Exploring Stroke-Level Modifications for Scene Text Editing

Yadong Qu,[1] Qingfeng Tan,[2*] Hongtao Xie,[1] Jianjun Xu,[1] Yuxin Wang,[1] Yongdong Zhang[1]

[1] University of Science and Technology of China, [2] GuangZhou University
{qqqyd, xujj1998, wangyx58}@mail.ustc.edu.cn, tqf528@gzhu.edu.cn, {htxie, zhyd73}@ustc.edu.cn

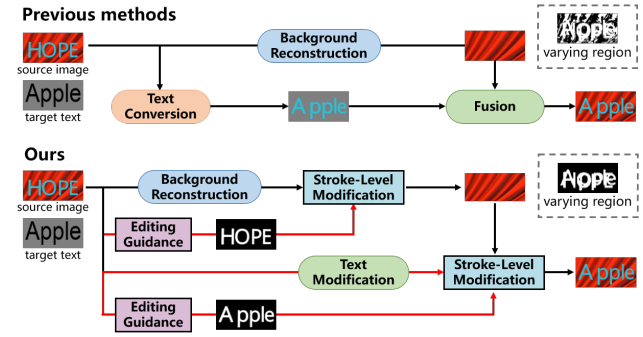
Abstract

Scene text editing (STE) aims to replace text with the desired one while preserving background and styles of the original text. However, due to the complicated background textures and various text styles, existing methods fall short in generating clear and legible edited text images. In this study, we attribute the poor editing performance to two problems: 1) Implicit decoupling structure. Previous methods of editing the whole image have to learn different translation rules of background and text regions simultaneously. 2) Domain gap. Due to the lack of edited real scene text images, the network can only be well trained on synthetic pairs and performs poorly on real-world images. To handle the above problems, we propose a novel network by MOdifying Scene Text image at strokE Level (MOSTEL). Firstly, we generate stroke guidance maps to explicitly indicate regions to be edited. Different from the implicit one by directly modifying all the pixels at image level, such explicit instructions filter out the distractions from background and guide the network to focus on editing rules of text regions. Secondly, we propose a Semi-supervised Hybrid Learning to train the network with both labeled synthetic images and unpaired real scene text images. Thus, the STE model is adapted to real-world datasets distributions. Moreover, two new datasets (Tamper-Syn2k and Tamper-Scene) are proposed to fill the blank of public evaluation datasets. Extensive experiments demonstrate that our MOSTEL outperforms previous methods both qualitatively and quantitatively. Datasets and code will be available at https://github.com/qqqyd/MOSTEL.

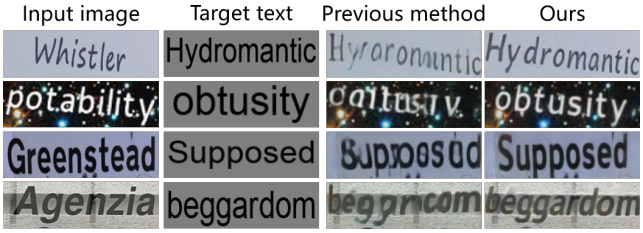
Introduction

As an emerging task in recent years, scene text editing (STE) has received increasing attention. It is designed to replace text in a scene image with new text while maintaining the original background textures and text styles (*e.g.* font, color, size, spatial transformation, etc.). STE can convert any word in an image to a desired one within a second and retain visual consistency before and after tampering, eliminating the need to spend hours manually editing images. The edited images can be used as augmented data to train the scene text detector (Qin et al. 2021; Qu et al. 2022) and recognizer (Sheng, Chen, and Xu 2019; Du et al. 2022). Compared with existing synthesis methods (Jaderberg et al. 2014), which simply place font-specific text on random background images, STE can better simulate various text styles and provide more reliable training images. It can also be used in sensitive data protection, smart city (Qiu et al. 2019) and augmented reality translation (Fragoso et al. 2011).

STE task has two main difficulties: 1) It is challenging to simulate various text styles while keeping the background textures intact. 2) The dearth of real-world training pairs causes the domain gap between synthetic training data and real scene text images. As shown in Fig. 1(a), previous methods (Wu et al. 2019; Yang, Huang, and Lin 2020) divide the STE task into three simpler subtasks: background reconstruction, text conversion and fusion. However, they address



(a)



(b)

Figure 1: The comparison between MOSTEL and previous methods. (a) is the pipeline. The dashed boxes indicate all the changed pixels throughout the editing process. (b) is the qualitative examples of previous method and ours.

*Corresponding author.

STE as a vanilla image-to-image translation task and directly use labeled synthetic images to train these modules. This implicit guidance by modifying all the pixels at image level aggravates the learning difficulty of editing rules. While learning how to edit texts, networks still strive to distinguish and preserve the regions that need to be retained. This way of implementing two tasks in a single module distracts the network from concentrating on the editing rules in text regions, resulting in illegible typeface and poor imitation of styles (Fig. 1(b)). In addition, the divide-and-conquer methods require supervision on each sub-network. The absence of corresponding labels on real-world images allows the network only to be trained with synthetic images, causing domain bias in real scene text images during test stage.

In this paper, we take a further step towards accurate scene text editing and propose a novel framework by MOdifying Scene Text image at strokE Level (MOSTEL). First, as shown in Fig. 1(a), we propose the editing guidance to predict the guidance maps to indicate the regions to be edited. They guide stroke-level modifications by explicitly filtering out invariant background regions. Compared with previous implicit guidance, our explicit one has two advantages. On the one hand, the editing guidance explicitly decomposes the modification of text regions and the maintenance of background regions, enabling the network to focus on the editing rules of text regions. By eliminating the distractions from invariant background, it considerably decreases the learning difficulty of editing rules and ensures the consistency of generated text styles with the original. On the other hand, as shown in the dash boxes in Fig. 1(a), previous implicit methods still cause subtle changes to the background regions. Since the invariant background is directly inherited from the source image, MOSTEL can maintain the integrity of the original image background to the greatest extent.

Second, a Semi-supervised Hybrid Learning is proposed to bridge the domain gap between training data and real-world test data. In the training stage, we introduce the unpaired scene text images by converting the text to itself. Specifically, we adopt the erase-and-write paradigm and propose Background Reconstruction Module and Text Modification Module to remove the unnecessary intermediate results. The real-world training images are first erased to generate text-free background, and then the same text with imitated style is rewritten onto the reconstructed background. In such a training scheme, to avoid the model degenerating into an identity mapping network, several countermeasures are adopted to separate the background and text, which are described in detail in Methodology. Therefore, we successfully adapt STE model to real-world scene text datasets distributions. Moreover, a scene text recognizer is employed in the training stage to ensure the generated text images clear and legible.

In addition, two new STE datasets named Tamper-Syn2k and Tamper-Scene are proposed to evaluate the performance of scene text editors. As far as we know, they are the first publicly available STE evaluation datasets, which will significantly facilitate a fair comparison of STE methods and promote the development of STE task. Extensive experiments on these datasets also demonstrate the superiority of our method both quantitatively and qualitatively.

Our contributions are summarized as follows:

- We propose a novel framework to perform stroke-level modifications, which explicitly guides the network to focus on the learning of editing rules in text regions and maximizes the integrity of background regions.
- We design a Semi-supervised Hybrid Learning that enables the model to be trained using both labeled synthetic images and unpaired real-world images.
- Two new STE datasets (Tamper-Syn2k and Tamper-Scene) are released to ensure a fair comparison of STE methods and promote the development of STE task.
- MOSTEL achieves promising performance in both quality and quantity. Our simple and powerful model will provide many insights for future STE works.

Related Work

Text Image Synthesis

Text image synthesis is a major trick for training robust DNN models. Several attempts have been made to generate synthetic text images in order to improve the accuracy of text detection and recognition. For example, (Jaderberg et al. 2014) use a word generator to insert texts into semantically sensible regions of background images. (Zhan, Lu, and Xue 2018) take semantic coherence, visual attention and adaptive text appearance into account to generate realistic synthetic text images. Most recently, text image synthesis methods have advanced significantly as a result of the development of Variational Auto Encoder (VAE) (Kingma and Welling 2013) and Generative Adversarial Networks (GAN) (Goodfellow et al. 2014). (Yang et al. 2019b) achieve real-time control of the crucial stylistic degree of the glyph via an adjustable parameter. (Sun et al. 2017) use a VAE structure to implement a stylized Chinese character generator. Meanwhile, (Azadi et al. 2018) propose an end-to-end solution to synthesize ornamented glyphs from images of several reference glyphs in the same style.

Style Transfer

Image style transfer is another challenging related task that aims to transfer visual style from a reference image to another target image. Most existing methods apply the encoder-decoder architecture that embeds the input into a subspace and then decodes it to generate desired images. (Isola et al. 2017) implement a learnable mapping from the input image to the output image using a training set of aligned image pairs. (Zhu et al. 2017) introduce cycle consistency loss to generalize the mapping relationship to unpaired cross-domain data. Similar ideas have been applied to various tasks, such as generating photos from sketches and face synthesis from attribute and semantic information. (Yang et al. 2017) are the first to apply style transfer methods to text images. They analyze and model the distance-based essential characteristics of text effects and leverage them to guide the synthesis process. Meanwhile, (Yang et al. 2019a) accomplish both the objective of style transfer and style removal by using stylization and de-stylization sub-networks.

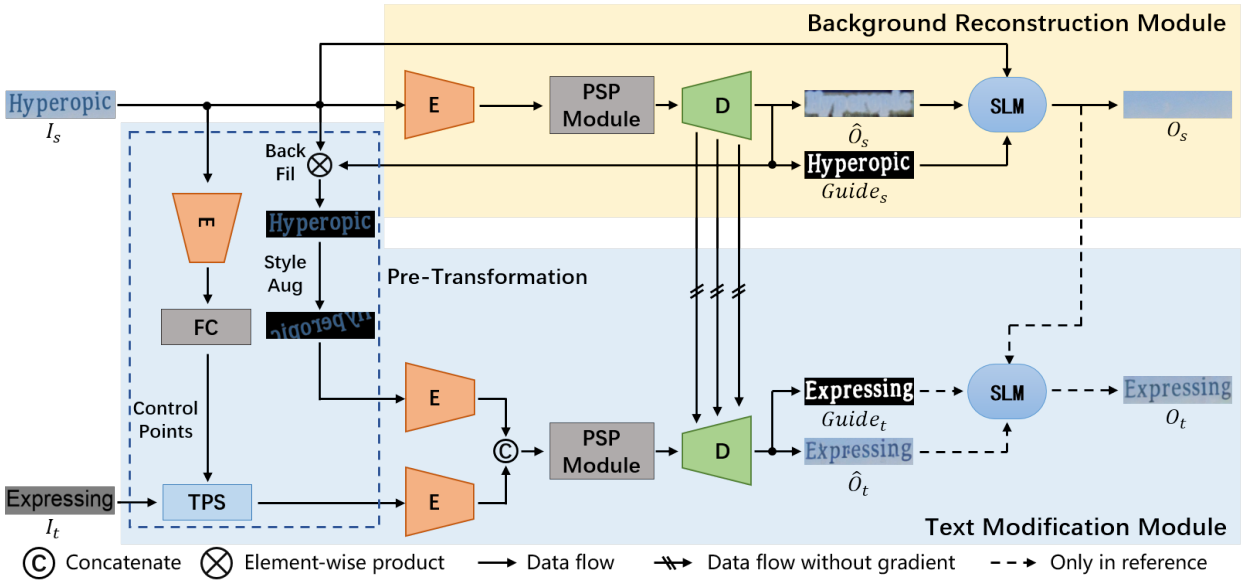


Figure 2: The overall pipeline of MOSTEL. The network consists of Background Reconstruction Module and Text Modification Module. Back Fil, Style Aug and SLM mean Background Filtering, Style Augmentation and Stroke-Level Modification.

Scene Text Editing

Due to the wide range of applications, GAN-based scene text editing methods attract increasing research interest. STEFANN (Roy et al. 2020) designs a Font Adaptive Neural Network to edit a single character. However, this character-level modification fails to replace a word with length changes, limiting performance in practical applications. SRNet (Wu et al. 2019) first proposes the word-level editing method by dividing the network into three sub-networks: background inpainting, text conversion and fusion. The divide-and-conquer strategy allows each module to handle only a relatively simple task. Extended on SRNet, SwapText (Yang, Huang, and Lin 2020) introduces the TPS module that isolates the spatial transformation from text styles, reducing the learning difficulty of text conversion module. (Zhao, Chen, and Huang 2021) apply scene text editing to document forgery and propose the forge-and-recapture operation to mitigate the visual artifacts. STRIVE (Subramanian et al. 2021) employs SRNet for video text editing. They only modify a selected frame as reference and add targeted photometric transformations to the reference frame to maintain the consistency of all other modified frames. These approaches, however, are mainly extended from SRNet, which may fail to replace a text with complex styles and can only be trained on synthetic datasets. To deal with these issues, we propose a stroke-level modification method to generate more legible text images. Our method also supports the semi-supervised training scheme and can be trained on both labeled synthetic datasets and unpaired scene text images.

Methodology

The proposed MOSTEL aims to generate legible scene text images and maintain the background integrity as much as possible. To this end, we follow the erase-and-write paradigm and design a novel stroke-level modification network. As the overall pipeline in Fig. 2 shows, MOSTEL is composed of Background Reconstruction Module (BRM) and Text Modification Module (TMM). BRM takes the source image I_s as input and outputs the stroke guidance map $Guide_s$ and text-free background O_s. In TMM, input pairs, source image I_s and standard targeted text image I_t, are first processed by Pre-Transformation. To be specific, a TPS module is used to adjust the orientation of I_t according to the geometrical attributes of I_s. I_s is processed by Background Filtering and Style Augmentation, which are supposed to filter out redundant background textures and preserve independent and robust text style information. Then the transformed pairs are fused with reconstructed background features to generate the edited text image. Stroke-Level Modification is implemented in both BRM and TMM to ensure more reliable outputs. Details will be introduced in the following sections.

Background Reconstruction Module

Background Reconstruction Module (BRM) aims to generate text-free background images with proper textures. Inspired by SRNet (Wu et al. 2019), BRM adopts an encoder-decoder structure. The encoder consists of three down-sampling layers and four residual blocks, and the decoder consists of three up-sampling layers. To obtain a more robust feature representation, a PSP Module (Zhao et al. 2017) is applied to enhance multi-scale features of the encoder. However, addressing scene text editing as a vanilla image-to-image translation problem is suboptimal and has two shortcomings. First, while the background is expected to remain the same, directly modifying all the pixels at image level still causes subtle, imperceptible changes to the background re-

gions. Second, the invariant background distracts the model from focusing on the varying text regions and hinders the learning of editing rules.

Therefore, inspired by PERT (Wang et al. 2021), we propose a structure equipped with Stroke-Level Modification (SLM) to perform explicit guiding editing by minimally modifying the varying regions. The prediction of editing guidance maps is considered a segmentation task of stroke regions. When generating the partially reconstructed image $\hat{O}_s$, the network also predicts editing guidance map $Guide_s$. Then $\hat{O}_s$ and $Guide_s$ are fed into SLM to obtain the text-free background image O_s. This process can be formulated as:

$$O_s = Guide_s \times \hat{O}_s + (1 - Guide_s) \times I_s. \tag{1}$$

This explicit guidance brought by SLM can generate more reliable edited images. On the one hand, background regions are directly inherited from source image I_s, maximizing the consistency with invariant regions of the source image. On the other hand, the model can get rid of distinguishing the invariant and varying regions and only focus on the editing rules of text regions, thus simplifying the task and facilitating the learning. SLM is further quantitatively proved to be beneficial in ablation experiments.

The text-free background image O_s is optimized with GAN loss and $L2$ loss.

$$\begin{aligned} L_{b\_s} = \mathbb{E}[logD_b(T_b, I_s) + log(1 - D_b(O_b, I_s))] + \\ \lambda_{b1} \|T_b - O_b\|_2, \end{aligned} \tag{2}$$

where T_b and O_b are the ground truth and predicted background images. λ_{b1} is set to 10 to balance numeric values. D_b indicates the background discriminator, which follows the structure in SRNet (Wu et al. 2019). The supervision on the editing guidance $Guide_s$ adopts dice loss, which can be formulated as:

$$L_{b\_guide} = 1 - \frac{2\,|T_{guide\_s} \cap Guide_s|}{|T_{guide\_s}| + |Guide_s|}, \tag{3}$$

where T_{guide_s} and $Guide_s$ are the ground truth and predicted guidance map. $|X|$ means the sum of pixels in X.

Text Modification Module

Text Modification Module (TMM) is composed of two parts: Pre-Transformation and Modification Module. They are used to preprocess input images and mix text styles with background features to generate the edited image, respectively.

Pre-Transformation applies a specialized spatial transformation to the input pairs, source image I_s and standard targeted text image I_t. Inspired by Swaptext (Yang, Huang, and Lin 2020), a feature extractor and two FC layers are used to obtain control points of I_s, which are several anchor points located on contour of the text. According to the predicted control points, a TPS module is used to geometrically transform I_t into the same orientation as I_s. By decoupling spatial attributes from the other text styles, it reduces the style transfer difficulty. As for I_s, because the jumbled background textures only distract the model from learning text styles, the editing guidance $Guide_s$ is first introduced to remove background noises in Background Filtering. To generate more robust style features, Style Augmentation is proposed to enhance text styles by applying random rotation and flipping. Since the spatial orientation is decomposed by the TPS module, rotating and flipping operations in Style Augmentation cause no effect on the other styles (such as font, color, size). When the proposed Semi-supervised Hybrid Learning is used to train the model on real-world scene text images, Pre-Transformation also plays a crucial role in preventing the network from turning into an identity mapping one, which will be described in detail below.

Modification Module adopts the same encoder-decoder structure as BRM. In the decoder, the targeted text features are fused with the corresponding up-sampling features from BRM to generate the edited text image $\hat{O}_t$ and guidance map $Guide_t$. The gradient of connections is blocked, which is another measure to avoid the network falling into identity mapping when adopting Semi-supervised Hybrid Learning. It is worth mentioning that we only apply SLM in the inference stage here. This is because SLM aims to explicitly guide the network to concentrate on text regions and avoid distractions from invariant background. Pre-Transformation, which may also be regarded as the stroke-level modification, has filtered out background regions in advance, making SLM less important here. We further provide experiments to verify the effectiveness of such a structure.

In addition, a pre-trained text recognizer is introduced to ensure the generated text image clear and legible. The recognizer can use any advanced scene text recognition method. In our implementation, considering the trade-off between performance and speed, we adopt (Baek et al. 2019) as our recognizer, which is made up of TPS transformation, BiLSTM decoder and attention-based prediction.

The loss function on edited scene text image $\hat{O}_t$ uses both GAN loss and $L2$ loss.

$$\begin{aligned} L_{t\_t} = \mathbb{E}[logD_t(T_t, I_t) + log(1 - D_t(\hat{O}_t, I_t))] + \\ \lambda_{t1} \|T_t - \hat{O}_t\|_2, \end{aligned} \tag{4}$$

where T_t and $\hat{O}_t$ are the ground truth and predicted edited images. λ_{t1} is the balance factor and set to 10. The edited image discriminator D_t has the same structure as D_b. The recognizer uses cross-entropy loss.

$$L_{rec} = -\frac{1}{N} \sum_{i=1}^{N} log(p_i | g_i), \tag{5}$$

where p_i and g_i represent the prediction and ground truth. N indicates the maximum prediction length and is set to 25. As with $Guide_s$, the supervision on $Guide_t$ also uses dice loss.

$$L_{t\_guide} = 1 - \frac{2\,|T_{guide\_t} \cap Guide_t|}{|T_{guide\_t}| + |Guide_t|}. \tag{6}$$

To generate more realistic images, VGG-loss is adopted, which is divided into a perceptual loss (Johnson, Alahi, and Fei-Fei 2016) and a style loss (Gatys, Ecker, and Bethge 2016).

$$L_{vgg} = \lambda_{v1} L_{per} + \lambda_{v2} L_{style}, \tag{7}$$

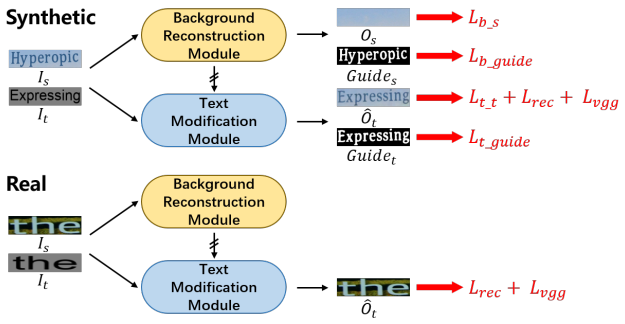


Figure 3: Illustration of Semi-supervised Hybrid Learning. Top and bottom are the training processes on labeled synthetic images and unpaired scene text images, respectively.

$$L_{per} = \mathbb{E}[\|\phi_i(T_t) - \phi_i(\hat{O}_t)\|_1], \tag{8}$$

$$L_{style} = \mathbb{E}_j[\|G_j^{\phi}(T_t) - G_j^{\phi}(\hat{O}_t)\|_1], \tag{9}$$

where ϕ_i is the activation map from *relu1_1* and *relu5_1* layer of VGG-19 model. G is the Gram matrix, and balance factors λ_{v1} and λ_{v2} are set to 1 and 500, respectively.

The whole loss function can be expressed as

$$L = \arg\min_{G} \max_{D_b, D_f} (L_{b\_s} + L_{t\_t} + \lambda_1(L_{b\_guide} + L_{t\_guide}) + \lambda_2 L_{vgg} + \lambda_3 L_{rec}), \tag{10}$$

where the weighting factors λ_1, λ_2 and λ_3 are set to 10, 1 and 0.1.

Semi-supervised Hybrid Learning

To adapt the model to real-world environments, we propose the Semi-supervised Hybrid Learning (SHL). In such a scheme, only transcript annotations are needed, which can be easily accessed in scene text recognition datasets (Karatzas et al. 2013, 2015).

As shown in Fig. 3, the network structure is the same when training on synthetic and real-world images. Due to the lack of required labels in real scene text images, we design a paradigm by converting the text to itself. To be specific, the source image I_s is first processed by BRM to generate text-free background. Standard targeted text image I_t still imitates styles from I_s and is then fused with erased background features from BRM. There are some changes in the loss functions. The supervision on guidance maps $Guide_s$, $Guide_t$ and reconstructed background image O_s are discarded. The loss function for edited image $\hat{O}_t$ is made up by only L_{rec} and L_{vgg}.

However, if the network directly uses SHL, because the supervision is the input image itself, the model may degenerate into an identity mapping network, resulting in disastrous performance in the inference stage. Different from existing methods (Wu et al. 2019; Yang, Huang, and Lin 2020), several operations are employed to prevent this collapse. As we stated before, Pre-Transformation is firstly used to perform the spatial transformation on the input images. With the background-free and augmented text styles, it is difficult for the network to find a simple correspondence between input and output images. Moreover, when fused with background features in TMM, we stop the gradient of the connections from BRM, that is, prevent the encoder and decoder in BRM from restoring the text style information. Through this specialized designed structure, we separate the background and text style apart in BRM and TMM, respectively. Extensive experiments further demonstrate the effectiveness of the network structure and the semi-supervised training scheme.

Experiment

Datasets

The datasets used for training and evaluation are introduced as follows. To our knowledge, there are no public evaluation datasets for scene text editing. Therefore, we release two new datasets, Tamper-Syn2k and Tamper-Scene, for a fair comparison between STE methods.

Synthetic Data. We generate 150k labeled images for the supervised training for MOSTEL and 2k paired images to compose Tamper-Syn2k for evaluation. The paired images are rendered with different texts and the same styles, such as font, size, color, spatial transformation and background image. In our implementation, a total of 300 fonts and 12,000 background images are used with random rotation, curve and perspective transformation.

Real Data. We use MLT-2017[1] to train MOSTEL on real-world scene text images, including 34,625 images. Only transcript annotations are required to indicate the targeted text and calculate recognizer loss. For evaluation, the proposed Tamper-Scene is a combination of several scene text datasets, including ICDAR 2013 (Karatzas et al. 2013), SVT (Wang, Babenko, and Belongie 2011), SVTP (Phan et al. 2013), IIIT (Mishra, Alahari, and Jawahar 2012), MLT-2019[2], and COCO-Text (Veit et al. 2016). By filtering the severely distorted and unrecognizable images, we select a total of 7,725 images to compose Tamper-Scene.

Implementation Details

Style Augmentation in Pre-Transformation includes random rotation with an angle from $-15°$ to $15°$ and random flipping with a probability of 0.5 during the training stage. Input images are resized to 256×64. We adopt Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$, and learning rate is set to 5×10^{-5}. We totally train 300k iterations with a batch size of 16, consisting of 14 labeled synthetic image pairs and 2 unannotated real scene text images. MOSTEL is implemented in PyTorch and trained on 1 NVIDIA 2080Ti GPU.

Evaluation Metrics

To comprehensively evaluate the edited images of our method, for paired synthetic images, we adopt the following commonly used metrics: 1). MSE, the $L2$ distances; 2). PSNR, the ratio of peak signal to noise; 3). SSIM, the mean

[1]https://rrc.cvc.uab.es/?ch=8

[2]https://rrc.cvc.uab.es/?ch=15

Figure 4: Two visual examples of guidance maps. From left to right, from top to bottom are I_s, $Guide_s$, O_s, result of SRNet, I_t, $Guide_t$, O_t, result of Swaptext.

Figure 5: Left and right are the results of adopting SLM in TMM during training and inference stage.

structural similarity; 4) FID (Heusel et al. 2017), the distances between features extracted by InceptionV3. A higher PSNR, SSIM and lower MSE, FID indicate better performance. For real scene text images, we adopt the recognition accuracy using an official text recognition algorithm (Baek et al. 2019) with its corresponding pre-trained model[3].

Ablation Study

In this section, we conduct an ablation study on Tamper-Syn2k and Tamper-Scene to show the effectiveness of our proposed Pre-Transformation, Stroke-Level Modifications and Semi-supervised Hybrid Learning. We also provide some qualitative examples in Fig. 6.

Stroke-Level Modifications: After removing SLM, there is no explicit editing guidance. Implicitly guiding the model to learn the reconstruction rules of both background and text regions undoubtedly increases the learning difficulty. Especially at the edge of text regions, the network is ambiguous in preserving the textures or applying the editing rules, resulting in poor representation of the generated text. The visualizations in Fig. 5 and Fig. 6 show that previous methods without SLM fail to generate clear and legible characters. The examples in Fig. 5 indicate precise instructions of guidance maps. The results in Tab. 1 also quantitatively verify the effectiveness of the proposed SLM.

Furthermore, we conduct an experiment on whether to apply SLM in the training process. As shown in Tab. 2, when we only adopt the results of SLM in BRM to training, the model has the best performance. We summarize the reasons as follows. First, SLM aims to explicitly guide the network to focus on editing rules of text regions and avoid the distractions from invariant background regions. In TMM, background regions are filtered out in advance in Pre-Transformation, so SLM is not particularly crucial here. Second, adopting SLM in TMM discards the background regions of the feature-level fused images $\hat{O}_t$ and directly inherits from O_s. Supervision on O_t is actually only the supervision on text regions. Such an image-level combination ignores the integrity between background and text, resulting in poor smoothness at the edge of text. According to Fig. 5, compared with the image-level fusion, the feature-level fusion can integrate the background and targeted text in a more harmonious way, resulting in a smooth and seamless presentation of characters. Therefore, we only adopt SLM in TMM at the inference stage.

[3]https://github.com/clovaai/deep-text-recognition-benchmark

Method	Tamper-Syn2k				Tamper-Scene
	MSE↓	PSNR↑	SSIM↑	FID↓	SeqAcc↑
pix2pix†	0.0732	12.01	0.3492	164.24	18.382
SRNet†	0.0193	18.66	0.6098	41.26	32.298
SwapText†	0.0174	19.43	0.6524	35.62	60.634
w/o SLM	0.0135	20.30	0.6917	33.59	70.395
w/o rec	0.0126	20.50	0.7072	36.61	68.375
w/o BF	0.0134	20.46	0.7061	34.68	72.362
w/o SA	0.0125	20.71	0.7157	36.50	26.408
w/ gradient	0.0131	20.59	0.7105	38.37	25.670
MOSTEL	**0.0123**	**20.81**	**0.7209**	**29.48**	**76.790**

Table 1: Quantitative results on Tamper-Syn2k and Tamper-Scene. †means the methods that we reproduce. w/o SLM, rec, BF, SA denote without Stroke-Level Modification, recognizer, Background Filtering, Style Augmentation. w/ gradient means allowing the gradient propagation of the connections between decoders in BRM and TMM.

BRM	TMM	Tamper-Syn2k				Tamper-Scene
		MSE↓	PSNR↑	SSIM↑	FID↓	SeqAcc↑
-	-	0.0124	20.77	0.7185	29.86	74.990
✓	-	**0.0123**	**20.81**	**0.7209**	**29.48**	**76.790**
-	✓	0.0130	20.66	0.7119	32.41	66.848
✓	✓	0.0126	20.72	0.7160	30.66	70.032

Table 2: Ablation experiments of whether adding the results of SLM into training process.

Semi-supervised Hybrid Learning. Several experiments about the number of real data in a batch have been conducted in Tab. 3, where the total batch size is set to 16. When the network is trained only with synthetic data, all loss functions in Fig. 3 work properly and each module can get the best optimization. The model performs best on the Tamper-Syn2k. When trained with real data, the network sacrifices part of its supervision, such as the reconstructed background and editing guidance maps, in exchange for adaptation to real-world scene text image distributions. The ratio of real data to synthetic data is a parameter to trade off the network's ability to edit scene text images and the ability to achieve the expected function of each module. With the increase of real data, due to the domain gap between synthetic and real data, the performance on Tamper-Syn2k is getting worse. Results in Tab. 3 show that when the number is set to 2, MOSTEL has the best performance on balancing these two capabilities. Therefore, we use batch size 16 with 14 synthetic data and 2 real data in all the following experiments.

As we stated in Methodology, such a scheme by converting the text in source image to itself faces the problem that

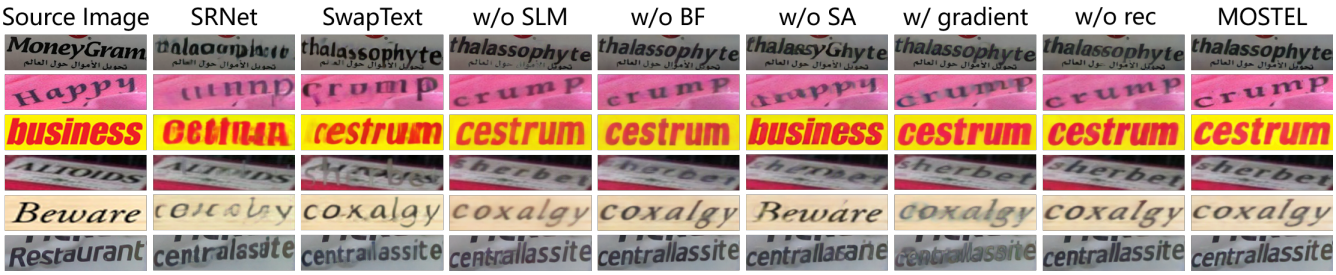


Figure 6: Some qualitative examples of previous methods and different configurations of the MOSTEL. w/o SLM, BF, SA, rec indicate MOSTEL without Stroke-Level Modification, Background Filtering, Style Augmentation, recognizer. w/ gradient means allowing the gradient propagation of the connections between decoders in BRM and TMM.

Real Data Number	Tamper-Syn2k				Tamper-Scene
	MSE↓	PSNR↑	SSIM↑	FID↓	SeqAcc↑
0	**0.0122**	**20.87**	**0.7221**	**28.73**	68.906
1	0.0123	20.83	0.7210	28.94	75.353
2	0.0123	20.81	0.7209	29.48	**76.790**
4	0.0125	20.76	0.7182	30.20	68.220
2 (SRNet†)	0.2045	9.71	0.3642	151.41	7.361
2 (Swaptext†)	0.2010	10.61	0.3976	108.98	9.682

Table 3: Performance about the number of real-world scene text images when using the proposed Semi-supervised Hybrid Learning. The total batch size is set to 16. †means that we reproduce the methods using the same configuration.

the network may degenerate into an identity mapping one. Three countermeasures (Background Filtering, Style Augmentation, blocking gradient) are adopted to prevent it. As Tab. 1 shows, these measures are all effective for improving the recognition results on Tamper-Scene. However, it is noteworthy that without Style Augmentation and blocking gradient, the model performs well on Tamper-Syn2k, but causes disastrous results on Tamper-Scene. This is because the metrics on Tamper-Syn2k (MSE, PSNR, SSIM) are only used to statistically judge the similarity between two images without considering the visual quality. In other words, when the model severely degenerates into an identity mapping network, the text of an input image is almost unchanged except for slight modifications in some pixel values. Although the text content in the predicted image and the ground-truth are different, they may have similar colors, positions and pixel numbers, resulting in a high statistical metric. While the recognition results focus on reflecting the legibility of the generated text images. Therefore, we believe that these evaluation metrics working together can better match the visual quality of edited images.

Furthermore, we reproduce previous methods (Wu et al. 2019; Yang, Huang, and Lin 2020) and train them using the same semi-supervised configuration as MOSTEL. To be specific, for real data, we remove supervisions on all the unavailable intermediate results and use the input image to supervise the final output. The batch size is also set to 16 with 14 synthetic data and 2 real data. The last two rows in Tab. 3 show that they are incapable of being semi-supervised trained with both synthetic and real data, which further verifies the robustness of our structure.

Recognizer. As indicated in Tab. 1, directly adopting a recognizer in training is proven to be beneficial to all the metrics. The visualization examples in Fig. 6 also show that the model is able to generate clearer and more legible with the supervision of recognition results.

Comparisons with Previous Methods

To our knowledge, the code and evaluation datasets of SRNet (Wu et al. 2019) and Swaptext (Yang, Huang, and Lin 2020) are not publicly available so far. Therefore, we reproduce these methods and train them with the same training datasets and iterations as MOSTEL. Since they do not support training with real data, which is proved in Ablation Study, we only use synthetic data to train them. As shown in Tab. 1, our MOSTEL outperforms them by at least 0.0051, 1.38, 0.0685 and 6.14 in MSE, PSNR, SSIM and FID respectively on Tamper-Syn2k. On Tamper-Scene, the recognition accuracy of MOSTEL surpasses them by over 16.156%, demonstrating the superiority of our proposed method.

Conclusion

This study proposes an end-to-end trainable framework named MOSTEL for scene text editing. We attribute the limited performance to implicit editing guidance and the domain gap between synthetic training data and real scene text images. Therefore, by introducing the Stroke-Level Modification, we propose to filter out the distractions from invariant background regions and explicitly guide the model to focus on the editing rules of text regions. In addition, a Semi-supervised Hybrid Learning is proposed to enable the network to be trained with both paired synthetic images and unlabeled real-world images. Several measures, such as Background Filtering, Style Augmentation and gradient-free connections, are introduced to adapt the model to such a scheme. Extensive quantitative experiments and qualitative results verify the superiority of our method. Besides MOSTEL, we also release two evaluation datasets named Tamper-Syn2k and Tamper-Scene to facilitate a fair comparison. As for future work, we will combine our MOSTEL with a scene text detection network for an end-to-end text editing system and further improve the performance.

Acknowledgments

This work was supported in part by the National Key Research and Development Program of China under Grant 2022YFB3104700, and in part by the National Nature Science Foundation of China under Grants 62121002, U1936210, 61972105, and 62102384.

References

Azadi, S.; Fisher, M.; Kim, V. G.; Wang, Z.; Shechtman, E.; and Darrell, T. 2018. Multi-content gan for few-shot font style transfer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7564–7573.

Baek, J.; Kim, G.; Lee, J.; Park, S.; Han, D.; Yun, S.; Oh, S. J.; and Lee, H. 2019. What is wrong with scene text recognition model comparisons? dataset and model analysis. In *Proceedings of the IEEE/CVF international conference on computer vision*, 4715–4723.

Du, Y.; Chen, Z.; Jia, C.; Yin, X.; Zheng, T.; Li, C.; Du, Y.; and Jiang, Y.-G. 2022. SVTR: Scene Text Recognition with a Single Visual Model. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, 884–890.

Fragoso, V.; Gauglitz, S.; Zamora, S.; Kleban, J.; and Turk, M. 2011. TranslatAR: A mobile augmented reality translator. In *2011 IEEE workshop on applications of computer vision (WACV)*, 497–502. IEEE.

Gatys, L. A.; Ecker, A. S.; and Bethge, M. 2016. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2414–2423.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. *Advances in neural information processing systems*, 27.

Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; and Hochreiter, S. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30.

Isola, P.; Zhu, J.-Y.; Zhou, T.; and Efros, A. A. 2017. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1125–1134.

Jaderberg, M.; Simonyan, K.; Vedaldi, A.; and Zisserman, A. 2014. Synthetic data and artificial neural networks for natural scene text recognition. *arXiv preprint arXiv:1406.2227*.

Johnson, J.; Alahi, A.; and Fei-Fei, L. 2016. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, 694–711.

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 *2015 13th international conference on document analysis and recognition (ICDAR)*, 1156–1160. IEEE.

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 *2013 12th international conference on document analysis and recognition*, 1484–1493. IEEE.

Kingma, D. P.; and Welling, M. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Mishra, A.; Alahari, K.; and Jawahar, C. 2012. Scene text recognition using higher order language priors. In *BMVC-British machine vision conference*. BMVA.

Phan, T. Q.; Shivakumara, P.; Tian, S.; and Tan, C. L. 2013. Recognizing text with perspective distortion in natural scenes. In *Proceedings of the IEEE International Conference on Computer Vision*, 569–576.

Qin, X.; Zhou, Y.; Guo, Y.; Wu, D.; Tian, Z.; Jiang, N.; Wang, H.; and Wang, W. 2021. Mask is all you need: Rethinking mask r-cnn for dense and arbitrary-shaped scene text detection. In *Proceedings of the 29th ACM International Conference on Multimedia*, 414–423.

Qiu, J.; Chai, Y.; Tian, Z.; Du, X.; and Guizani, M. 2019. Automatic concept extraction based on semantic graphs from big data in smart city. *IEEE Transactions on Computational Social Systems*, 7(1): 225–233.

Qu, Y.; Xie, H.; Fang, S.; Wang, Y.; and Zhang, Y. 2022. ADNet: Rethinking the Shrunk Polygon-Based Approach in Scene Text Detection. *IEEE Transactions on Multimedia*, 1–14.

Roy, P.; Bhattacharya, S.; Ghosh, S.; and Pal, U. 2020. STEFANN: scene text editor using font adaptive neural network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13228–13237.

Sheng, F.; Chen, Z.; and Xu, B. 2019. NRTR: A no-recurrence sequence-to-sequence model for scene text recognition. In *International Conference on Document Analysis and Recognition (ICDAR)*, 781–786.

Subramanian, J.; Chordia, V.; Bart, E.; Fang, S.; Guan, K.; Bala, R.; et al. 2021. STRIVE: Scene Text Replacement In Videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 14549–14558.

Sun, D.; Ren, T.; Li, C.; Su, H.; and Zhu, J. 2017. Learning to write stylized chinese characters by reading a handful of examples. *arXiv preprint arXiv:1712.06424*.

Veit, A.; Matera, T.; Neumann, L.; Matas, J.; and Belongie, S. 2016. Coco-text: Dataset and benchmark for text detection and recognition in natural images. *arXiv preprint arXiv:1601.07140*.

Wang, K.; Babenko, B.; and Belongie, S. 2011. End-to-end scene text recognition. In *2011 International conference on computer vision*, 1457–1464. IEEE.

Wang, Y.; Xie, H.; Fang, S.; Qu, Y.; and Zhang, Y. 2021. PERT: A Progressively Region-based Network for Scene Text Removal. *arXiv preprint arXiv:2106.13029*.

Wu, L.; Zhang, C.; Liu, J.; Han, J.; Liu, J.; Ding, E.; and Bai, X. 2019. Editing text in the wild. In *Proceedings of the 27th ACM international conference on multimedia*, 1500–1508.

Yang, Q.; Huang, J.; and Lin, W. 2020. Swaptext: Image based texts transfer in scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14700–14709.

Yang, S.; Liu, J.; Lian, Z.; and Guo, Z. 2017. Awesome typography: Statistics-based text effects transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7464–7473.

Yang, S.; Liu, J.; Wang, W.; and Guo, Z. 2019a. TET-GAN: Text effects transfer via stylization and destylization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 1238–1245.

Yang, S.; Wang, Z.; Wang, Z.; Xu, N.; Liu, J.; and Guo, Z. 2019b. Controllable artistic text style transfer via shape-matching gan. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4442–4451.

Zhan, F.; Lu, S.; and Xue, C. 2018. Verisimilar image synthesis for accurate detection and recognition of texts in scenes. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 249–266.

Zhao, H.; Shi, J.; Qi, X.; Wang, X.; and Jia, J. 2017. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2881–2890.

Zhao, L.; Chen, C.; and Huang, J. 2021. Deep Learning-based Forgery Attack on Document Images. *IEEE Transactions on Image Processing*, 30: 7964–7979.

Zhu, J.-Y.; Park, T.; Isola, P.; and Efros, A. A. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, 2223–2232.