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

Dezhi Peng1,2, Chongyang Liu1, Yuliang Liu4, Lianwen Jin1,2,3,*
1South China University of Technology
2SCUT-Zhuhai Institute of Modern Industrial Innovation
3INTSIG-SCUT Joint Lab of Document Image Analysis and Recognition
4Huazhong 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).

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 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)
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 Toutanov 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.
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).

ViT Eraser

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 ViT Eraser 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 ViT Eraser 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 \( \{ f_{enc}^{i} \}_{i=1}^{4} \) 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 \( 1 \times 1 \) convolution layer is applied to transform this intermediate feature map into the output \( f_{out} \in \mathbb{R}^{\frac{H}{d} \times \frac{W}{d} \times c_{out}} \).

Decoder

The decoder contains five stages as illustrated in Fig. 2(b). Based on the final feature \( f_{dec}^{4} \) of the encoder, the decoder hierarchically generates five feature maps \( \{ f_{dec}^{i} \}_{i=1}^{5} \). 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 \( \{ f_{dec}^{i-1} \}_{i=1}^{3} \) of the decoder. Finally, the text-erased image is predicted via a \( 3 \times 3 \) convolution based on the feature \( f_{5}^{dec} \).

Patch Splitting Layer

Patch splitting is designed as the inverse operation of the patch 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 \( 2 \times 2 \) patch with \( \frac{H}{2} \times \frac{W}{2} \) dimension, expanding the input feature map to a shape of \( 2H \times 2W \times \frac{c_{in}}{4} \). After that, a \( 1 \times 1 \) 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 and (b) multi-scale text erasing results. (TS: Text Segmentation, TE: Text Erasing)

Figure 2: Overall architecture of ViT Eraser. The ViT Eraser 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.

Figure 3: Auxiliary outputs of ViT Eraser during training, including (a) text box segmentation map and (b) multi-scale text erasing results. (TS: Text Segmentation, TE: Text Erasing)
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 $f^{enc}_{seg} \in \mathbb{R}^{H \times W \times C_{enc}}$ of the encoder, a $1 \times 1$ convolution layer changes its dimension from $C_{enc}$ to 1024. Subsequently, after transforming each 1024-dimension vector to a $32 \times 32$ 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 $3 \times 3$ convolution, a reconstructed image $I_{rec} \in \mathbb{R}^{H \times W \times 3}$ is predicted using the final feature $f^{dec}_{seg} \in \mathbb{R}^{H \times W \times C_{dec}}$ of the decoder.

Optimization

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

$$
L_{pre} = L_{dice} + L_{mim}, \quad (1)
$$

$$
L_{dice} = 1 - \frac{2 \sum_{i,j} I_{seg}(i,j) \times S_{gt}(i,j)}{\sum_{i,j} (I_{seg}(i,j))^2 + \sum_{i,j} (S_{gt}(i,j))^2}, \quad (2)
$$

$$
L_{mim} = ||\Psi(I_{rec}, M_{mim}) - \Psi(I_{in}, M_{mim})||_1, \quad (3)
$$

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

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.
finetuning lasts for 20 epochs with an initial learning rate of 0.0001 and then linearly decayed to 0.00001 afterward. Because the mask token can introduce the encoders and decoders compared in Tab. 1.

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 et al. 2021, 2022b), we obtain four scales of PVT-based ViTEraser (ViTEraser-PVT-Tiny/Small/Medium/Large), to implement the proposed ViTEraser. Based on the original scale settings of these ViTs (Wang et al. 2021; Liu et al. 2020, 2022a), 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 et al. 2021, 2022b), we obtain four scales of PVT-based ViTEraser (ViTEraser-PVT-Tiny/Small/Medium/Large), to implement the proposed ViTEraser. Based on the original scale settings of these ViTs (Wang et al. 2021; Liu et al. 2020, 2022a), 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.
Table 4: Comparison with state of the arts on SCUT-EnsText. (Bold: state of the art, underline: the second best)

<table>
<thead>
<tr>
<th>Method</th>
<th>Image-Eval</th>
<th>Detection-Eval</th>
<th>Params (M)</th>
<th>Speed (fps)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PSNR↑</td>
<td>MSSIM↑</td>
<td>MSE↑</td>
<td>AGE↓</td>
</tr>
<tr>
<td>Original</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Pix2pix (Isola et al. 2017)</td>
<td>26.70</td>
<td>88.56</td>
<td>0.37</td>
<td>6.09</td>
</tr>
<tr>
<td>STE (Nakamura et al. 2017)</td>
<td>25.47</td>
<td>90.14</td>
<td>0.47</td>
<td>6.01</td>
</tr>
<tr>
<td>EnsNet (Zhang et al. 2019b)</td>
<td>29.54</td>
<td>92.74</td>
<td>0.24</td>
<td>4.16</td>
</tr>
<tr>
<td>MTRNet++ (Tursun et al. 2020)</td>
<td>29.63</td>
<td>93.71</td>
<td>0.28</td>
<td>3.51</td>
</tr>
<tr>
<td>EraseNet (Liu et al. 2020)</td>
<td>32.30</td>
<td>95.42</td>
<td>0.15</td>
<td>3.02</td>
</tr>
<tr>
<td>SSTE (Tang et al. 2021)</td>
<td>35.34</td>
<td>96.24</td>
<td>0.09</td>
<td>-</td>
</tr>
<tr>
<td>PSSTRNet (Lyu and Zhu 2022)</td>
<td>34.65</td>
<td>96.75</td>
<td>0.14</td>
<td>1.72</td>
</tr>
<tr>
<td>CTRNet (Liu et al. 2022a)</td>
<td>35.20</td>
<td>97.36</td>
<td>0.09</td>
<td>2.20</td>
</tr>
<tr>
<td>GaNet (Lee and Choi 2022)</td>
<td>35.45</td>
<td>97.14</td>
<td>0.08</td>
<td>1.90</td>
</tr>
<tr>
<td>MBE (Hou, Chen, and Wang 2022)</td>
<td>35.03</td>
<td>97.31</td>
<td>-</td>
<td>2.06</td>
</tr>
<tr>
<td>PEN (Du et al. 2023b)</td>
<td>35.21</td>
<td>96.32</td>
<td>0.08</td>
<td>2.14</td>
</tr>
<tr>
<td>PEN* (Du et al. 2023b)</td>
<td>35.72</td>
<td>96.68</td>
<td>0.05</td>
<td>1.95</td>
</tr>
<tr>
<td>PERT (Wang et al. 2023)</td>
<td>33.62</td>
<td>97.00</td>
<td>0.13</td>
<td>2.19</td>
</tr>
<tr>
<td>SAEN (Du et al. 2023a)</td>
<td>34.75</td>
<td>96.53</td>
<td>0.07</td>
<td>1.98</td>
</tr>
<tr>
<td>FETNet (Lyu et al. 2023)</td>
<td>34.53</td>
<td>97.01</td>
<td>0.13</td>
<td>1.75</td>
</tr>
<tr>
<td>ViTEraser-Tiny</td>
<td>36.32</td>
<td>97.48</td>
<td>0.0569</td>
<td>1.81</td>
</tr>
<tr>
<td>ViTEraser-Tiny + SegMIM</td>
<td>36.80</td>
<td>97.55</td>
<td>0.0491</td>
<td>1.79</td>
</tr>
<tr>
<td>ViTEraser-Small</td>
<td>36.55</td>
<td>97.56</td>
<td>0.0497</td>
<td>1.73</td>
</tr>
<tr>
<td>ViTEraser-Small + SegMIM</td>
<td>37.08</td>
<td>97.62</td>
<td>0.0447</td>
<td>1.69</td>
</tr>
<tr>
<td>ViTEraser-Base</td>
<td>36.32</td>
<td>97.51</td>
<td>0.0565</td>
<td>1.86</td>
</tr>
<tr>
<td>ViTEraser-Base + SegMIM</td>
<td>37.11</td>
<td>97.61</td>
<td>0.0474</td>
<td>1.70</td>
</tr>
</tbody>
</table>

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 × 16 × 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 (2nd & 3rd rows v.s. 1st 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 (4th to 6th 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. 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).
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.

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
<table>
<thead>
<tr>
<th>Method</th>
<th>PSNR↑</th>
<th>MSSIM↑</th>
<th>MSE↓</th>
<th>AGE↓</th>
<th>pEPs↓</th>
<th>pCEPs↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pix2pix (Isola et al. 2017)</td>
<td>26.76</td>
<td>91.08</td>
<td>0.27</td>
<td>5.47</td>
<td>0.0473</td>
<td>0.0244</td>
</tr>
<tr>
<td>STE (Nakamura et al. 2017)</td>
<td>25.40</td>
<td>90.12</td>
<td>0.65</td>
<td>9.49</td>
<td>0.0553</td>
<td>0.0347</td>
</tr>
<tr>
<td>EnsNet (Zhang et al. 2019b)</td>
<td>37.36</td>
<td>94.44</td>
<td>0.21</td>
<td>1.73</td>
<td>0.0069</td>
<td>0.0020</td>
</tr>
<tr>
<td>MTRNet++ (Tursun et al. 2020)</td>
<td>34.55</td>
<td>98.45</td>
<td>0.04</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>EraseNet (Liu et al. 2020)</td>
<td>38.32</td>
<td>97.67</td>
<td>0.02</td>
<td>1.60</td>
<td>0.0048</td>
<td>0.0004</td>
</tr>
<tr>
<td>Zdenek and Nakayama (2020)</td>
<td>37.46</td>
<td>93.64</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Conrad and Chen (2021)</td>
<td>32.97</td>
<td>94.90</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>SST (Tang et al. 2021)</td>
<td>38.60</td>
<td>97.55</td>
<td>0.02</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>PSSTNet (Lu and Zhu 2022)</td>
<td>39.25</td>
<td>95.15</td>
<td>0.02</td>
<td>1.20</td>
<td>0.0043</td>
<td>0.0008</td>
</tr>
<tr>
<td>CTRNet (Liu et al. 2022a)</td>
<td>41.28</td>
<td>98.52</td>
<td>0.02</td>
<td>1.33</td>
<td>0.0030</td>
<td>0.0007</td>
</tr>
<tr>
<td>MBE (Hou, Chen, and Wang 2022)</td>
<td>43.85</td>
<td>98.64</td>
<td>-</td>
<td>0.94</td>
<td>0.0013</td>
<td>0.00004</td>
</tr>
<tr>
<td>PEN (Du et al. 2023b)</td>
<td>39.26</td>
<td>98.03</td>
<td>0.02</td>
<td>1.23</td>
<td>0.0021</td>
<td>0.00002</td>
</tr>
<tr>
<td>PEN (Du et al. 2023b)</td>
<td>38.87</td>
<td>97.83</td>
<td>0.03</td>
<td>1.38</td>
<td>0.0041</td>
<td>0.0004</td>
</tr>
<tr>
<td>PERT (Wang et al. 2023)</td>
<td>39.40</td>
<td>97.87</td>
<td>0.02</td>
<td>1.41</td>
<td>0.0046</td>
<td>0.0007</td>
</tr>
<tr>
<td>SEAN (Du et al. 2023a)</td>
<td>38.63</td>
<td>98.27</td>
<td>0.03</td>
<td>1.39</td>
<td>0.0043</td>
<td>0.0004</td>
</tr>
<tr>
<td>FETNet (Lyu et al. 2023)</td>
<td>39.14</td>
<td>97.97</td>
<td>0.02</td>
<td>1.26</td>
<td>0.0046</td>
<td>0.0008</td>
</tr>
<tr>
<td>ViTEraser-Tiny</td>
<td>42.42</td>
<td>98.42</td>
<td>0.0112</td>
<td>1.23</td>
<td>0.0021</td>
<td>0.00002</td>
</tr>
<tr>
<td>ViTEraser-Tiny + SegMIM</td>
<td>42.40</td>
<td>98.44</td>
<td>0.0106</td>
<td>1.17</td>
<td>0.0018</td>
<td>0.000015</td>
</tr>
<tr>
<td>ViTEraser-Small</td>
<td>42.45</td>
<td>98.43</td>
<td>0.0109</td>
<td>1.19</td>
<td>0.0020</td>
<td>0.000019</td>
</tr>
<tr>
<td>ViTEraser-Small + SegMIM</td>
<td>42.66</td>
<td>98.49</td>
<td>0.0099</td>
<td>1.13</td>
<td>0.0016</td>
<td>0.000012</td>
</tr>
<tr>
<td>ViTEraser-Base</td>
<td>42.53</td>
<td>98.45</td>
<td>0.0102</td>
<td>1.19</td>
<td>0.0018</td>
<td>0.000016</td>
</tr>
<tr>
<td>ViTEraser-Base + SegMIM</td>
<td>42.97</td>
<td>98.55</td>
<td>0.0092</td>
<td>1.11</td>
<td>0.0015</td>
<td>0.000011</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Method</th>
<th>Tampered Text</th>
<th>Real Text</th>
<th>mF↑</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R↑</td>
<td>P↑</td>
<td>F↑</td>
</tr>
<tr>
<td>S3R (Wang et al. 2022b) + EAST</td>
<td>69.97</td>
<td>70.23</td>
<td>69.94</td>
</tr>
<tr>
<td>ViTEraser-Tiny + EAST</td>
<td>77.87</td>
<td>79.66</td>
<td>78.76</td>
</tr>
<tr>
<td>S3R (Wang et al. 2022b) + PSENet</td>
<td>79.43</td>
<td>79.92</td>
<td>79.67</td>
</tr>
<tr>
<td>ViTEraser-Tiny + PSENet</td>
<td>82.38</td>
<td>83.23</td>
<td>82.80</td>
</tr>
<tr>
<td>S3R (Wang et al. 2022b) + ContourNet</td>
<td>91.45</td>
<td>86.68</td>
<td>88.99</td>
</tr>
<tr>
<td>ViTEraser-Tiny + ContourNet</td>
<td>92.62</td>
<td>85.77</td>
<td>89.06</td>
</tr>
</tbody>
</table>

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

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 Foundation (CG-0274-200703).

References


