, A., Vedaldi, A., Zisserman, A.: Synthetic data for text localisation in natural images. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2315–2324 (2016)
11. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 16000–16009 (2022)
12. Hou, Y., Chen, J.J., Wang, Z.: Multi-branch network with ensemble learning for text removal in the wild. In: Proceedings of the Asian Conference on Computer Vision. pp. 1333–1349 (2022)
13. Jiang, G., Wang, S., Ge, T., Jiang, Y., Wei, Y., Lian, D.: Self-supervised text erasing with controllable image synthesis. In: Proceedings of the 30th ACM International Conference on Multimedia. pp. 1973–1983 (2022)
14. Jiang, L., Dai, B., Wu, W., Loy, C.C.: Focal frequency loss for image reconstruction and synthesis. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 13919–13929 (2021)

15. Jiang, Q., Wang, J., Peng, D., Liu, C., Jin, L.: Revisiting scene text recognition: A data perspective. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 20543–20554 (2023)
16. Karatzas, D., Gomez-Bigorda, L., Nicolaou, A., Ghosh, S., Bagdanov, A., Iwamura, M., Matas, J., Neumann, L., Chandrasekhar, V.R., Lu, S., et al.: Icdar 2015 competition on robust reading. In: 2015 13th international conference on document analysis and recognition (ICDAR). pp. 1156–1160. IEEE (2015)
17. Khodadadi, M., Behrad, A.: Text localization, extraction and inpainting in color images. In: 20th Iranian Conference on Electrical Engineering (ICEE2012). pp. 1035–1040. IEEE (2012)
18. Lee, H., Choi, C.: The surprisingly straightforward scene text removal method with gated attention and region of interest generation: A comprehensive prominent model analysis. In: European Conference on Computer Vision. pp. 457–472. Springer (2022)
19. Li, W., Lin, Z., Zhou, K., Qi, L., Wang, Y., Jia, J.: Mat: Mask-aware transformer for large hole image inpainting. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10758–10768 (2022)
20. Liu, C., Jin, L., Liu, Y., Luo, C., Chen, B., Guo, F., Ding, K.: Don't forget me: accurate background recovery for text removal via modeling local-global context. In: European Conference on Computer Vision. pp. 409–426. Springer (2022)
21. Liu, C., Liu, Y., Jin, L., Zhang, S., Luo, C., Wang, Y.: Erasenet: End-to-end text removal in the wild. IEEE Transactions on Image Processing **29**, 8760–8775 (2020)
22. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)
23. Lyu, G., Liu, K., Zhu, A., Uchida, S., Iwana, B.K.: Fetnet: Feature erasing and transferring network for scene text removal. Pattern Recognition **140**, 109531 (2023)
24. Lyu, G., Zhu, A.: Psstrnet: progressive segmentation-guided scene text removal network. In: 2022 IEEE International Conference on Multimedia and Expo (ICME). pp. 1–6. IEEE (2022)
25. Nayef, N., Patel, Y., Busta, M., Chowdhury, P.N., Karatzas, D., Khlif, W., Matas, J., Pal, U., Burie, J.C., Liu, C.l., et al.: Icdar2019 robust reading challenge on multilingual scene text detection and recognition—rrc-mlt-2019. In: 2019 International conference on document analysis and recognition (ICDAR). pp. 1582–1587. IEEE (2019)
26. Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A.A.: Context encoders: Feature learning by inpainting. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2536–2544 (2016)
27. Peng, D., Liu, C., Liu, Y., Jin, L.: Viteraser: Harnessing the power of vision transformers for scene text removal with segmim pretraining. arXiv preprint arXiv:2306.12106 (2023)
28. Singh, A., Pang, G., Toh, M., Huang, J., Galuba, W., Hassner, T.: Textocr: Towards large-scale end-to-end reasoning for arbitrary-shaped scene text. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8802–8812 (2021)
29. Sun, Y., Ni, Z., Chng, C.K., Liu, Y., Luo, C., Ng, C.C., Han, J., Ding, E., Liu, J., Karatzas, D., et al.: Icdar 2019 competition on large-scale street view text with partial labeling-rrc-lsvt. In: 2019 International Conference on Document Analysis and Recognition (ICDAR). pp. 1557–1562. IEEE (2019)

30. Suvorov, R., Logacheva, E., Mashikhin, A., Remizova, A., Ashukha, A., Silvestrov, A., Kong, N., Goka, H., Park, K., Lempitsky, V.: Resolution-robust large mask inpainting with fourier convolutions. In: Proceedings of the IEEE/CVF winter conference on applications of computer vision. pp. 2149–2159 (2022)
31. Tang, Z., Miyazaki, T., Sugaya, Y., Omachi, S.: Stroke-based scene text erasing using synthetic data for training. IEEE Transactions on Image Processing **30**, 9306–9320 (2021)
32. Tian, K., Jiang, Y., Diao, Q., Lin, C., Wang, L., Yuan, Z.: Designing bert for convolutional networks: Sparse and hierarchical masked modeling. arXiv preprint arXiv:2301.03580 (2023)
33. Tursun, O., Denman, S., Zeng, R., Sivapalan, S., Sridharan, S., Fookes, C.: Mtrnet++: One-stage mask-based scene text eraser. Computer Vision and Image Understanding **201**, 103066 (2020)
34. Tursun, O., Zeng, R., Denman, S., Sivapalan, S., Sridharan, S., Fookes, C.: Mtrnet: A generic scene text eraser. In: 2019 International Conference on Document Analysis and Recognition (ICDAR). pp. 39–44. IEEE (2019)
35. Veit, A., Matera, T., Neumann, L., Matas, J., Belongie, S.: Coco-text: Dataset and benchmark for text detection and recognition in natural images. arXiv preprint arXiv:1601.07140 (2016)
36. Wagh, P.D., Patil, D.: Text detection and removal from image using inpainting with smoothing. In: 2015 International Conference on Pervasive Computing (ICPC). pp. 1–4. IEEE (2015)
37. Wang, K., Xie, H., Wang, Y., Zhang, D., Qu, Y., Gao, Z., Zhang, Y.: Masked text modeling: A self-supervised pre-training method for scene text detection. In: Proceedings of the 31st ACM International Conference on Multimedia. pp. 2006–2015 (2023)
38. Wang, Y., Xie, H., Wang, Z., Qu, Y., Zhang, Y.: What is the real need for scene text removal? exploring the background integrity and erasure exhaustivity properties. IEEE Transactions on Image Processing (2023)
39. Wang, Z., Cun, X., Bao, J., Zhou, W., Liu, J., Li, H.: Uformer: A general u-shaped transformer for image restoration. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 17683–17693 (2022)
40. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing **13**(4), 600–612 (2004)
41. Xie, Z., Zhang, Z., Cao, Y., Lin, Y., Bao, J., Yao, Z., Dai, Q., Hu, H.: Simmim: A simple framework for masked image modeling. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9653–9663 (2022)
42. Yang, X., Song, Z., King, I., Xu, Z.: A survey on deep semi-supervised learning. IEEE Transactions on Knowledge and Data Engineering **35**(9), 8934–8954 (2022)
43. Zdenek, J., Nakayama, H.: Erasing scene text with weak supervision. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 2238–2246 (2020)
44. Zhang, R., Zhou, Y., Jiang, Q., Song, Q., Li, N., Zhou, K., Wang, L., Wang, D., Liao, M., Yang, M., et al.: Icdar 2019 robust reading challenge on reading chinese text on signboard. In: 2019 international conference on document analysis and recognition (ICDAR). pp. 1577–1581. IEEE (2019)
45. Zhang, S., Liu, Y., Jin, L., Huang, Y., Lai, S.: Ensnet: Ensconce text in the wild. In: Proceedings of the AAAI conference on artificial intelligence. vol. 33, pp. 801–808 (2019)

46. Zhang, W., Zhu, J., Tai, Y., Wang, Y., Chu, W., Ni, B., Wang, C., Yang, X.: Context-aware image inpainting with learned semantic priors. arXiv preprint arXiv:2106.07220 (2021)