Modeling Stroke Mask for End-to-End Text Erasing

Xiangcheng Du[1*], Zhao Zhou[1,2*], Yingbin Zheng[2], Tianlong Ma[1], Xingjiao Wu[1], Cheng Jin[1,3†]

[1]School of Computer Science, Fudan University, Shanghai, China [2]Videt Technology, Shanghai, China

[3]Key Laboratory of Culture & Tourism Intelligent Computing of Ministry of Culture & Tourism, Shanghai, China

Abstract

Scene text erasing aims to wipe text regions in scene images with reasonable background. Most previous approaches employ scene text detectors to assist localization of the text regions. However, detected text boxes contain both text strokes and background clutters, and directly inpainting on the whole boxes may remain text artifacts and make regions unnatural. In this paper, we present an end-to-end network that focuses on modeling text stroke masks that provide more accurate locations to compute erased images. The network consists of two stages, i.e., a basic network with stroke generation and a refinement network with stroke awareness. The basic network predicts the text stroke masks and initial erasing results simultaneously. The refinement network receives the masks as supervision to generate natural erased results. Experiments on both synthetic and real-world scene images demonstrate the effectiveness of our framework in producing high quality erasing results.

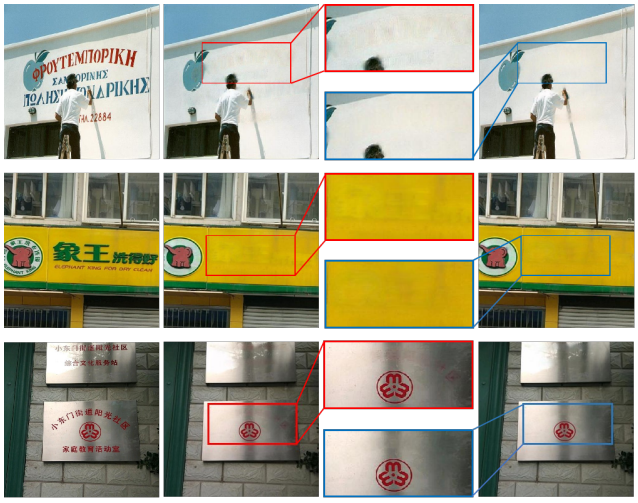


Figure 1. From left to right: original scene images, text erased images by text boxes-aware approach, and text erased images by the proposed method. Text boxes-aware results are unnatural due to redundant background content, and our method leverages text stroke masks to generate more natural results.

1. Introduction

Scene text contains numerous privacy information, and exposed information can easily be used for illegal purposes. Recently, scene text erasing technology has obtained much attention and is widely used in securing sensitive private information [13, 20]. The text erasing task aims to erase text and maintain original background content as possible, which is extremely challenging.

A popular practice is to decompose the text erasing into two subtasks, *i.e.*, text detection and image inpainting. The networks for these subtasks can be trained separately, and the text contents are erased within the text bounding boxes. With the development of recent text detectors and image inpainting, the approaches under this pipeline achieve good performance on the scene text erasing benchmarks.

However, due to the text boxes from text detectors containing both text strokes and background content, directly inpainting the whole boxes will also replace the background content, which may make the regions unnatural. Some erasing samples are illustrated in Figure 1. We believe accurate modeling of the real text segments would improve the visual effects of the erasing and reduce the artifacts.

In this paper, we propose a new erasing method called *Stroke-Aware Erasing Network* (SAEN), which is built upon the modeling of the text stroke mask to compute the erased images. Starting from a basic encoder-decoder network to generate an initial erased image, an additional decoder is connected after the basic encoder and designed to predict the text stroke masks which provide the text location information. Since the erased regions usually contain remnants of texts, we further introduce another encoder-decoder structure that considers the concatenation of predicted stroke and imprecise initial erased image as input, leading to the stroke-aware module to generate a more realistic and natural erased image.

*These authors contributed to the work equally.

†Corresponding author. Email: jc@fudan.edu.cn

Our contributions are summarized as follows:

- We propose an end-to-end stroke-aware erasing network, which can naturally and effectively erase scene texts. It could distinguish text strokes from the non-text areas and provide important location information for text erasure.
- We design a stroke-aware erasing module, which could enhance erasing performance. The module aims to erase residual text by combining the initial-erased results and stroke masks. Meanwhile, adversarial techniques are applied to generate natural images.
- Extensive evaluations demonstrate the superior performance of the proposed method. Notably, we achieve state-of-the-art results on SCUT-Syn and SCUT-EnsText datasets compared with the existing methods.

The rest of our paper is organized as follows. Section 2 briefly reviews the scene text erasing approach. In Section 3, we give the details of the proposed Stroke-Aware Erasing Network. We show the experimental results conducted on the datasets to demonstrate the effectiveness in Section 4. Section 5 concludes the paper.

2. Related Work

Early text erasing approaches mainly focuses on the text properties, such as color, fonts, and layout. The text regions are first extracted using a color histogram or threshold-based methods [7, 4] and then pixel similarity is calculated to replace the text area with the background. These methods are limited to complex scenarios. With the advancement of deep learning, methods based on deep inpainting and text detection are designed to handle scene text images.

Scene Text Eraser [13] is the first method that addresses the scene text erasing with the CNN. The method divides the image into small patches and uses a U-Net shaped neural network to erase the text. This patch-based processing fails to localize text with complex shapes and inevitably damaged the consistency of erased results. [19] adopts a separately trained scene text detector and an inpainting network. The text detector predicts segmentation maps of text instances that are fed into the inpainting network. The inpainting network fills in text regions and generates a final image. At inference time, the two modules are connected into one model for scene text erasing. [3] also adopts a two-stage approach and optimizes the mask generator and inpainting model separately. [17] introduces MTRNet that considers text region mask as an additional input into the network. However, the disadvantage of this method is that it requires text region masks as auxiliary information, which means that text removal cannot be guaranteed without additional information. There are also one-stage approaches that combine the two subtasks into a unified network. [20] designs an end-to-end architecture that employs a GAN with a UNet shaped encoder-decoder generator, and trains it with several losses. [10] proposes an end-to-end network that consists of a generator as well as a discriminator conditioned with the text regions. [18] introduces PERT that explicitly erases with a progressive strategy to remove texts accurately. [16] extends MTRNet, which is free from text location methods. [11] introduces the GAN-based EraseNet which adopts a coarse-to-fine erasure architecture with an additional segmentation head for learning text mask to help with text region localization. [2] proposes and end-to-end framework consists of a text stroke detection network and a text removal generation network. [9] uses text synthesis module for text erasing task. The synthesis module aims to generate training samples and learn a policy network by selecting more realistic training data.

In this paper, we also employ the one-stage strategy to avoid the complex training process for separate text detection and inpainting networks. Different from previous one-stage approaches, we focus on modeling the text stroke masks and finally achieve outstanding results. Very recently, [15] also utilizes the text stroke for erasing, where the text regions are firstly extracted by an off-the-shelf text detector [1]. [15] focuses the text erasing on the cropped text regions. The text detector and text eraser are trained separately, and the text erasing depends on the results of text detectors learned from extra datasets. In contrast, our method adopts an end-to-end framework, which reduces the complexity of the text erasure algorithm. without auxiliary images and annotations to train the text detector. Moreover, [15] needs the ground truth of the text stroke in the erasing scheme, which is a huge challenge for real-world datasets. In our framework, the text stroke is calculated from the original image and the ground truth without manual annotations. In addition, our method outperforms these state-of-the-art methods as shown in the experiments.

3. Method

In this section, we give the details of the proposed SAEN. As shown in Figure 2, the stroke-aware erasing network consists of a basic network, text stroke generation decoder, and the stroke-aware module. Here we focus on pixel-level text erasing, and introduce accurate text stroke masks to help to improve the performance of the process. The basic network is designed to generate initial erased images. Predicted text stroke can be fully exploited by merging with initial erase results. The stroke-aware module aims to erase the residual text and mask the text region more naturally. We start by introducing the basic network for stroke and initial erased image generation.

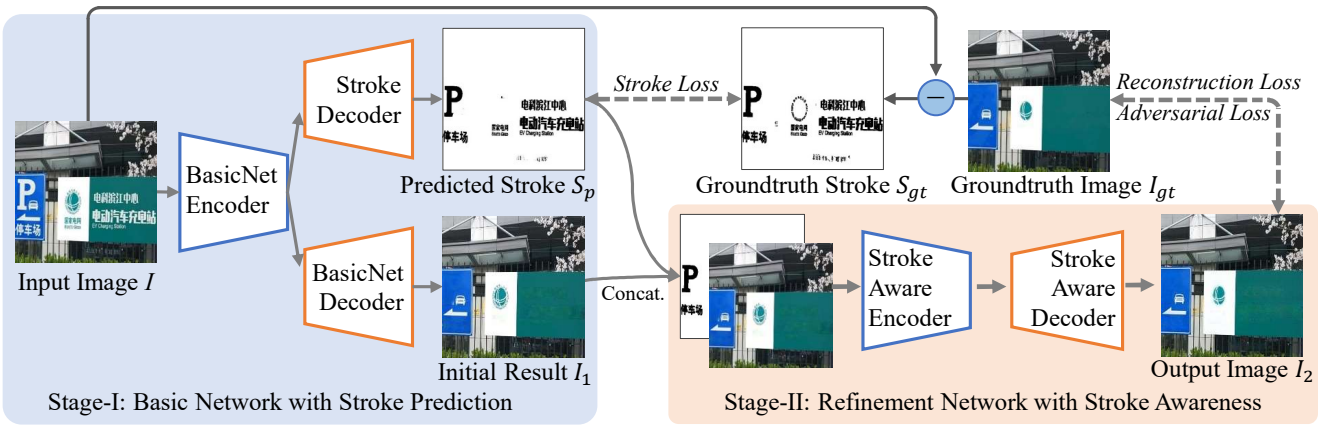


Figure 2. The pipeline of the proposed Stroke-Aware Erasing Network.

3.1. Basic Encoder-Decoder Structure

The basic network (BasicNet) employs the U-Net-like architecture composed of an encoder and decoder. The encoder follows the typical architecture of a convolutional network, composed of repeated convolutional layers and residual basic blocks. We control the stride parameter to reduce the resolution of the feature map. At each downsampling step, we double the number of feature channels and half the size of feature maps. The residual basic blocks follow ResNet [6] to retain more shallow information. The decoder consists of five up-sampling transposed convolutional layers with kernel size, stride step, and padding size set to 3, 2, 1, respectively. Additionally, the features from shallow layers are concated with the high-level ones to recover the structure and the texture details. The architecture of the BasicNet is shown in Figure 3(a). Given the original image I, the initial erased image I_1 is obtained by going through the basic network Θ_{bn}, *i.e.*,

$$I_1 = \Theta_{bn}(I) \tag{1}$$

3.2. Stroke Decoder

To generate the text stroke mask, a second stroke decoder is connected after the basic encoder and designed with the similar structure of the BasicNet decoder. The learning target of this module, *i.e.*, the ground-truth text stroke mask S_{gt}, is computed by subtracting original image I from ground-truth image I_{gt} and then thresholding. Specifically, When the absolute value of the difference is less than the set threshold, we set the value of S_{gt} at the corresponding position to 1; otherwise, the value is 0. Different threshold values result in generating different strokes (see Figure 4), which will further affects the erasing result directly; we will discuss the choice of this parameter in the evaluations.

The inputs of the stroke decoder are the output feature maps from the 6-th residual block. We introduce 3×3 convolutional operation between each transposed convolutional layer in the stroke decoder to reduce dimension and improve the representation ability of the features. The up-sampled features maps are concated with the corresponding ones from the down-sampling step. With the combination of BasicNet encoder and stroke decoder (Θ_{sd}), the text stroke S_p can be predicted:

$$S_p = \Theta_{sd}(I) \tag{2}$$

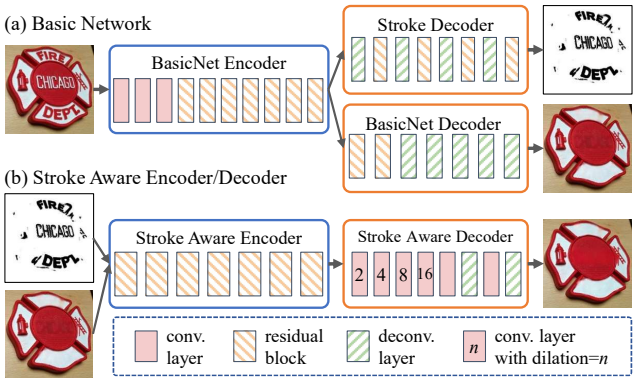


Figure 3. Network structures for the encoders and decoders.

3.3. Modeling Stroke Mask

The initial erased image I_1 has an approximate result, but sometimes retains slight text residue and the whole image is intuitively unnatural. The factors leading to this situation can be attributed to the lack of sufficient supervision. To address the issue and improve the erasing performance, we jointly model stroke mask S_p and initial output I_1.

As shown in Figure 2, we design a stroke-aware encoder-decoder module as the second stage. The stroke obtained from the stroke decoder assists in text localization for high-quality erased results. The encoder receives the initial erased results I_1 concat with the text stroke S_p as input. The encoder obtains more accurate text stroke distribution

Original Image | Threshold=20 | Threshold=25 | Threshold=30 | Threshold=35 | Threshold=40 | Threshold=45

Figure 4. Text strokes obtained by different threshold.

in an incremental manner by combining the predicted stroke masks and thus can remove residual text effectively. The stroke-aware module Θ_{sam} returns the final erased image I_2, which has the same resolution as I_1, *i.e.*,

$$I_2 = \Theta_{sam}\left(I_1, S_p\right) \tag{3}$$

3.4. Model Training

Several loss terms are employed to train a robust erasing model. We adopt the reconstruction loss to measure pixel-wise similarity between the ground-truth and predicted erased images. Since the task focus on the text regions, we set different weights for text stroke pixels and background pixels and formulate the reconstruction loss as

$$\begin{aligned} L_r = \tfrac{1}{N}\sum_{k=1}^{N}\{\left\|(I_{gt}^k - I_2^k) * (1 - S_{gt}^k)\right\|_1 + \\ \lambda_r \left\|(I_{gt}^k - I_2^k) * S_{gt}^k\right\|_1\} \end{aligned} \tag{4}$$

where I_{gt}^k and S_{gt}^k are the k-th ground-truth erased image and stroke mask, I_2^k is the k-th predicted erased image, N is the image number, $*$ denotes element-wise multiplication, and λ_r is the weights for the non-stroke pixels.

The stroke loss is then designed for the predictions of the stroke mask. Text stroke usually occupies a small proportion in entire scene images. Therefore, to reduce the imbalance issue, dice [12] between the ground-truth stroke S_{gt} and prediction stroke S_p are computed. The stroke loss is calculated as

$$L_s = \frac{1}{N}\sum_{k=1}^{N}\{1 - \mathrm{Dice}(S_p^k, S_{gt}^k)\} \tag{5}$$

$$\mathrm{Dice}(S_p^k, S_{gt}^k) = \frac{2 \cdot \sum_{i,j}(S_p^k[i,j] \cdot S_{gt}^k[i,j])}{\sum_{i,j}(S_p^k[i,j])^2 + \sum_{i,j}(S_{gt}^k[i,j])^2} \tag{6}$$

where $S_*^k[i,j]$ denotes the value of stroke mask S_*^k at the coordinate (i,j).

In addition, we utilize the adversarial loss L_a to generate realistic and natural erased images, which is formulated as

$$L_a = -\frac{1}{N}\sum_{k=1}^{N} D(I_2) \tag{7}$$

where $D(I)$ is an one-dimensional output of discriminator D denoting the probability of an erased image I as fake generated images. Here we follow the design in [11] and D is an extra classification network with VGG structure. We train the discriminator iteratively together with the proposed erasing network with a loss function $L_D = \frac{1}{N}\sum_k\{\mathrm{ReLU}(1 - D(I_{gt}^k)) + \mathrm{ReLU}(1 + D(I_2^k))\}$.

Finally, the overall loss function L is expressed as a combination of these losses, *i.e.*,

$$L = L_r + \lambda_s L_s + \lambda_a L_a \tag{8}$$

Implementation Details. Our text-erased model is implemented using PyTorch [14]. We resize images to 512×512 and normalize the pixel values to the range of (0, 1) with random horizontal flip and rotation. The default threshold is 35 for ground-truth stroke generation. For the loss function, we set $\lambda_r = 0.2$, $\lambda_s = 0.2$, and $\lambda_a = 0.01$. The network is initialized using a gaussian distribution and the Adam optimizer is used with $\beta = (0.5, 0.9)$. The initial learning rate is set to 10^{-4}. We set the β of the discriminator D to (0.0, 0.9) with learning rate of 10^{-5}. For other settings of D, we follow [11]. The training process of the proposed stroke-aware network reaches convergence after 300 epochs.

4. Experiment

Datasets We evaluate the proposed stroke-aware network on both the synthesis and real-world scene image datasets. For the synthesis data, we use the SCUT-Syn dataset [11], which is generated by text synthesis technology [5]. The

Table 1. Comparisons with previous approaches on the SCUT-Syn and SCUT-EnsText datasets. The bold and underline results denote the best performer and the runner-up in each metric, respectively.

SCUT-EnsText	PSNR↑	SSIM↑	MSE↓	AGE↓	pEPS↓	pCEPS↓
SceneTextEraser [13]	25.47	90.14	0.0047	6.01	0.0533	0.0296
Pix2Pix [8]	26.70	88.56	0.0037	6.09	0.0480	0.0270
EnsNet [20]	29.54	92.47	0.0024	4.16	0.0307	0.0136
EraseNet [11]	32.30	95.42	0.0015	3.02	0.0160	0.0090
CRAFT+SBSTE [15]	**35.34**	96.24	0.0009	-	-	-
PERT [18]	33.25	**96.95**	0.0014	2.18	0.0136	0.0088
SAEN [This paper]	34.75	96.53	**0.0007**	**1.98**	**0.0125**	**0.0073**

SCUT-Syn	PSNR↑	SSIM↑	MSE↓	AGE↓	pEPS↓	pCEPS↓
SceneTextEraser [13]	14.68	46.13	0.7148	13.29	0.1859	0.0936
Pix2Pix [8]	25.60	89.86	0.2465	5.60	0.0571	0.0423
EnsNet [20]	37.36	96.44	0.0021	1.73	0.0069	0.0020
MTRNet [17]	29.71	94.43	0.0004	-	-	-
MTRNet++ [16]	34.55	**98.45**	0.0004	-	-	-
EraseNet [11]	38.32	97.67	**0.0002**	1.60	0.0048	**0.0004**
Weak Supervision [19]	37.46	93.64	-	-	-	-
PERT [18]	**39.40**	97.87	**0.0002**	1.41	0.0045	0.0006
SAEN [This paper]	38.63	98.27	0.0003	**1.39**	**0.0043**	**0.0004**

(a) SCUT-Syn (b) SCUT-EnsText

Figure 5. Sample images and the ground-truth erased images.

background images are collected from ICDAR 2013 and ICDAR MLT-2017 and the texts are manually erased. The dataset contains 8,000 training images and 800 testing images, and all the training and test images are resized to 512×512. Some examples of the SCUT-Syn dataset are shown in Figure 5(a).

For the real-world scene images, we evaluate the performance on the SCUT-EnsText dataset [11]. The dataset contains 2,749 training and 813 testing images, which are selected from public scene text detection benchmarks. The dataset takes into account the diversity of text types, including text fonts, text orientations, text shapes and scene diversity. The erased ground-truth is computed by manually erasing all text instances using Adobe Photoshop. Figure 5(b) shows some examples of the dataset.

Evaluation metrics We adopted the image inpainting metrics to evaluate our method. The first set of metrics measure the statistics between the ground-truth and predicted image and is widely used in image processing tasks, including PSNR (peak signal-to-noise ratio), MSE (mean squared error), and SSIM (structural similarity). We also employ three metrics that focus on pixel-level information: AGE that computes the average of gray-level absolute difference, pEPS to calculate the percentage of error pixels, and pCEPS to calculate percentage of four-connected neighbors error

Table 2. Comparison between different settings of the network on SCUT-EnsText. (a) The effect of different stages. (b) The effect of stroke mask modeling, where Box-aware indicates the network replacing the stroke units with text box mask. (c) Different terms in loss functions. (d) The results of the model trained with different thresholds for the stroke generation.

Settings		PSNR↑	SSIM↑	MSE↓
(a)	BasicNet only	29.18	94.54	0.0026
	Stage-I only	32.27	95.07	0.0017
	Stage-I+Stage-II	34.75	96.53	0.0007
	SAEN+	34.91	96.64	0.0006
(b)	Box-aware	33.26	95.81	0.0008
	Stroke-aware	34.75	96.53	0.0007
(c)	L_r	30.74	94.85	0.0019
	L_r, L_s	33.94	95.76	0.0009
	L_r, L_s, L_a	34.75	96.53	0.0007
(d)	Threshold=20	33.27	95.39	0.0010
	Threshold=25	33.86	95.87	0.0008
	Threshold=30	34.40	96.15	0.0008
	Threshold=35	34.75	96.53	0.0007
	Threshold=40	34.56	96.27	0.0007
	Threshold=45	34.38	96.33	0.0009

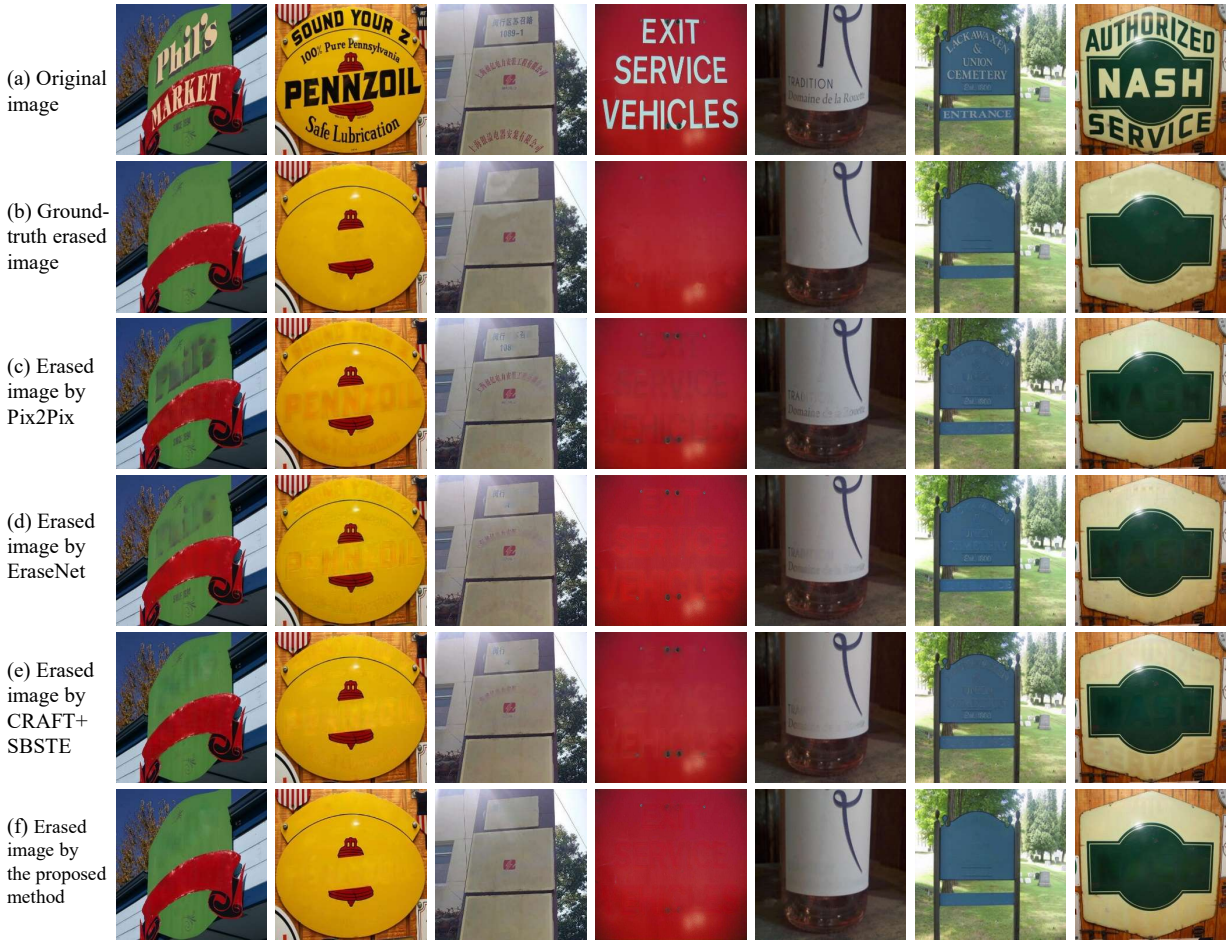


Figure 6. Erasing results of the proposed approach compared with some previous methods on real-world scene images.

pixels. A higher value of PSNR and SSIM or lower MSE, AGE, pEPs, and pCEPS represent better results.

4.1. Comparison with State-of-the-arts

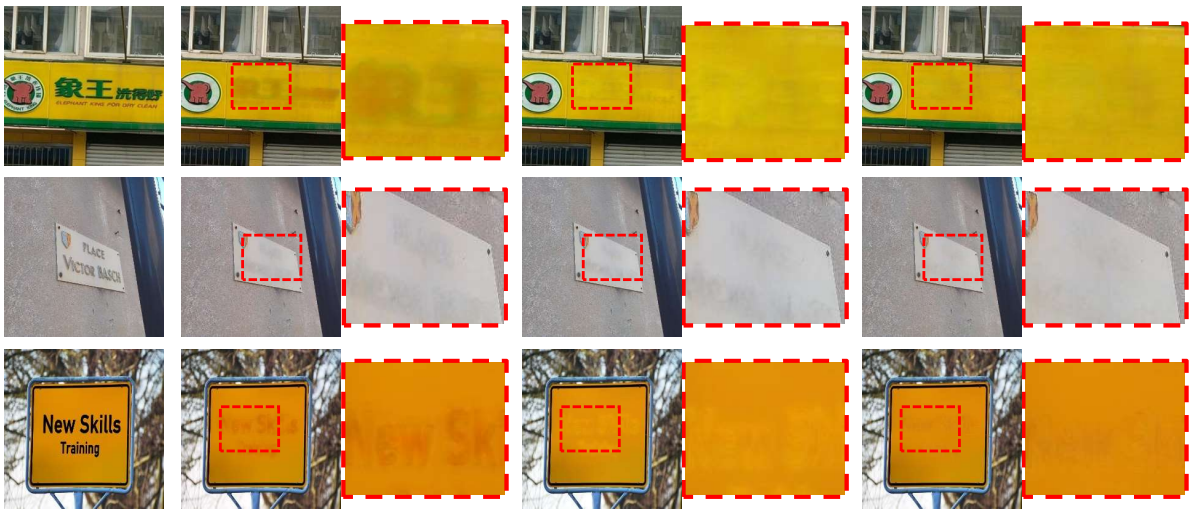
Table 1 summarizes all metrics for both datasets. We compare SAEN with several baselines and state-of-the-art approaches. Among them, the first group contains the well-known general image inpainting approach (Pix2Pix). We also compare ours with the two-stage methods (SceneTextEraser, CRAFT+SBSTE), the one-stage methods (EnsNet, MTRNet, MTRNet++, EraseNet), as well as the recent progressive network (PERT).

For both the synthesis data and real-world images, SAEN can reach a comparable results under different metrics. Compared with the other one-stage erasing methods, SAEN shows very competitive performance with PSNR of 34.75 (SCUT-EnsText) and 38.63 (SCUT-Syn), which outperform the other one-stage erasing methods and is significantly better for the real-world images. Compared with the two-stage method [15] that also employ the stroke information, our method performs better five metrics and achieves comparable results on PSNR, while ours do not rely on the text detector trained from auxiliary data. We also compare SAEN with recent progressive method [18]. Our network is more compact and we can achieve a comparable performance. The results confirm the effectiveness of the proposed text stroke modeling approach for erasing texts from scene images. Figure 6 shows some text erased images from different methods from the SCUT-EnsText dataset.

4.2. Ablation Study

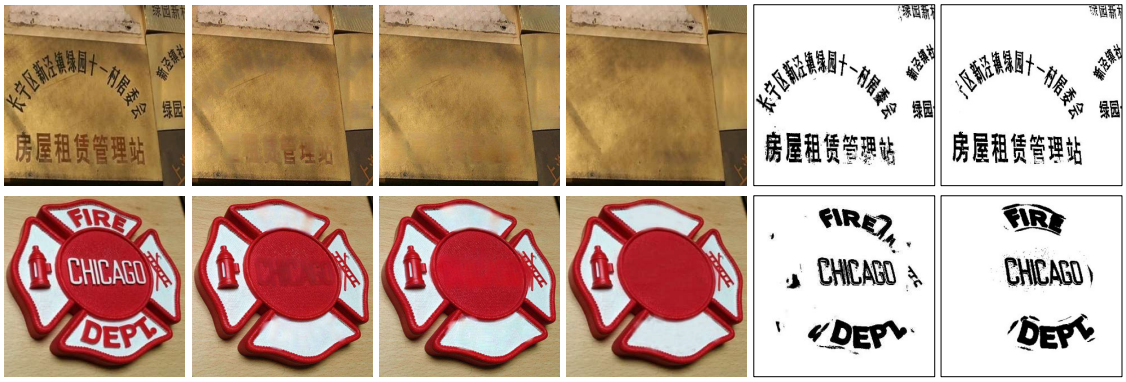
In this subsection, we evaluate the alternative implementation and parameters for constructing the stroke-aware erasing network. We report results on the SCUT-EnsText dataset as it is more challenging than the other.

Network components. In Table 2(a), we first evaluate the effect of different network components. Using the baseline encoder-decoder structure without stroke module, the eraser reaches a PSNR of 29.18 and SSIM of 94.93. Incorporating the stroke decoder boosts the PSNR with a margin of 3.09, indicating that the representation ability of the BasicNet encoder is improved with the help of stroke decoder.



(a) Input images (b) Erased images by stage-I only (c) Erased images with box-aware model (d) Erased images with full model

Figure 7. Qualitative comparisons with different network settings.



Original image | Erased image after stage-I | Erased image after stage-II | Ground-truth erased image | Predicted text stroke | Ground-truth text stroke

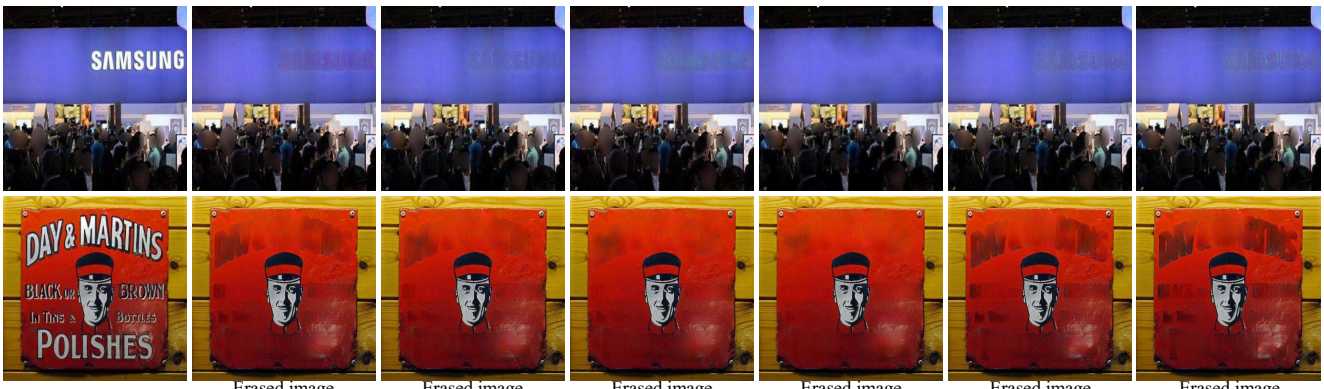
Figure 8. Outputs from different stages of the proposed network.

When we leverage the stroke-aware encoder and decoder in stage-II, we obtain a PSNR of 34.75 and SSIM of 96.53. To understand the effect of stage-II, we conduct qualitative comparisons on Figure 7 and observe that the erased images by full model are more clean and natural than the results by the model with only stage-I. Moreover, we also add another refinement network (with the same structure of stage-II) and denote it as SAEN+. The images from SAEN+ reach slightly better erasing results than those from stage-II, while the overall performance tends to be saturated. In order to make a good trade-off between performance and model efficiency, we adapt the two-stage model in this work.

The importance of stroke mask. The outputs from different stages in SAEN are further visualized in Figure 8. We can see that it is hard for a single network to erase the text completely since the stroke information is embedded implicitly. The stroke decoder can generate masks that consist of most real text strokes as well as some text-like edges and symbols. With the stroke-aware module, the residual texts are then erased and the refined results can be obtained.

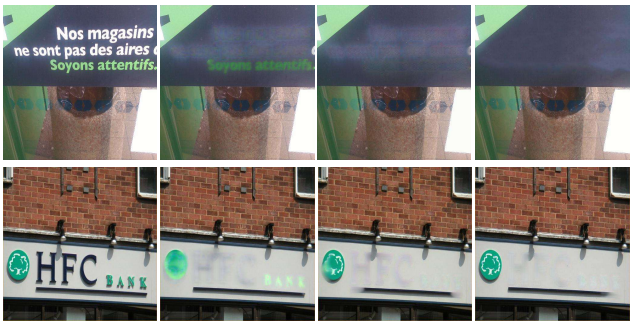
Recall that previous methods [19, 15] utilize text box masks to locate text and then enhance the performance. We believe the stroke generator helps to provide clues to the erased texts. Therefore, we quantitatively compared the results of box mask aware and text stroke aware on text erasing. Table 2(b) shows that the stroke-aware erasing model has better performance than the box mask aware method. It has gained 1.49 and 0.72 improvements on PSNR and SSIM compared with box-aware network. Some qualitative erased results are illustrated in Figure 7(c,d).

Loss function. The weights of the full model are learned by three terms, *i.e.*, the reconstruction loss, the stroke loss, and the adversarial loss, as described in Section 3.4. Adding the adversarial loss to the loss function leads to a slight improvement of the metrics (*e.g.*, 0.81 for PSNR and 0.77 for SSIM, see Table 2(c)). More importantly, from the qualita-



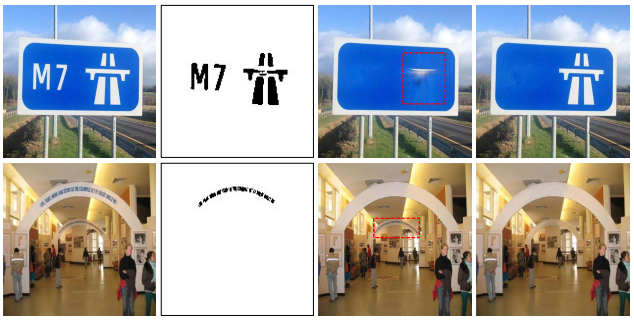
Input image | Erased image Threshold=20 | Erased image Threshold=25 | Erased image Threshold=30 | Erased image Threshold=35 | Erased image Threshold=40 | Erased image Threshold=45

Figure 9. Qualitative comparisons with the model trained with different thresholds for the stroke generation.



Input image | L_r only | $L_r + L_s$ | $L_r + L_s + L_a$

Figure 10. Erased images with different loss functions.



Input image | Predicted stroke | Erased result | Ground-truth

Figure 11. Samples of failure cases of our method. The failures are highlighted with red bounding box.

tive comparisons we can see that the erased images are more realistic and the generated regions are more smooth. When considering the reconstruction loss only, the network degenerates to a stack of two encoder-decoder modules. The results (PSNR 30.74, SSIM 94.85) are much lower than the full model but close to the result of BasicNet only, which also validate the importance of stroke in our framework.

Threshold for generating stroke mask. We compare the performance of our frameworks with different settings of the threshold for generating stroke masks. As illustrated in Figure 4, a low threshold usually contains redundant background, while some text strokes are filtered out with a high threshold. According to the results shown in Table 2(d), under the same experimental setting, threshold=35 is a suitable trade-off for the erasing. Figure 9 shows the outputs of SAEN with different thresholds. With the appropriate threshold, SAEN can restore a natural background.

Failure case analysis. There are generally two common types of failure text erasing. The first is the false-erased text-like symbols, as illustrated in the top row of Figure 11. We also observe that some small curve text regions are ignored, due to failure of the stroke prediction.

5. Conclusions

We introduced SAEN, a simple yet effective framework for scene text erasing. SAEN is an end-to-end framework that consists of a basic erasing network and a text stroke decoder, as well as the stroke-aware module. In particular, the basic network first uses a shared encoder to learn the representation and feed into two decoders that generate the predicted stroke and initial erased image, respectively. The stroke-aware module aims to repair the imprecise initial erased image with more discriminative features. This is achieved by concatenating the initial erased image and predicted stroke and then sending them to the module. We conduct extensive experiments on SCUT-Syn and SCUT-EnsText for both synthesis and real-world scene text erasing. The results demonstrate the effectiveness of the proposed approach.

Acknowledgement. This work was supported by National Archives Administration of China Research Program (2021-X-25), Inovation Program of Ministry of Culture and Tourism (CXGCXM2021-039), and Shanghai Archives Research Program (2108).

References

[1] Youngmin Baek, Bado Lee, Dongyoon Han, Sangdoo Yun, and Hwalsuk Lee. Character region awareness for text detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 9365–9374, 2019.

[2] Xuewei Bian, Chaoqun Wang, Weize Quan, Juntao Ye, Xiaopeng Zhang, and Dong-Ming Yan. Scene text removal via cascaded text stroke detection and erasing. *Computational Visual Media*, 8(2):273–287, 2022.

[3] Benjamin Conrad and Pei-I Chen. Two-stage seamless text erasing on real-world scene images. In *IEEE International Conference on Image Processing*, pages 1309–1313, 2021.

[4] Boris Epshtein, Eyal Ofek, and Yonatan Wexler. Detecting text in natural scenes with stroke width transform. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2963–2970, 2010.

[5] Ankush Gupta, Andrea Vedaldi, and Andrew Zisserman. Synthetic data for text localisation in natural images. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2315–2324, 2016.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[7] Weilin Huang, Yu Qiao, and Xiaoou Tang. Robust scene text detection with convolution neural network induced mser trees. In *European Conference on Computer Vision*, pages 497–511, 2014.

[8] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1125–1134, 2017.

[9] Gangwei Jiang, Shiyao Wang, Tiezheng Ge, Yuning Jiang, Ying Wei, and Defu Lian. Self-supervised text erasing with controllable image synthesis. *arXiv preprint arXiv:2204.12743*, 2022.

[10] Prateek Keserwani and Partha Pratim Roy. Text region conditional generative adversarial network for text concealment in the wild. *IEEE Transactions on Circuits and Systems for Video Technology*, 2021.

[11] Chongyu Liu, Yuliang Liu, Lianwen Jin, Shuaitao Zhang, Canjie Luo, and Yongpan Wang. Erasenet: End-to-end text removal in the wild. *IEEE Transaction on Image Processing*, 29:8760–8775, 2020.

[12] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *International Conference on 3D Vision*, pages 565–571, 2016.

[13] Toshiki Nakamura, Anna Zhu, Keiji Yanai, and Seiichi Uchida. Scene text eraser. In *IAPR International Conference on Document Analysis and Recognition*, pages 832–837, 2017.

[14] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32, 2019.

[15] Zhengmi Tang, Tomo Miyazaki, Yoshihiro Sugaya, and Shinichiro Omachi. Stroke-based scene text erasing using synthetic data for training. *IEEE Transaction on Image Processing*, 30:9306–9320, 2021.

[16] Osman Tursun, Simon Denman, Rui Zeng, Sabesan Sivapalan, Sridha Sridharan, and Clinton Fookes. Mtrnet++: One-stage mask-based scene text eraser. *Computer Vision and Image Understanding*, 201:103066, 2020.

[17] Osman Tursun, Rui Zeng, Simon Denman, Sabesan Sivapalan, Sridha Sridharan, and Clinton Fookes. Mtrnet: A generic scene text eraser. In *IAPR International Conference on Document Analysis and Recognition*, pages 39–44, 2019.

[18] Yuxin Wang, Hongtao Xie, Shancheng Fang, Yadong Qu, and Yongdong Zhang. Pert: A progressively region-based network for scene text removal. *arXiv preprint arXiv:2106.13029*, 2021.

[19] Jan Zdenek and Hideki Nakayama. Erasing scene text with weak supervision. In *IEEE Winter Conference on Applications of Computer Vision*, pages 2238–2246, 2020.

[20] Shuaitao Zhang, Yuliang Liu, Lianwen Jin, Yaoxiong Huang, and Songxuan Lai. Ensnet: Ensconce text in the wild. In *AAAI Conference on Artificial Intelligence*, pages 801–808, 2019.