ViTEraser: Harnessing the Power of Vision Transformers for Scene Text Removal with SegMIM Pretraining

Dezhi Peng[1,3], **Chongyu Liu**[1], **Yuliang Liu**[4], **Lianwen Jin**[1,2,3,*]
[1]South China University of Technology
[2]SCUT-Zhuhai Institute of Modern Industrial Innovation
[3]INTSIG-SCUT Joint Lab of Document Image Analysis and Recognition
[4]Huazhong University of Science and Technology
pengdzscut@foxmail.com, eelwjin@scut.edu.cn

Abstract

Scene text removal (STR) aims at replacing text strokes in natural scenes with visually coherent backgrounds. Recent STR approaches rely on iterative refinements or explicit text masks, resulting in high complexity and sensitivity to the accuracy of text localization. Moreover, most existing STR methods adopt convolutional architectures while the potential of vision Transformers (ViTs) remains largely unexplored. In this paper, we propose a simple-yet-effective ViT-based text eraser, dubbed ViTEraser. Following a concise encoder-decoder framework, ViTEraser can easily incorporate various ViTs to enhance long-range modeling. Specifically, the encoder hierarchically maps the input image into the hidden space through ViT blocks and patch embedding layers, while the decoder gradually upsamples the hidden features to the text-erased image with ViT blocks and patch splitting layers. As ViTEraser implicitly integrates text localization and inpainting, we propose a novel end-to-end pretraining method, termed SegMIM, which focuses the encoder and decoder on the text box segmentation and masked image modeling tasks, respectively. Experimental results demonstrate that ViTEraser with SegMIM achieves state-of-the-art performance on STR by a substantial margin and exhibits strong generalization ability when extended to other tasks, *e.g.*, tampered scene text detection. Furthermore, we comprehensively explore the architecture, pretraining, and scalability of the ViT-based encoder-decoder for STR, which provides deep insights into the application of ViT to the STR field. Code is available at https://github.com/shannanyinxiang/ViTEraser.

Introduction

Scene text removal (STR) aims to realistically erase the text strokes in the wild by replacing them with visually plausible background, which has been widely applied to privacy protection (Inai et al. 2014), image editing (Wu et al. 2019), and image retrieval (Tursun et al. 2019a). Existing approaches to STR have evolved from the one-stage paradigm which implicitly integrates the text localization and background inpainting into a single network without the guidance of explicit text masks (Nakamura et al. 2017; Zhang et al. 2019b; Liu et al. 2020), to the two-stage framework which contains explicit text localizing processes and uses the resulting text masks to facilitate background inpainting (Tang et al. 2021; Lee and Choi 2022; Wang et al. 2023; Du et al. 2023b).

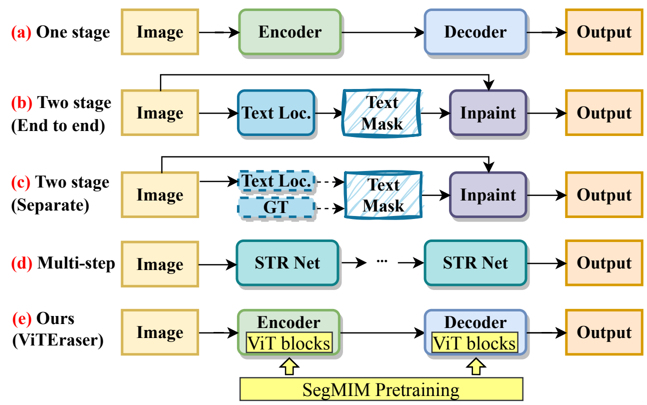


Figure 1: Comparison of ViTEraser with existing STR paradigms. Our method revisits the conventional single-step one-stage framework and improves it with ViTs for feature modeling and the proposed SegMIM pretraining. Dashed arrows indicate cutting off gradient flow. (Loc.: Localization)

Despite the great success achieved by previous methods, there still remain two critical issues. **(1)** The dominant two-stage methods suffer from the complex system design with two sub-tasks. The sequential text localizing and background inpainting pipeline introduces additional parameters, decreases the inference speed, and, more importantly, breaks the integrity of the entire model. The error of text localization can be easily propagated to the background inpainting, especially for the methods that require pre-supplied text detectors (Tang et al. 2021; Liu et al. 2022a; Lee and Choi 2022). **(2)** Recent advances (Liu et al. 2020; Lyu and Zhu 2022; Du et al. 2023b; Wang et al. 2023) tend to employ a multi-step paradigm in a coarse-to-fine or progressive fashion, which significantly undermines efficiency and makes it difficult to balance the parameters involved in multiple steps.

To this end, we revisit the one-stage paradigm and propose a novel simple-yet-effective ViT-based method for STR, termed as ViTEraser. Fig. 1 compares our method with existing STR approaches. The ViTEraser follows the conventional one-stage framework which comprises a single-step encoder-decoder network and is free of text mask in-

*Corresponding author

put or text localizing processes. This concise pipeline perfectly gets rid of the drawbacks of the two-stage and multi-step approaches mentioned above, but has been discarded in recent advances because of the unexpected artifacts and inexhaustive erasure issues caused by the implicit text localization mechanism. However, we argue that these limitations are actually due to the insufficient capacity of previous CNN-based architectures. Recently, vision Transformers (ViTs) (Dosovitskiy et al. 2021) have achieved incredible success on diverse computer vision tasks (Han et al. 2022) but are rarely investigated for STR. Nevertheless, ViT is perfect for STR since global information is indispensable for determining text locations and background textures, especially for large texts that existing STR systems still struggle with. Therefore, for the first time in the STR field, the proposed ViTEraser thoroughly utilizes ViTs for feature representation in both the encoder and decoder. Specifically, the encoder hierarchically maps the input image into the hidden space through ViT blocks and patch embedding layers, while the decoder gradually upsamples the hidden features to the text-erased image with ViT blocks and patch splitting layers. Thanks to its high generality, ViTEraser can be effortlessly integrated with various ViTs, *e.g.*, Swin Transformer (v2) (Liu et al. 2021, 2022b), PVT (Wang et al. 2021, 2022a).

Despite the powerful ViT-based structure, the implicit integration of text localizing and background inpainting still significantly challenges the model capacity of ViTEraser, requiring both high-level text perception and fine-grained pixel infilling abilities. However, the insufficient scale of existing STR datasets (Liu et al. 2020) limits the full learning of these abilities and makes the large-capacity ViT-based model prone to overfitting. To solve similar issues, pretraining plays a crucial role in a variety of fields (Kenton and Toutanova 2019; Xu et al. 2020; Yang et al. 2022) but is quite under-explored in the STR realm. Moreover, with the rapid development of large-scale scene text detection datasets and commercial optical character recognition (OCR) APIs, numerous real-world images with text bounding boxes are easily available. Therefore, we propose SegMIM which fully pretrains STR models using large-scale scene text detection data. Concretely, by assigning two pretraining tasks of text box segmentation and mask image modeling (MIM) (He et al. 2022; Xie et al. 2022) to the output features of the encoder and decoder, respectively, the STR performance can be effectively boosted with enhanced text localizing, inpainting, and global reasoning abilities.

Extensive experiments are conducted on two STR benchmarks including SCUT-EnsText (Liu et al. 2020) and SCUT-Syn (Zhang et al. 2019b). Furthermore, we comprehensively explore the architecture, pretraining, and scalability of the ViT-based encoder-decoder for STR. The experimental results demonstrate the clear superiority of ViTEraser with and without the SegMIM pretraining. Additionally, ViTEraser also achieves state-of-the-art performance on tampered scene text detection using the Tampered-IC13 (Wang et al. 2022b) dataset, exhibiting strong generalization ability.

In summary, the contributions of this paper are as follows.

- We propose a novel ViT-based one-stage method for STR, termed as ViTEraser. The ViTEraser adopts a concise single-step encoder-decoder paradigm, thoroughly integrating ViTs for feature representation in both the encoder and decoder.
- We propose SegMIM, a new pretraining scheme for STR. With SegMIM, ViTEraser acquires enhanced global reasoning capabilities, enabling it to effectively distinguish and generate text and background regions.
- We conduct a comprehensive investigation into the architecture, pretraining, and scalability of the ViT-based encoder-decoder for STR, which provides deep insights into the application of ViT to the STR field.
- The experiments demonstrate that ViTEraser achieves state-of-the-art performance on STR, and its potential for extension to other domains is also highlighted.

Related Work

Scene Text Removal

Scene text removal aims at realistically erasing the texts in natural scenes. Existing methods can be divided into one-stage and two-stage categories based on whether there are explicit text localizing processes.

One-stage methods follow a concise image-to-image translation pipeline, implicitly integrating text localizing and background inpainting procedures into a single network. Nakamura et al. (2017) pioneered in erasing texts at patch level using a convolution-to-deconvolution encoder-decoder structure. Inspired by Pix2Pix (Isola et al. 2017), Zhang et al. (2019b) proposed an end-to-end cGAN-based (Mirza and Osindero 2014) EnsNet which directly erases texts at image level. EraseNet (Liu et al. 2020) further improved EnsNet following a coarse-to-refine pipeline. From a data perspective, Jiang et al. (2022) proposed a controllable synthesis module based on EraseNet.

Two-stage methods decompose STR into the text localizing and background inpainting processes. The text localizing component produces explicit text masks which are fed into subsequent modules to facilitate background inpainting. The two-stage methods can be further divided into separate and end-to-end categories. The **separate two-stage methods** depend on separately trained text detectors (Zdenek and Nakayama 2020; Conrad and Chen 2021; Liu et al. 2022a) or ground truth (GT) (Qin, Wei, and Manduchi 2018; Tursun et al. 2019b; Tang et al. 2021; Lee and Choi 2022) to obtain text masks. In contrast, **end-to-end two-stage methods** end-to-end optimize the text localizing modules with other components (Keserwani and Roy 2021). Under this paradigm, recent advances tended to devise coarse-to-refine (Tursun et al. 2020; Du et al. 2023a) or progressive frameworks (Lyu and Zhu 2022; Bian et al. 2022; Du et al. 2023b; Wang et al. 2023) with text segmentation modules. On the contrary, Hou, Chen, and Wang (2022) expanded the width of the network in a multi-branch fashion. Additionally, Lyu et al. (2023) incorporated text segmentation maps at feature level using the proposed FET mechanism. Although the two-stage methods have dominated the STR field, they suffer from the high complexity caused by multiple modules and progressive erasing and are prone to text localizing accuracy.

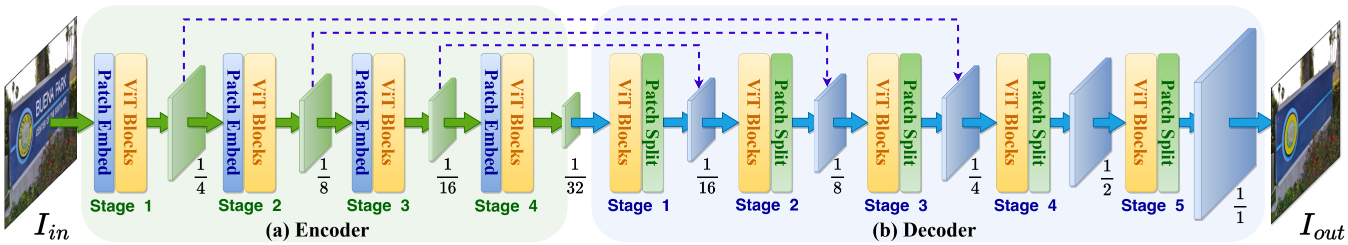


Figure 2: Overall architecture of ViTEraser. The ViTEraser follows the one-stage paradigm but is thoroughly equipped with ViTs, yielding a simple-yet-effective STR approach that is free of progressive refinements and text localizing processes.

Vision Transformer

The Transformer (Vaswani et al. 2017) was first proposed for natural language processing (Kenton and Toutanova 2019) but rapidly swept the computer vision field (Han et al. 2022). Early ViTs (Dosovitskiy et al. 2021; Touvron et al. 2021) first tokenized images with large window sizes and then kept the feature size throughout all Transformer layers. Recently, the research on ViTs has focused on producing pyramid feature maps, *e.g.*, PVT (Wang et al. 2021, 2022a), HVT (Pan et al. 2021), Swin Transformer (Liu et al. 2021, 2022b), and PiT (Heo et al. 2021). Nowadays, ViTs have played an important role in many tasks, such as object detection (Carion et al. 2020), semantic segmentation (Xie et al. 2021; Cao et al. 2022), text spotting (Peng et al. 2022; Liu et al. 2023), and document understanding (Xu et al. 2020).

ViTEraser

As shown in Fig. 2, we revisit the conventional single-step one-stage paradigm, getting rid of the complicated iterative refinement and the susceptibility to text localizing accuracy. The proposed ViTEraser pioneers in thoroughly employing ViTs instead of CNN in both the encoder and decoder, yielding a simple-yet-effective pipeline. Concretely, the encoder hierarchically maps the input image into the hidden space through successive ViT blocks and patch embedding layers, while the decoder gradually upsamples the hidden features to the text-erased image with successive ViT blocks and patch splitting layers. ViT blocks throughout the encoder-decoder provide sufficient global context information, enabling the implicit integration of text localization and background inpainting into a single network within a single forward pass. Moreover, lateral connections are devised between the encoder and decoder to preserve the input details.

Encoder

As shown in Fig. 2(a), the encoder of ViTEraser consists of four stages. Given an input image $I_{in} \in \mathbb{R}^{H \times W \times 3}$, the encoder hierarchically produces four feature maps $\{f_i^{enc}\}_{i=1}^4$ with strides of $\{2^{i+1}\}_{i=1}^4$ *w.r.t* the input image and channels of $\{C_i^{enc}\}_{i=1}^4$, respectively. Specifically, the i-*th* stage first downsamples the spatial size using a patch embedding layer and then captures global correlation through a stack of N_i^{enc} ViT blocks.

Patch Embedding Layer Given an input feature map $f_{in} \in \mathbb{R}^{h \times w \times c_{in}}$, a patch embedding layer with a downsampling ratio of d and an output channel of c_{out} first flattens each $d \times d$ patch, yielding a $\frac{h}{d} \times \frac{w}{d} \times (d^2 \times c_{in})$ feature map. Then a 1×1 convolution layer is applied to transform this intermediate feature map into the output $f_{out} \in \mathbb{R}^{\frac{h}{d} \times \frac{w}{d} \times c_{out}}$.

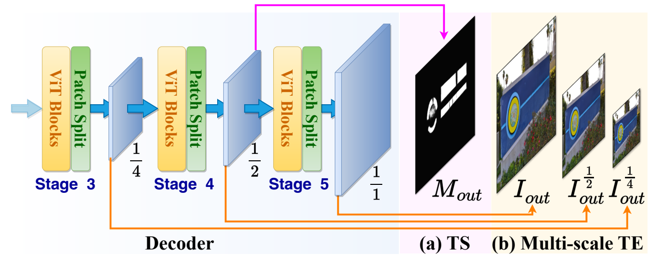


Figure 3: Auxiliary outputs of ViTEraser during training, including (a) text box segmentation map and (b) multi-scale text erasing results. (TS: Text Segmentation, TE: Text Erasing)

Decoder

The decoder contains five stages as illustrated in Fig. 2(b). Based on the final feature f_4^{enc} of the encoder, the decoder hierarchically generates five feature maps $\{f_i^{dec} \in \mathbb{R}^{\frac{H}{2^{5-i}} \times \frac{W}{2^{5-i}} \times C_i^{dec}}\}_{i=1}^5$. Concretely, in each stage, the feature is first processed with N_i^{dec} ViT blocks and then upsampled by 2 via a patch splitting layer. Moreover, lateral connections (Liu et al. 2020) are built between the features $\{f_i^{enc}\}_{i=1}^3$ of the encoder and the features $\{f_{4-i}^{dec}\}_{i=1}^3$ of the decoder. Finally, the text-erased image is predicted via a 3×3 convolution based on the feature $f_5^{dec} \in \mathbb{R}^{H \times W \times C_5^{dec}}$.

Patch Splitting Layer Patch splitting is designed as the inverse operation of the patched embedding to upsample the spatial size of features. Fed with an input feature map $f_{in} \in \mathbb{R}^{h \times w \times c_{in}}$, the patch splitting layer first decomposes each c_{in}-dimensional token into a 2×2 patch with $\frac{c_{in}}{4}$ dimension, expanding the input feature map to a shape of $2h \times 2w \times \frac{c_{in}}{4}$. After that, a 1×1 convolution layer is adopted to produce the output feature map $f_{out} \in \mathbb{R}^{2h \times 2w \times c_{out}}$.

Training

As depicted in Fig. 3, auxiliary outputs are produced during only training, including a text box segmentation map M_{out} and multi-scale text erasing results $\{I_{out}^{\frac{1}{2}}, I_{out}^{\frac{1}{4}}\}$. Specifically,

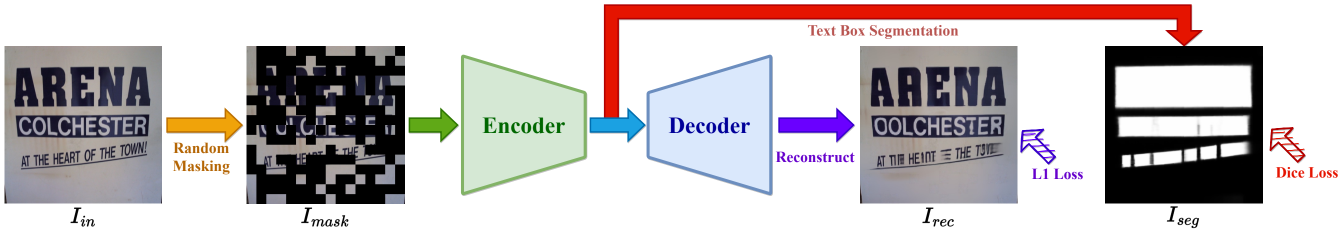


Figure 4: Pipeline of the proposed SegMIM pretraining. Given a randomly masked image, the text box segmentation and masked image modeling tasks are accomplished on top of the encoder and decoder, respectively.

M_{out} is predicted based on f_4^{dec} via a 3×3 deconvolution for $2\times$ upsampling and a 3×3 convolution with Sigmoid activation. Besides, $I_{out}^{\frac{1}{2}} \in \mathbb{R}^{\frac{H}{2} \times \frac{W}{2} \times 3}$ and $I_{out}^{\frac{1}{4}} \in \mathbb{R}^{\frac{H}{4} \times \frac{W}{4} \times 3}$ are generated from features f_4^{dec} and f_3^{dec}, respectively, each through a 3×3 convolution.

The model training adopts a GAN-based paradigm with a local-global discriminator (Liu et al. 2020). Given an input image with corresponding text-erased image and groud-truth (GT) text box mask, the losses comprise multi-scale reconstruction loss, perceptual loss, style loss, segmentation loss, and adversarial loss, following EraseNet (Liu et al. 2020).

SegMIM Pretraining

Unlike two-stage methods that utilize task-specific modules and training objectives, ViTEraser implicitly integrates text localizing and background inpainting tasks into a single encoder-decoder, thus facing challenges in fully learning to handle both tasks and susceptible to overfitting when scaled up. This limitation arises from the scarcity of training samples in STR and the high costs associated with annotating them. Moreover, enormous natural scene images with text bounding boxes are easily available as the advancing of scene text detection datasets and OCR APIs. To this end, leveraging the availability of extensive scene text detection datasets, we exploit large-scale pretraining techniques which have recently shown significant advancements (Yang et al. 2022; Xu et al. 2020) but rarely been investigated for STR.

Because STR can be decomposed into text localizing and background inpainting sub-tasks, we intuitively propose a new pretraining paradigm for STR, termed SegMIM, which focuses the encoder on text box segmentation and the decoder on masked image modeling (MIM) as shown in Fig. 4. Despite its simplicity, the clear advantages and interpretability of SegMIM are manifold. **(1)** The model learns the discriminative representation of texts and backgrounds through the text box segmentation task, which is crucial to STR. **(2)** The model learns the generative features of texts and backgrounds via MIM, enhancing the text perception and background recovery. **(3)** The global reasoning capacity is significantly improved due to the high mask ratio (0.6).

Architecture

The network architecture during pretraining inherits the encoder-decoder structure as shown in Fig. 2 but adds two extra heads for text box segmentation and image reconstruction, respectively. Given an input image $I_{in} \in \mathbb{R}^{H \times W \times 3}$, a binary mask $M_{mim} \in \mathbb{R}^{H \times W \times 1}$ is randomly generated following SimMIM (Xie et al. 2022). Then, the masked image I_{mask} combining I_{in} and M_{mim} is fed into the network.

Text Box Segmentation Head Based on the final feature $f_4^{enc} \in \mathbb{R}^{\frac{H}{32} \times \frac{W}{32} \times C_4^{enc}}$ of the encoder, a 1×1 convolution layer changes its dimension from C_4^{enc} to 1024. Subsequently, after transforming each 1024-dimension vector to a 32×32 patch and activating using a sigmoid function, a text box segmentation map $I_{seg} \in \mathbb{R}^{H \times W \times 1}$ is obtained.

Image Reconstruction Head Through a 3×3 convolution, a reconstructed image $I_{rec} \in \mathbb{R}^{H \times W \times 3}$ is predicted using the final feature $f_5^{dec} \in \mathbb{R}^{H \times W \times C_5^{dec}}$ of the decoder.

Optimization

The loss $\mathcal{L}_{pre}$ for pretraining is the sum of a text box segmentation loss $\mathcal{L}_{dice}$ and a MIM loss $\mathcal{L}_{mim}$ as follows.

$$\mathcal{L}_{pre} = \mathcal{L}_{dice} + \mathcal{L}_{mim}, \tag{1}$$

$$\mathcal{L}_{dice} = 1 - \frac{2\sum_{i,j} I_{seg(i,j)} \times S_{gt(i,j)}}{\sum_{i,j}(I_{seg(i,j)})^2 + \sum_{i,j}(S_{gt(i,j)})^2}, \tag{2}$$

$$\mathcal{L}_{mim} = ||\Psi(I_{rec}, M_{mim}) - \Psi(I_{in}, M_{mim})||_1, \tag{3}$$

where $S_{gt} \in \mathbb{R}^{H \times W \times 1}$ is the GT text box mask and the function Ψ fetches the image pixels at masked positions.

Experiments

Datasets

Scene Text Removal Datasets include **SCUT-EnsText** (Liu et al. 2020) and **SCUT-Syn** (Zhang et al. 2019b). Specifically, SCUT-EnsText is a real-world dataset containing 2,749 samples for training and 813 samples for testing. SCUT-Syn is a synthetic dataset with 8,000 and 800 samples for training and testing, respectively.

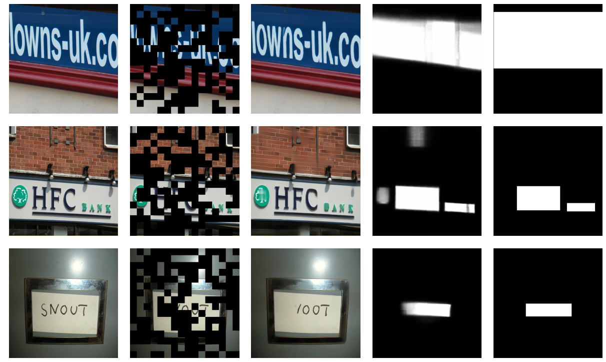
Pretraining Datasets include the training sets of **ICDAR2013** (Karatzas et al. 2013), **ICDAR2015** (Karatzas et al. 2015), **MLT2017** (Nayef et al. 2017), **ArT** (Chng et al. 2019), **LSVT** (Sun et al. 2019), and **ReCTS** (Zhang et al. 2019a), as well as the training and validating sets of **TextOCR** (Singh et al. 2021). After removing the overlapping samples with the test set of SCUT-EnsText (Liu et al. 2020), there are totally 88,340 valid samples for pretraining.

Encoder	Decoder	SCUT-EnsText			Params↓ (M)
		PSNR↑	MSSIM↑	MSE↓	
Conv	Deconv	35.05	97.20	0.0893	131.45
Conv+TE	Deconv	34.85	97.13	0.1043	133.04
Conv+TE	TD+Deconv	34.89	97.14	0.1007	139.43
Swinv2-Tiny	Deconv	36.06	97.40	0.0573	65.83
Swinv2-Tiny	TD+Deconv	35.92	97.39	0.0591	71.37
Swinv2-Tiny	MLP	26.18	81.07	0.3532	**28.21**
ViTEraser-Swinv2-Tiny		**36.32**	**97.48**	**0.0569**	65.39

Table 1: Comparison of different Transformer-based STR architectures.

Dataset	Encoder	Decoder	SCUT-EnsText		
			PSNR↑	MSSIM↑	MSE↓
×	×	×	33.34	96.70	0.1854
ImageNet-1k	CLS	×	36.55	97.56	0.0497
	SimMIM	×	36.38	97.51	0.0622
	CLS	CLS	36.54	97.55	0.0517
	CLS	SimMIM	36.45	97.55	0.0508
Scene Text Detection Dataset	Text Seg.	×	36.89	97.59	0.0490
	SimMIM	×	36.43	97.49	0.0554
	Text Seg.		36.78	97.57	0.0487
	SimMIM		36.68	97.58	0.0480
	SegMIM		**37.08**	**97.62**	**0.0447**

Table 2: Comparison of different pretraining strategies of ViTEraser-Swinv2-Small. (CLS: Classification)



(a) Input (b) Masked (c) Reconstruct (d) Pred. Mask (e) GT Mask

Figure 5: Visualizations of SegMIM. (Pred.: Predicted)

Architecture	Scale	SCUT-EnsText			Params↓ (M)
		PSNR↑	MSSIM↑	MSE↓	
ViTEraser (PVT)	Tiny	34.48	96.98	0.1251	**37.85**
	Small	35.03	97.15	0.1019	60.36
	Medium	35.09	97.16	0.1089	99.80
	Large	34.91	97.08	0.1183	134.13
ViTEraser (Swin)	Tiny	35.95	97.41	0.0647	65.26
	Small	35.81	97.44	0.0589	107.90
	Base	36.17	97.47	0.0637	191.66
ViTEraser (Swinv2)	Tiny	36.32	97.48	0.0569	65.39
	Small	**36.55**	**97.56**	**0.0497**	108.15
	Base	36.32	97.51	0.0565	191.97

Table 3: Comparison of different scales of ViTEraser.

Implementation Details

Network Architecture We explore three types of ViT blocks, *i.e.*, Pyramid Vision Transformer block (**PVT**), Swin Transformer block (**Swin**), and Swin Transformer v2 block (**Swinv2**), to implement the proposed ViTEraser. Based on the original scale settings of these ViTs (Wang et al. 2021; Liu et al. 2021, 2022b), we obtain four scales of PVT-based ViTEraser (**ViTEraser-PVT-Tiny/Small/Medium/Large**), three scales of Swin-based ViTEraser (**ViTEraser-Swin-Tiny/Small/Base**), and three scales of Swinv2-based ViTEraser (**ViTEraser-Swinv2-Tiny/Small/Base**). For conciseness, **ViTEraser refers to the Swinv2-based ViTEraser by default**.

Pretraining The input image is resized to 512×512. Random masking is performed on the input image with a ratio of 0.6 and a patch size of 32. Besides, a mask token is added to the encoder to represent the masked patches. Using 4 NVIDIA A6000 GPUs with 48GB memory, the network is pretrained for 100 epochs with an AdamW optimizer, a batch size of 64, and learning rates of 0.0001 before the 80*th* epoch and 0.00001 afterward. Because the mask token can negatively affect the encoder, the encoder will be finetuned solely with the text box segmentation task after end-to-end pretraining, following the training strategy of SimMIM. The finetuning lasts for 20 epochs with an initial learning rate of 0.00125 and a cosine decay learning rate schedule.

Training The training procedure on SCUT-EnsText or SCUT-Syn uses only its corresponding training set. The input size of the images is set to 512×512. The network is trained with an AdamW optimizer for 300 epochs using 2 NVIDIA A6000 GPUs with 48GB memory. The learning rate is initialized as 0.0005 and linearly decayed to 0.00001 at the last epoch. The training batch size is set to 16.

Evaluation Metrics

Following previous studies (Liu et al. 2020, 2022a), the image-eval metrics include PSNR, MSSIM, MSE, AGE, pEPs, pCEPs, and FID, while the detection-eval metrics involve the precision (P), recall (R), and f-measure (F) using the pretrained CRAFT (Baek et al. 2019) for text detection.

Experiments on Architecture

Which architecture is the best for the integration of Transformer into STR models? To answer this question, we first introduce the encoders and decoders compared in Tab. 1.
Encoder (1) *Conv* represents a ResNet50 (He et al. 2016). **(2)** *Conv+TE* indicates the concatenation of a ResNet50 and a 6-layer Transformer encoder with 256 channels. **(3)** *Swinv2-Tiny* is the tiny version of Swin Transformer v2 (Liu

Method	Image-Eval							Detection-Eval↓			Params↓ (M)	Speed↑ (fps)
	PSNR↑	MSSIM↑	MSE↓	AGE↓	pEPs↓	pCEPs↓	FID↓	R	P	F		
Original	-	-	-	-	-	-	-	69.5	79.4	74.1	-	-
Pix2pix (Isola et al. 2017)	26.70	88.56	0.37	6.09	0.0480	0.0227	46.88	35.4	69.7	47.0	54.42	<u>133</u>
STE (Nakamura et al. 2017)	25.47	90.14	0.47	6.01	0.0533	0.0296	43.39	5.9	40.9	10.2	-	-
EnsNet (Zhang et al. 2019b)	29.54	92.74	0.24	4.16	0.0307	0.0136	32.71	32.8	68.7	44.4	12.40	**199**
MTRNet++ (Tursun et al. 2020)	29.63	93.71	0.28	3.51	0.0305	0.0168	35.68	15.1	63.8	24.4	18.67	53
EraseNet (Liu et al. 2020)	32.30	95.42	0.15	3.02	0.0160	0.0090	19.27	4.6	53.2	8.5	17.82	71
SSTE (Tang et al. 2021)	35.34	96.24	0.09	-	-	-	-	3.6	-	-	30.75	7.8
PSSTRNet (Lyu and Zhu 2022)	34.65	96.75	0.14	1.72	0.0135	0.0074	-	5.1	47.7	9.3	**4.88**	56
CTRNet (Liu et al. 2022a)	35.20	97.36	0.09	2.20	0.0106	0.0068	13.99	1.4	38.4	2.7	159.81	5.1
GaRNet§ (Lee and Choi 2022)	35.45	97.14	0.08	1.90	0.0105	0.0062	15.50	1.6	42.0	3.0	33.18	22
MBE (Hou, Chen, and Wang 2022)	35.03	97.31	-	2.06	0.0128	0.0088	-	-	-	-	-	-
PEN (Du et al. 2023b)	35.21	96.32	0.08	2.14	0.0097	0.0037	-	2.6	33.5	4.8	-	-
PEN* (Du et al. 2023b)	35.72	96.68	0.05	1.95	0.0071	0.0020	-	2.1	**26.2**	3.9	-	-
PERT (Wang et al. 2023)	33.62	97.00	0.13	2.19	0.0135	0.0088	-	4.1	50.5	7.6	14.00	24
SAEN (Du et al. 2023a)	34.75	96.53	0.07	1.98	0.0125	0.0073	-	-	-	-	19.79	62
FETNet (Lyu et al. 2023)	34.53	97.01	0.13	1.75	0.0137	0.0080	-	5.8	51.3	10.5	<u>8.53</u>	77
ViTEraser-Tiny	36.32	97.48	0.0569	1.81	0.0073	0.0040	11.77	0.717	32.7	1.403	65.39	24
ViTEraser-Tiny + SegMIM	36.80	97.55	0.0491	1.79	0.0067	0.0036	10.79	<u>0.430</u>	<u>27.3</u>	<u>0.847</u>	65.39	24
ViTEraser-Small	36.55	97.56	0.0497	1.73	0.0072	0.0039	11.46	0.778	42.2	1.528	108.15	17
ViTEraser-Small + SegMIM	<u>37.08</u>	**97.62**	**0.0447**	**1.69**	**0.0064**	**0.0034**	<u>10.16</u>	<u>0.430</u>	30.9	0.848	108.15	17
ViTEraser-Base	36.32	97.51	0.0565	1.86	0.0074	0.0041	11.68	0.635	37.8	1.248	191.97	15
ViTEraser-Base + SegMIM	**37.11**	<u>97.61</u>	<u>0.0474</u>	<u>1.70</u>	<u>0.0066</u>	<u>0.0035</u>	**10.15**	**0.389**	29.7	**0.768**	191.97	15

Table 4: Comparison with state of the arts on SCUT-EnsText. (Bold: state of the art, underline: the second best)

et al. 2022b). The ResNet50 and Swinv2-Tiny are pretrained using ImageNet-1k (Deng et al. 2009).

Decoder (1) *Deconv* decoder hierarchically upsamples a 16×16 feature map with 2048 channels to sizes of $\{32, 64, 128, 256, 512\}$ and channels of $\{1024, 512, 256, 64, 64\}$ through five deconvolution layers. **(2)** *TD+Deconv* decoder adds a 6-layer Transformer decoder with 256 channels before a *Deconv* decoder. With 256 learnable queries, the Transformer decoder produces a hidden feature of 256 tokens and 256 channels which is subsequently resized to a $16 \times 16 \times 256$ feature map. This feature map then undergoes a 1×1 convolution with 2048 channels before being processed by the *Deconv* decoder. **(3)** *MLP* follows the decoder of SegFormer (Xie et al. 2021). The multi-scale features produced by the encoder are transformed to 256 channels, then interpolated to a size of 512×512, and finally fused via a 1×1 convolution.

Based on the results in Tab. 1, the discussions are as follows. **(1)** Inserting Transformer into CNNs can not effectively improve the STR results (2*nd* & 3*rd* rows *v.s.* 1*st* row). The Transformer encoder only performs global attention on high-level features produced by CNN, omitting fine-grained correlations such as detailed textures. Moreover, the learnable queries adopted in the Transformer decoder may cause spatial misalignment. **(2)** Pure ViT-based encoder (4*th* to 6*th* rows) makes a significant improvement. The window-based Swinv2-Tiny can effectively capture local and global dependencies at both low- and high-level feature spaces. **(3)** *The ViTEraser, which thoroughly utilizes ViTs in both the encoder and decoder, provides the best architecture for applying Transformer to STR with a substantial margin.* The Swinv2 blocks enable the decoder to fill the background considering both surrounding and long-distance context.

Experiments on Pretraining

In this section, we comprehensively explore pretraining schemes for STR based on ViTEraser. The pretraining strategies for comparison include: **(1)** When using ImageNet-1k, the encoder can be pretrained with the classification or SimMIM tasks. Moreover, because the first four stages of the decoder are empirically set to be symmetric to the encoder, their parameters can also be initialized by the encoder's pretrained weights symmetrically. **(2)** When using scene text detection datasets, the encoder or encoder-decoder can be pretrained with text box segmentation or SimMIM tasks.

The experiment results in Tab. 2 demonstrate that SegMIM achieves the best performance. Moreover, Fig. 5 illustrates the visualizations of SegMIM. It can be seen that the pretrained model can accurately determine text locations and realistically reconstruct masked patches.

Experiments on Scalability

We investigate the scalability of ViTEraser in Tab. 3. It can be seen that as the scale goes up, the performance tends to increase in general. However, for ViTEraser-Swinv2 and ViTEraser-PVT, the performance of the largest scale is inferior to a smaller one. This may be due to the overfitting caused by the dramatically increased parameters and limited training samples. However, the potential of large models can be stimulated when pretrained with SegMIM (Tab. 4).

(a) Input (b) GT (c) MTRNet++ (d) EraseNet (e) SSTE (f) GaRNet (g) CTRNet (h) PERT (i) ViTEraser

Figure 6: Qualitative comparison of existing methods and ViTEraser-Small (w/ SegMIM) on SCUT-EnsText.

Comparison with State of the Arts

SCUT-EnsText In Tab. 4, we compare ViTEraser with existing approaches on SCUT-EnsText. For a fair comparison, instead of using GT text box masks, MTRNet++ (Tursun et al. 2020) uses empty coarse masks and GaRNet (Lee and Choi 2022) uses the text box masks produced by pretrained CRAFT (Baek et al. 2019). All inference speeds are tested using an RTX3090 GPU with a batch size of 1, considering the time consumption of the model forward and post-processing. As for the model size, we calculate the number of minimum required parameters during inference. Besides, the parameters and time cost of external text detectors are considered for SSTE (Tang et al. 2021), GaRNet, and CTRNet (Liu et al. 2022a).

The quantitative results in Tab. 4 demonstrate the state-of-the-art performance of ViTEraser on real-world STR. For image-eval metrics, a substantial improvement can be observed over previous methods, *e.g.*, boosting PSNR from 35.72 dB to 37.11 dB. For detection-eval metrics, the recall and f-measure reach a milestone of lower than 1%, indicating nearly all the texts have been effectively erased. Especially for ViTEraser-Base with SegMIM, remarkably low recall (0.389%) and f-measure (0.768%) have been achieved. Moreover, SegMIM significantly boosts all three scales of ViTEraser, improving PSNRs of ViTEraser-Tiny, Small, and Base by 0.48, 0.53, and 0.79 dB, respectively.

The visualizations on SCUT-EnsText are shown in Fig. 6, qualitatively demonstrating the effectiveness of ViTEraser.

SCUT-Syn The quantitative and qualitative comparisons on synthetic SCUT-Syn are presented in Tab. 5 and Fig. 7, respectively. It can be observed that ViTEraser outperforms existing methods except for the MBE (Hou, Chen, and Wang 2022) that ensembles multiple STR networks.

(a) Input (b) GT (c) EraseNet (d) SSTE (e) ViTEraser

Figure 7: Qualitative comparison of previous methods and ViTEraser-Base (w/ SegMIM) on SCUT-Syn.

(a) Input (b) GT (c) Real (d) Tampered (e) Text

Figure 8: Visualization results on Tampered-IC13.

Extension to Tampered Scene Text Detection

To verify the generalization ability of ViTEraser, we extend it to the tampered scene text detection (TSTD) task (Wang et al. 2022b) that aims to localize both tampered and real texts from natural scenes. Using Tampered-IC13 dataset (Wang et al. 2022b), we train a ViTEraser-Tiny (w/o SegMIM) whose three-channel outputs correspond to the box-level segmentation of real texts, tampered texts, and both of

Method	PSNR↑	MSSIM↑	MSE↓	AGE↓	pEPs↓	pCEPs↓
Pix2pix (Isola et al. 2017)	26.76	91.08	0.27	5.47	0.0473	0.0244
STE (Nakamura et al. 2017)	25.40	90.12	0.65	9.49	0.0553	0.0347
EnsNet (Zhang et al. 2019b)	37.36	96.44	0.21	1.73	0.0069	0.0020
MTRNet++ (Tursun et al. 2020)	34.55	98.45	0.04	-	-	-
EraseNet (Liu et al. 2020)	38.32	97.67	0.02	1.60	0.0048	0.0004
Zdenek and Nakayama (2020)	37.46	93.64	-	-	-	-
Conrad and Chen (2021)	32.97	94.90	-	-	-	-
SSTE (Tang et al. 2021)	38.60	97.55	0.02	-	-	-
PSSTRNet (Lyu and Zhu 2022)	39.25	98.15	0.02	1.20	0.0043	0.0008
CTRNet (Liu et al. 2022a)	41.28	98.52	0.02	1.33	0.0030	0.0007
MBE (Hou, Chen, and Wang 2022)	**43.85**	**98.64**	-	**0.94**	**0.0013**	0.00004
PEN (Du et al. 2023b)	39.26	98.03	0.02	1.29	0.0038	0.0004
PEN* (Du et al. 2023b)	38.87	97.83	0.03	1.38	0.0041	0.0004
PERT (Wang et al. 2023)	39.40	97.87	0.02	1.41	0.0046	0.0007
SEAN (Du et al. 2023a)	38.63	98.27	0.03	1.39	0.0043	0.0004
FETNet (Lyu et al. 2023)	39.14	97.97	0.02	1.26	0.0046	0.0008
ViTEraser-Tiny	42.24	98.42	0.0112	1.23	0.0021	0.000020
ViTEraser-Tiny + SegMIM	42.40	98.44	0.0106	1.17	0.0018	0.000015
ViTEraser-Small	42.45	98.43	0.0109	1.19	0.0020	0.000019
ViTEraser-Small + SegMIM	42.66	98.49	0.0099	1.13	0.0016	0.000012
ViTEraser-Base	42.53	98.45	0.0102	1.19	0.0018	0.000016
ViTEraser-Base + SegMIM	42.97	98.55	**0.0092**	1.11	0.0015	**0.000011**

Table 5: Comparison with state of the arts on SCUT-Syn.

Method	Tampered Text			Real Text			mF↑
	R↑	P↑	F↑	R↑	P↑	F↑	
S3R (Wang et al. 2022b) + EAST	69.97	70.23	69.94	27.32	50.46	35.45	52.70
ViTEraser-Tiny + EAST	77.87	79.66	78.76	32.45	65.23	43.34	61.05
S3R (Wang et al. 2022b) + PSENet	79.43	79.92	79.67	41.89	61.56	49.85	64.76
ViTEraser-Tiny + PSENet	82.38	83.23	82.80	39.70	64.96	49.28	66.04
S3R (Wang et al. 2022b) + ContourNet	91.45	**86.68**	88.99	54.80	**77.88**	64.33	76.66
ViTEraser-Tiny + ContourNet	**92.62**	85.77	**89.06**	**56.84**	75.82	**64.97**	**77.02**

Table 6: Comparison with existing methods on Tampered-IC13. (mF: Average f-measure of real and tampered texts)

them, respectively. The dice losses on these three segmentation maps are utilized to optimize the network. Furthermore, to calculate the evaluation metrics including recall (R), precision (P), and f-measure (F) of tampered and real texts, we incorporate an EAST (Zhou et al. 2017), PSENet (Wang et al. 2019), or ContourNet (Wang et al. 2020) trained with Tampered-IC13 to produce text bounding boxes. Specifically, a bounding box will be regarded as tampered if more than 50% pixels within it are classified as tampered by ViTEraser. Similarly, the bounding boxes of real texts can also be determined. The quantitative performance is presented in Tab. 6 and the visualizations are shown in Fig. 8. It can be seen that ViTEraser can achieve state-of-the-art performance on Tampered-IC13, showing strong generalization potential.

Conclusion

In this paper, we propose a novel simple-yet-effective one-stage ViT-based approach for STR, termed ViTEraser. ViTEraser employs a concise encoder-decoder paradigm, eliminating the need for text localizing modules, external text detectors, and progressive refinements. Moreover, ViTEraser pioneers in thoroughly utilizing ViTs in place of CNNs in both the encoder and decoder, significantly enhancing the long-range modeling ability. Furthermore, we propose a novel pretraining scheme, called SegMIM, which focuses the encoder and decoder on the text box segmentation and MIM tasks, respectively. Without bells and whistles, the proposed method substantially outperforms previous STR approaches. ViTEraser also exhibits outstanding performance in tampered scene text detection, exhibiting strong generalization potential. Additionally, we comprehensively explore the architecture, pretraining, and scalability of ViT-based encoder-decoder for STR. We believe this study can inspire more research on ViT-based STR and contribute to the development of the unified model for pixel-level OCR tasks.

Acknowledgements

This research is supported in part by NSFC (Grant No.: 61936003), National Key Research and Development Program of China (2022YFC3301703), and INTSIG-SCUT

Joint Lab Fundation (CG-0274-200703).

References

Baek, Y.; Lee, B.; Han, D.; Yun, S.; and Lee, H. 2019. Character region awareness for text detection. In *Proc. CVPR*, 9365–9374.

Bian, X.; Wang, C.; Quan, W.; Ye, J.; Zhang, X.; and Yan, D.-M. 2022. Scene text removal via cascaded text stroke detection and erasing. *Comput. Vis. Media*, 8: 273–287.

Cao, H.; Wang, Y.; Chen, J.; Jiang, D.; Zhang, X.; Tian, Q.; and Wang, M. 2022. Swin-Unet: Unet-like pure Transformer for medical image segmentation. In *Proc. ECCV Workshop*, 205–218.

Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; and Zagoruyko, S. 2020. End-to-end object detection with Transformers. In *Proc. ECCV*, 213–229.

Chng, C. K.; Liu, Y.; Sun, Y.; Ng, C. C.; Luo, C.; Ni, Z.; Fang, C.; Zhang, S.; Han, J.; Ding, E.; et al. 2019. ICDAR2019 robust reading challenge on arbitrary-shaped text-RRC-ArT. In *Proc. ICDAR*, 1571–1576.

Conrad, B.; and Chen, P.-I. 2021. Two-stage seamless text erasing on real-world scene images. In *Proc. ICIP*, 1309–1313.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. ImageNet: A large-scale hierarchical image database. In *Proc. CVPR*, 248–255.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proc. ICLR*, 1–21.

Du, X.; Zhou, Z.; Zheng, Y.; Ma, T.; Wu, X.; and Jin, C. 2023a. Modeling stroke mask for end-to-end text erasing. In *Proc. WACV*, 6151–6159.

Du, X.; Zhou, Z.; Zheng, Y.; Wu, X.; Ma, T.; and Jin, C. 2023b. Progressive scene text erasing with self-supervision. *Comput. Vis. Image Underst.*, 233: 103712.

Han, K.; Wang, Y.; Chen, H.; Chen, X.; Guo, J.; Liu, Z.; Tang, Y.; Xiao, A.; Xu, C.; Xu, Y.; et al. 2022. A survey on vision Transformer. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(1): 87–110.

He, K.; Chen, X.; Xie, S.; Li, Y.; Dollár, P.; and Girshick, R. 2022. Masked autoencoders are scalable vision learners. In *Proc. CVPR*, 16000–16009.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proc. CVPR*, 770–778.

Heo, B.; Yun, S.; Han, D.; Chun, S.; Choe, J.; and Oh, S. J. 2021. Rethinking spatial dimensions of vision Transformers. In *Proc. ICCV*, 11936–11945.

Hou, Y.; Chen, J. J.; and Wang, Z. 2022. Multi-branch network with ensemble learning for text removal in the wild. In *Proc. ACCV*, 1333–1349.

Inai, K.; Pålsson, M.; Frinken, V.; Feng, Y.; and Uchida, S. 2014. Selective concealment of characters for privacy protection. In *Proc. ICPR*, 333–338.

Isola, P.; Zhu, J.-Y.; Zhou, T.; and Efros, A. A. 2017. Image-to-image translation with conditional adversarial networks. In *Proc. CVPR*, 1125–1134.

Jiang, G.; Wang, S.; Ge, T.; Jiang, Y.; Wei, Y.; and Lian, D. 2022. Self-supervised text erasing with controllable image synthesis. In *Proc. ACM MM*, 1973–1983.

Karatzas, D.; Gomez-Bigorda, L.; Nicolaou, A.; Ghosh, S.; Bagdanov, A.; Iwamura, M.; Matas, J.; Neumann, L.; Chandrasekhar, V. R.; Lu, S.; et al. 2015. ICDAR 2015 competition on robust reading. In *Proc. ICDAR*, 1156–1160.

Karatzas, D.; Shafait, F.; Uchida, S.; Iwamura, M.; i Bigorda, L. G.; Mestre, S. R.; Mas, J.; Mota, D. F.; Almazan, J. A.; and De Las Heras, L. P. 2013. ICDAR 2013 robust reading competition. In *Proc. ICDAR*, 1484–1493.

Kenton, J. D. M.-W. C.; and Toutanova, L. K. 2019. BERT: Pre-training of deep bidirectional Transformers for language understanding. In *Proc. NAACL-HLT*, 4171–4186.

Keserwani, P.; and Roy, P. P. 2021. Text region conditional generative adversarial network for text concealment in the wild. *IEEE Trans. Circuits Syst. Video Technol.*, 32(5): 3152–3163.

Lee, H.; and Choi, C. 2022. The surprisingly straightforward scene text removal method with gated attention and region of interest generation: A comprehensive prominent model analysis. In *Proc. ECCV*, 457–472.

Liu, C.; Jin, L.; Liu, Y.; Luo, C.; Chen, B.; Guo, F.; and Ding, K. 2022a. Don't forget me: Accurate background recovery for text removal via modeling local-global context. In *Proc. ECCV*, 409–426.

Liu, C.; Liu, Y.; Jin, L.; Zhang, S.; Luo, C.; and Wang, Y. 2020. EraseNet: End-to-end text removal in the wild. *IEEE Trans. Image Process.*, 29: 8760–8775.

Liu, Y.; Zhang, J.; Peng, D.; Huang, M.; Wang, X.; Tang, J.; Huang, C.; Lin, D.; Shen, C.; Bai, X.; and Jin, L. 2023. SPTS v2: Single-Point Scene Text Spotting. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(12): 15665–15679.

Liu, Z.; Hu, H.; Lin, Y.; Yao, Z.; Xie, Z.; Wei, Y.; Ning, J.; Cao, Y.; Zhang, Z.; Dong, L.; et al. 2022b. Swin Transformer v2: Scaling up capacity and resolution. In *Proc. CVPR*, 12009–12019.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021. Swin Transformer: Hierarchical vision Transformer using shifted windows. In *Proc. ICCV*, 10012–10022.

Lyu, G.; Liu, K.; Zhu, A.; Uchida, S.; and Iwana, B. K. 2023. FETNet: Feature erasing and transferring network for scene text removal. *Pattern Recognit.*, 109531.

Lyu, G.; and Zhu, A. 2022. PSSTRNet: Progressive segmentation-guided scene text removal network. In *Proc. ICME*, 1–6.

Mirza, M.; and Osindero, S. 2014. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*.

Nakamura, T.; Zhu, A.; Yanai, K.; and Uchida, S. 2017. Scene text eraser. In *Proc. ICDAR*, 832–837.

Nayef, N.; Yin, F.; Bizid, I.; Choi, H.; Feng, Y.; Karatzas, D.; Luo, Z.; Pal, U.; Rigaud, C.; Chazalon, J.; et al. 2017. ICDAR2017 robust reading challenge on multi-lingual scene text detection and script identification-RRC-MLT. In *Proc. ICDAR*, volume 1, 1454–1459.

Pan, Z.; Zhuang, B.; Liu, J.; He, H.; and Cai, J. 2021. Scalable vision Transformers with hierarchical pooling. In *Proc. ICCV*, 377–386.

Peng, D.; Wang, X.; Liu, Y.; Zhang, J.; Huang, M.; Lai, S.; Li, J.; Zhu, S.; Lin, D.; Shen, C.; et al. 2022. SPTS: Single-point text spotting. In *Proc. ACM MM*, 4272–4281.

Qin, S.; Wei, J.; and Manduchi, R. 2018. Automatic semantic content removal by learning to neglect. In *Proc. BMVC*, 1–12.

Singh, A.; Pang, G.; Toh, M.; Huang, J.; Galuba, W.; and Hassner, T. 2021. TextOCR: Towards large-scale end-to-end reasoning for arbitrary-shaped scene text. In *Proc. CVPR*, 8802–8812.

Sun, Y.; Ni, Z.; Chng, C.-K.; Liu, Y.; Luo, C.; Ng, C. C.; Han, J.; Ding, E.; Liu, J.; Karatzas, D.; et al. 2019. ICDAR 2019 competition on large-scale street view text with partial labeling-RRC-LSVT. In *Proc. ICDAR*, 1557–1562.

Tang, Z.; Miyazaki, T.; Sugaya, Y.; and Omachi, S. 2021. Stroke-based scene text erasing using synthetic data for training. *IEEE Trans. Image Process.*, 30: 9306–9320.

Touvron, H.; Cord, M.; Douze, M.; Massa, F.; Sablayrolles, A.; and Jégou, H. 2021. Training data-efficient image transformers & distillation through attention. In *Proc. ICML*, 10347–10357.

Tursun, O.; Denman, S.; Sivapalan, S.; Sridharan, S.; Fookes, C.; and Mau, S. 2019a. Component-based attention for large-scale trademark retrieval. *IEEE Trans. Inf. Forensic Secur.*, 17: 2350–2363.

Tursun, O.; Denman, S.; Zeng, R.; Sivapalan, S.; Sridharan, S.; and Fookes, C. 2020. MTRNet++: One-stage mask-based scene text eraser. *Comput. Vis. Image Underst.*, 201: 103066.

Tursun, O.; Zeng, R.; Denman, S.; Sivapalan, S.; Sridharan, S.; and Fookes, C. 2019b. MTRNet: A generic scene text eraser. In *Proc. ICDAR*, 39–44.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Proc. NeurIPS*, 30.

Wang, W.; Xie, E.; Li, X.; Fan, D.-P.; Song, K.; Liang, D.; Lu, T.; Luo, P.; and Shao, L. 2021. Pyramid vision Transformer: A versatile backbone for dense prediction without convolutions. In *Proc. ICCV*, 568–578.

Wang, W.; Xie, E.; Li, X.; Fan, D.-P.; Song, K.; Liang, D.; Lu, T.; Luo, P.; and Shao, L. 2022a. PVT v2: Improved baselines with pyramid vision Transformer. *Comput. Vis. Media*, 8(3): 415–424.

Wang, W.; Xie, E.; Li, X.; Hou, W.; Lu, T.; Yu, G.; and Shao, S. 2019. Shape robust text detection with progressive scale expansion network. In *Proc. CVPR*, 9336–9345.

Wang, Y.; Xie, H.; Wang, Z.; Qu, Y.; and Zhang, Y. 2023. What is the Real Need for Scene Text Removal? Exploring the Background Integrity and Erasure Exhaustivity Properties. *IEEE Trans. Image Process.*, 32: 4567–4580.

Wang, Y.; Xie, H.; Xing, M.; Wang, J.; Zhu, S.; and Zhang, Y. 2022b. Detecting tampered scene text in the wild. In *Proc. ECCV*, 215–232.

Wang, Y.; Xie, H.; Zha, Z.-J.; Xing, M.; Fu, Z.; and Zhang, Y. 2020. ContourNet: Taking a further step toward accurate arbitrary-shaped scene text detection. In *Proc. CVPR*, 11753–11762.

Wu, L.; Zhang, C.; Liu, J.; Han, J.; Liu, J.; Ding, E.; and Bai, X. 2019. Editing text in the wild. In *Proc. ACM MM*, 1500–1508.

Xie, E.; Wang, W.; Yu, Z.; Anandkumar, A.; Alvarez, J. M.; and Luo, P. 2021. SegFormer: Simple and efficient design for semantic segmentation with Transformers. In *Proc. NeurIPS*, 12077–12090.

Xie, Z.; Zhang, Z.; Cao, Y.; Lin, Y.; Bao, J.; Yao, Z.; Dai, Q.; and Hu, H. 2022. SimMIM: A simple framework for masked image modeling. In *Proc. CVPR*, 9653–9663.

Xu, Y.; Li, M.; Cui, L.; Huang, S.; Wei, F.; and Zhou, M. 2020. LayoutLM: Pre-training of text and layout for document image understanding. In *Proc. KDD*, 1192–1200.

Yang, M.; Liao, M.; Lu, P.; Wang, J.; Zhu, S.; Luo, H.; Tian, Q.; and Bai, X. 2022. Reading and writing: Discriminative and generative modeling for self-supervised text recognition. In *Proc. ACM MM*, 4214–4223.

Zdenek, J.; and Nakayama, H. 2020. Erasing scene text with weak supervision. In *Proc. WACV*, 2238–2246.

Zhang, R.; Zhou, Y.; Jiang, Q.; Song, Q.; Li, N.; Zhou, K.; Wang, L.; Wang, D.; Liao, M.; Yang, M.; et al. 2019a. ICDAR 2019 robust reading challenge on reading Chinese text on signboard. In *Proc. ICDAR*, 1577–1581.

Zhang, S.; Liu, Y.; Jin, L.; Huang, Y.; and Lai, S. 2019b. EnsNet: Ensconce text in the wild. In *Proc. AAAI*, 801–808.

Zhou, X.; Yao, C.; Wen, H.; Wang, Y.; Zhou, S.; He, W.; and Liang, J. 2017. EAST: An efficient and accurate scene text detector. In *Proc. CVPR*, 5551–5560.