Text's Armor: Optimized Local Adversarial Perturbation Against Scene Text Editing Attacks

Tao Xiang
Chongqing University
Chongqing, China
txiang@cqu.edu.cn

Hangcheng Liu
Chongqing University
Chongqing, China
hcliu@cqu.edu.cn

Shangwei Guo
Chongqing University
Chongqing, China
swguo@cqu.edu.cn

Hantao Liu
Cardiff University
Cardiff, United Kingdom
liuh35@cardiff.ac.uk

Tianwei Zhang
Nanyang Technological University
Singapore
tianwei.zhang@ntu.edu.sg

ABSTRACT

Deep neural networks (DNNs) have shown their powerful capability in scene text editing (STE). With carefully designed DNNs, one can alter texts in a source image with other ones while maintaining their realistic look. However, such editing tools provide a great convenience for criminals to falsify documents or modify texts without authorization. In this paper, we propose to actively defeat text editing attacks by designing invisible "armors" for texts in the scene. We turn the adversarial vulnerability of DNN-based STE into strength and design local perturbations (i.e., "armors") specifically for texts using an optimized normalization strategy. Such local perturbations can effectively mislead STE attacks without affecting the perceptibility of scene background. To strengthen our defense capabilities, we systemically analyze and model STE attacks and provide a precise defense method to defeat attacks on different editing stages. We conduct both subjective and objective experiments to show the superior of our optimized local adversarial perturbation against state-of-the-art STE attacks. We also evaluate the portrait and landscape transferability of our perturbations.

CCS CONCEPTS

• **Security and privacy** → **Human and societal aspects of security and privacy**; • **Computing methodologies** → *Artificial intelligence.*

KEYWORDS

Deepfake, scene text editing, adversarial perturbation

ACM Reference Format:
Tao Xiang, Hangcheng Liu, Shangwei Guo, Hantao Liu, and Tianwei Zhang. 2022. Text's Armor: Optimized Local Adversarial Perturbation Against Scene Text Editing Attacks. In *Proceedings of the 30th ACM International Conference on Multimedia (MM '22), October 10–14, 2022, Lisboa, Portugal.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3503161.3548103

MM '22, October 10–14, 2022, Lisboa, Portugal

ACM ISBN 978-1-4503-9203-7/22/10…$15.00
https://doi.org/10.1145/3503161.3548103

1 INTRODUCTION

Scene text editing (STE) [18, 22], converting texts in an image into the desired texts while preserving their original style, has been becoming a powerful tool for many applications, such as text image synthesis [31] and augmented reality translation [16]. With the rapid development of deep neural networks (DNNs) [12, 24], the performance of DNN-based STE has also been greatly improved, which can generate "fake" scene images with the replaced texts that are naturalistic and hardly distinguished from the original ones.

However, due to the powerful capability of DNN-based STE schemes, they may be maliciously used for scene text forgery, which highly increases the threat to scene text-based applications. For example, attacker can use existing DNN-based STE schemes [18, 22] to edit handwritten texts, advertising words, or slogans in posters and images. Such malicious forgery will undoubtedly have a negative impact on the reputation of the corresponding individuals or organizations. Hence, defending against malicious usage of STE is worthy of attention.

Several attempts have been proposed to resist malicious alterations [14, 15, 21, 28], most of which focus on passively detecting whether the texts in a suspicious image are altered. But passive defenses have the following limitations. First, passive detection cannot prevent the occurrence of malicious editing and requires much professional knowledge, which is not applicable to ordinary users. Second, most existing detection schemes are *ad-hoc* and designed to detect particular tamper patterns (e.g., the methods in [14, 15] are designed to detect copy-paste-based forgery) and they may be not applicable for unknown STE attacks.

In this paper, complementary to passively defending against STE attacks, we intend to defeat malicious text editing in an active and efficient way. Specifically, inspired by the adversarial vulnerability of DNNs [2, 5, 9, 11, 23], we propose a general solution to defend against DNN-based STE attacks, which utilizes adversarial perturbations to avoid the STE attacks. One straightforward way to generate adversarial perturbations is using existing methods such as [2, 9, 11]. However, challenges arise when applying existing adversarial perturbations to defend against STE attacks: 1) previous *global* schemes tend to reduce the usability of the whole image since they generate unnecessary perturbations to the background. 2) existing STE schemes are complex systems consisting of multiple stages and breaking STE attacks using adversarial perturbations has not been explored yet. To address the challenges, we aim to

answer the following two questions in this paper: 1) *how to generate optimized adversarial perturbations without affecting the usability of the image?*, and 2) *how to effectively defend against STE attacks?*

To solve the first question, we propose a method of generating optimized local adversarial perturbations specifically for texts to protect them from malicious editing. In particular, we design local adversarial perturbations as texts' armors without affecting the visual quality of the background. But the defense performance of the local armors may be unsatisfactory when using the sign function to determine the update direction because it sets all partial derivatives in the gradient to {-1, 0, 1} roughly. As result, the directions seriously deviate from the ideal ones pointed to the gradient. To obtain more efficient local armors, we propose a novel gradient normalization strategy to reduce such deviation. We set partial derivatives in the dominant dimensions as -1 or 1 to ensure the progress of the generation of armors. For the remaining dimensions, we scale the corresponding partial derivatives adaptively, which does not cause deviation. We also theoretically prove that our normalization strategy can make the update direction closer to the ideal one.

To defeat the STE attacks more effectively, we analyze existing STE attacks and model the text editing process with four necessary stages. For each stage, we propose a fine-grained defense goal (e.g. distorting the altered texts) to precisely defend against the attacks on different editing stages. We conduct comprehensive experiments (both subjective and objective) to evaluate our active defense on two state-of-the-art STE attacks. Subjective evaluations show that our defense can actively resist STE attacks with a very high probability (nearly 100%) even without original images as reference. Objective evaluations on each stage show that our perturbations can precisely defeat STE attacks and preserve the visual quality of source images. We also evaluate the capabilities of our defense by analyzing both the portrait and landscape transferability of our perturbations.

The main contributions in this paper include:

- We propose an active defense to resist STE attacks by misleading the STE processing using adversarial perturbations.
- We propose a novel gradient normalization strategy to generate optimized local perturbations.
- We systemically analyze and model the STE attacks and propose fine-grained defense goals for each editing stage.
- We conduct both subjective and objective experiments to evaluate our defense against state-of-the-art STE attacks.

2 RELATED WORK

2.1 Scene Text Editing

According to different workflows, we classify current STE schemes into two categories: style transfer-based STE [22, 30, 32, 34] and parameterization-based STE [18].

2.1.1 Style Transfer-based STE. This approach considers the text editing as a style transfer problem in the spatial domain. Wu et al. [22] first proposed a classic STE scheme consisting of three modules: text conversion, background inpainting, and fusion. Following the workflow, subsequent researchers proposed several improvements to improve the three modules. For example, Zhao et al. [34] proposed a two-step fusion and use an adversarial loss to make the altered results more realistic. Yang et al. [30] proposed to use geometric control points of characters to move text locations. Yu et al. [32] generate a three-channel mask to capture the location and shape of text body, outline, and shadow.

2.1.2 Parameterization-based STE. Recently, Shimoda et al. [18] proposed to convert a source image into a parametric representation and then reconstruct the image from the representation. The representation is a complete description of the image, so the attacker can obtain a new representation for the fake image by manipulating the text in the original representation. After that, the fake image can be drawn easily based on the new representation.

2.2 Defenses against STE

2.2.1 Passive Defense. Existing defense schemes usually use a passive way to detect whether a given image is fake. For example, Nandanwar et al. [14] identified a forgery image by analyzing the shapes of the Fourier spectrum. They also proposed to train a DNN-based classifier to detect forgery images [15]. Yan et al. [28] detected alterations in document images by analyzing the texture and reflectance characteristics. Wang et al. [21] applied Faster R-CNN to capture inconsistent features between the repaired and authentic regions. Although these passive defenses work properly for particular tamper patterns, they may fail in detecting unknown patterns and cannot prevent the generation of forgery images. Therefore, existing passive defenses are inefficient to identify forgery images and defeat STE attacks in advance.

2.2.2 Active Defense. To the best of our knowledge, there are few works to resist DNN-based STE attacks actively. The active defense has made some achievements in other fields. For example, in [1, 19, 26, 27, 33], adversarial perturbations have successfully defeated malicious scene text recognition (STR). Similarly, to prevent the abuse of deepfake [10, 13], many works [6, 17, 29] proposed to add adversarial perturbations to face images for maximizing the distortion in synthetic face images. These successful cases inspire us to turn the adversarial vulnerability of DNN-based STE attacks into strength to resist unauthorized text editing. However, due to the complexity of STE attacks and the characteristics of texts, one can hardly apply existing adversarial perturbation generation methods into defending against STE attacks. In the following sections, we systemically analyze STE attacks and propose optimized local adversarial perturbations specifically for texts to effectively defeat STE attacks at different editing stages.

3 PROBLEM STATEMENT

3.1 Scene Text Editing Modeling

Let i^s, w^s be a source image and a source text in the image. A scene text editing scheme aims to generate a forgery image o^f, in which w^s has been altered as another target text w^t while keeping its realistic look. The public can hardly identity whether o^f is fake only through visual observation.

Existing STE schemes are complex systems consisting of multiple modules. We fully analyze the state-of-the-art STE schemes [18, 22, 30, 34] and summarize the following four necessary stages of STE:

(1) Text location. This stage is responsible for determining the location of texts in the source image manually [22] or through a well-trained locator [18].

(2) Background inpainting. This stage removes the source texts from the source image and also fills the corresponding holes. The repaired pixels should be consistent with the background.
(3) Text style parsing. This stage extracts the styles of source texts, such as fonts, borders, and shadows. It ensures the consistency of the text styles before and after the alteration.
(4) Fake image generation. This stage produces the fake image by combining the altered texts and the repaired background.

3.2 Threat Model

We mainly study the active defense in a white-box setting, where the defender tries to prevent an attacker from altering the target texts using the targeted STE scheme without authorization. As we show in the transferability analysis of Section 5.4, our defense can be applied to black-box scenarios, in which we do not need any prior knowledge about the STE scheme.

Let P be a defense method. D is a visual distance metric between images and d is a discriminator for identifying whether the input is real. $Pr(d(\hat{o^f}) = 0)$ is the probability that the discriminator d regards $\hat{o^f}$ as a forgery image. To defeat STE attacks and preserve the functionality of the source image, we can formalize an active defense based on two goals: 1) the defense should be effective to defend against STE attacks, and 2) the defense can not reduce the functionality of the source image.

Definition 3.1. (ϵ,δ)-Active Defense) Let i^s be a source image, where an attacker wants to alter the texts in i^s using an STE scheme f. P is a defense scheme and $\hat{i^s}$ be the protected image ($\hat{i^s} = P(i^s)$). $\hat{o^f} = f(\hat{i^s})$. P is a (ϵ,δ)-Active Defense if for $\forall\ i$, $D(i^s, \hat{i^s}) < \epsilon$ and $Pr(d(\hat{o^f}) = 0) > \delta$.

In this paper, we are inspired by the vulnerability of DNNs and propose to defend against STE attacks by carefully designing optimized local adversarial perturbations. To the best of our knowledge, this is the first work to defeat STE attacks by turning the vulnerability of STE attacks into strength. We have to emphasize that existing adversarial perturbation generation methods are not proper for resisting malicious STE schemes due to the particularity and complexity of STE tasks. To achieve a better active defense, we would propose a novel adversarial perturbation generation method and precisely defeat all editing stages of STE attacks as described below.

4 METHODOLOGY

4.1 Overview

Insights. We turn the adversarial vulnerability of DNNs [20] into strength and design a novel method for generating optimized local adversarial perturbations to resist malicious STE. Our design strategies are twofold according to the two goals in Definition 3.1. *First*, we propose to design local perturbations specifically for the text areas of a source image. Most of the existing adversarial perturbations are global, which will reduce the visual quality of the background. Instead, we intend to provide local protection for the texts and reduce the negative impact of the perturbations on the background. *Second*, we optimize the iteration process of perturbation generation to obtain more precise update directions. Existing gradient-based adversarial attacks [2–4, 9, 11, 25] usually use the sign function to normalize partial derivatives of all dimensions into {-1, 0, 1}, which makes the update direction seriously deviate from the real one pointed to the gradient. Our design strategy is to optimize the sign function and find a more accurate direction for each update during the perturbation generation.

Pipeline. Besides generating the text's armor, another question we want to explore is how to maximize the defense performance. An STE scheme is a complex system and the vulnerability of a system is determined by its most vulnerable part (Cannikin Law). To this end, we propose fine-grained goals to explore the vulnerability of each stage of the STE attacks (i.e. precise defense). We illustrate the pipeline of our active defense in Fig. 1. Given a source image and a specific defense goal, we generate the optimized armors for resisting the corresponding editing stage using our gradient normalization strategy. With the protection of these carefully designed armors, the forgery images would be distorted and can be identified as false easily even without any professional knowledge.

4.2 Optimized Local Adversarial Perturbations

The key component in our defense pipeline is the generation of the local adversarial perturbations, which is formalized as:

$$\begin{aligned} &\max_{\eta} \mathcal{L}_\star(i^s + \eta, f) \\ &s.t.\ \|\eta\|_\infty < \epsilon \end{aligned}. \tag{1}$$

$\mathcal{L}_\star$ is a cost function designed for defending against the editing stage $\star$ ($\star$ denotes one of the four necessary editing stages or their combination). We will discuss how to design the $\mathcal{L}_\star$ in Section 4.3. η is the local armors we want to generate. We can solve the above optimization problem based on existing gradient-based methods [3, 4, 9, 11, 25]. Without loss of generality, we take BIM [9] as an instance that generates adversarial perturbations iteratively using

$$\hat{i^s}_{t+1} = \text{Clip}_\epsilon\left(\hat{i^s}_t + \alpha \times \text{sign}\left(\nabla_{\hat{i^s}_t}\mathcal{L}_\star\left(\hat{i^s}_t, f\right)\right)\right) \text{ and } \hat{i^s}_0 = i^s, \tag{2}$$

where Clip_ϵ clips the input values to the required range ($\|\eta\|_\infty < \epsilon$). However, these gradient-based methods have two problems: generating unnecessary perturbations for non-text areas and updating perturbations with inaccurate directions.

Locality. To produce perturbations only for pixels in the text areas (i.e. local perturbations), we add a mask operation before the calculation of the sign function,

$$\hat{i^s}_{t+1} = \text{Clip}_\epsilon\left(\hat{i^s}_t + \alpha \times \text{sign}\left(\text{M}\left(\nabla_{\hat{i^s}_t}\mathcal{L}_\star\left(\hat{i^s}_t, f\right)\right)\right)\right) \text{ and } \hat{i^s}_0 = i^s. \tag{3}$$

M represents the mask operation that makes the sign function ignore the partial derivatives of pixels in the non-text areas by directly setting these partial derivatives as 0.

Optimized update direction. To reduce the deviation at each iteration, we propose a new gradient normalization strategy, partial sign (PS), that divides all dimensions into *dominant* and *non-dominant* dimensions and only makes the full use of the perturbation budget in the dominant dimensions. Specifically, for a gradient g (i.e. $\nabla_{\hat{i^s}_t}\mathcal{L}_\star$), we have $g = [g_1, g_2, \cdots, g_n]$, where n is the number of dimensions. Then, we rank all partial derivatives (g_i) in descending order according to their absolute values as

$$g^r = [g_1^r, g_2^r, \cdots, g_n^r],\ \|g_i^r\| \geq \|g_j^r\|\ if\ i \leq j. \tag{4}$$

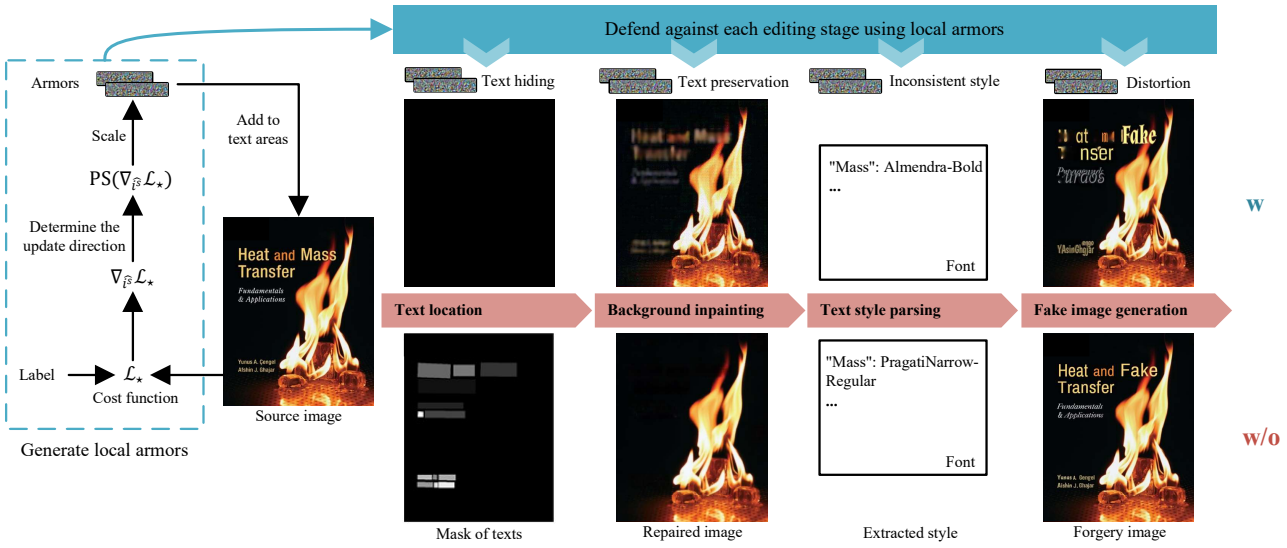


Figure 1: Our active defense pipeline against STE attacks. Given a source image and a specific defense goal represented by a cost function $\mathcal{L}_\star$, we first generate the local armors and then add them to the corresponding text areas. "w" means with defense, and "w/o" means without defense.

The top p percent of the dimensions in g^r are dominant because the gradient vector will be closer to the axes of these dimensions. The other dimensions are non-dominant. Next, we scale g by $g^s = s \times g$ where $s = \frac{1}{g^r_{\lfloor n \times p\% \rfloor} + \gamma}$ (γ is a tiny value and we set 10^{-7} in our experiments). Finally, we clip all values in g^s into the range of $[-1, 1]$ to obtain the normalized result, i.e., PS(g). With our PS, the complete generation of local perturbations can be described as

$$\hat{i^s}_{t+1} = \text{Clip}_\epsilon\left(\hat{i^s}_t + \alpha \times \text{PS}\left(\text{M}\left(\nabla_{\hat{i^s}_t}\mathcal{L}_\star\left(\hat{i^s}_t, E\right)\right)\right)\right) \text{ and } \hat{i^s}_0 = i^s. \tag{5}$$

We summarize the generation process in Algorithm 1.

We theoretically analyze the effectiveness of the proposed optimized normalization strategy. Compared with existing gradient-based adversarial perturbation generation methods, the optimized local perturbation can gradually approximate the ideal one by providing a more accurate update direction at each iteration.

Theorem 4.1. *For any update gradient $\nabla_{\hat{i^s}_t}\mathcal{L}_\star$, the PS strategy generates a normalization result of the gradient with a smaller offset than that produced by the sign function, i.e.,*

$$\cos(\nabla_{\hat{i^s}_t}\mathcal{L}_\star,\ PS(\nabla_{\hat{i^s}_t}\mathcal{L}_\star)) \geq \cos(\nabla_{\hat{i^s}_t}\mathcal{L}_\star,\ sign(\nabla_{\hat{i^s}_t}\mathcal{L}_\star)) \tag{6}$$

4.3 Defense to Each Stage

A complete STE attack consists of multiple stages and defeating any stage would lead to the success of the defense. Thus, we provide fine-grained defense goals for defeating each editing stage precisely as the following:

- Texts hiding. It is proposed to hinder the text detection. If one text cannot be detected by the locator, all subsequent editing operations cannot be applied to this text.
- Texts preservation. It breaks the inpainting by preserving the target texts. The preserved texts will overlap with altered texts, which can be easily identified as false.

Algorithm 1: Protecting texts with local armors.

Input : Source image i^s, cost function $\mathcal{L}_\star$, perturbation budget ϵ, step size α, T, p, γ.
Output: Protected source image $\hat{i^s}$.

$\hat{i^s_0} \leftarrow i^s$;
for $t \leftarrow 0$ *to* $T-1$ **do**
  $g \leftarrow \nabla_{\hat{i^s}_t}\mathcal{L}_\star$;
  $g \leftarrow$Set all partial derivatives of the corresponding dimensions of all non-text area pixels in g to 0;
  $g' \leftarrow$All non-zero partial derivatives in g;
  $g^r \leftarrow$Rank values in abs(g') in descending order;
  $s \leftarrow \frac{1}{g^r_{\lfloor n \times p\% \rfloor} + \gamma}$;
  $g^s \leftarrow s \times g$;
  $g^{sc} \leftarrow \text{clip}(g^s, -1, 1)$;
  $\hat{i^s} \leftarrow \text{clip}(\hat{i^s} + \alpha \times g^{sc},\ i^s - \epsilon,\ i^s + \epsilon)$;
$\hat{i^s} \leftarrow \text{clip}(\hat{i^s}, -1, 1)$;
return $\hat{i^s}$;

- Inconsistent text style. This goal is to mislead the text style parsing, and chaotic text styles are also suspicious.
- Distortion in fake images. It is proposed to break the last stage. Unnatural distortions reduce the realness.

The design of cost functions is related to both the defense goal and the type of the specific sub-model used in each editing stage. For sub-models of classification tasks (e.g. the model for predicting fonts), the cost function can be formalized as

$$\mathcal{L}_\star = -\max(z^\star_y - \max(z^\star_i : i \neq y), \tau), \tag{7}$$

where $z^\star_y$ is the logit value of the corresponding label y, and τ controls the strength of defense (we set it as 0 in the experiments). For sub-models of regression tasks (e.g. the model for generating

fake images), we establish the cost function in the form of L_1-norm as

$$\mathcal{L}_\star = \|y^\star - o^\star\|_1, \tag{8}$$

where $o^\star$ is the output of the editing stage $\star$ and $y^\star$ is the corresponding label. In addition, the goal of text preservation is targeted and the inpainting model is regression, where the corresponding cost function is

$$\mathcal{L}_b = -\|i^s - o^b\|_1, \tag{9}$$

where o^b is the repaired image.

Combine all defense goals. Besides considering these defense goals individually, we can also combine them for realizing multiple defense goals. In this case, the cost function is

$$\mathcal{L}_{com} = \sum_\star \lambda_\star \mathcal{L}_\star, \tag{10}$$

where $\lambda_\star$ is the weight hyperparameters.

5 EXPERIMENTS

5.1 Configurations

Datasets. Following the default settings in state-of-the-art STE schemes, we use two datasets in our experiments: the synthetic dataset [22] (short as SYN) and the book cover dataset [7] (short as BOOK). In SYN, the source images are the extracted text areas and the text location detection stage is not applicable in this case. Besides, there is also other auxiliary ground truth. BOOK consists of whole source images, each of which contains multiple texts.

Attack configurations. We choose two state-of-the-art STE attacks in our experiments, Derendering [18] and SRNet [22]. Specifically, we use SRNet to edit texts in the source images of SYN. *SRNet does not consider the stage of text location that is completed manually.* The stage of text style parsing in SRNet consists of two sub-models for generating text skeletons and foreground texts. The sub-models of each editing stage in SRNet are summarized in Table 1.

We use Derendering to vectorize the source images in Book. Since there are no ground truth of altered images in Book for comparison, we do not modify the parametric representation for editing, but directly reconstruct the original images from them. The defense performance can be quantified by the difference between the original and reconstructed images. The text style parsing in Derendering includes five submodules for predicting the fonts, the visibility of shadow (Shadow-V) and border (Border-V), and the effect of shadow (Shadow-E) and border (Border-E), respectively. The sub-models of each editing stage in Derendering are shown in Table 2. For the stage of fake image generation, we implement it using APIs provided in Sika [1].

Defense configurations. We follow [2, 11] to set $T = 40$ and $\epsilon = 0.3$. To balance the deviation and the update progress, we intuitively set $p = 50$, i.e., half of the dimensions will be truncated as -1 or 1. For different editing stages, we choose appropriate cost functions for achieving the fine-grained defense goals, which are summarized in Table 1 and 2. Note that, due to the lack of ground truth in BOOK, we use the outputs of each editing stage in Derendering w.r.t. the original source image as the pseudo labels to generate armors.

Without loss of generality, we choose two popular gradient-based methods, BIM [9] and PGD [11], as baselines. Based on the two methods, we apply our design strategies and obtain PS-M-BIM and PS-M-PGD by adding the mask operation and optimizing the sign function. We also conduct ablation experiments by only considering one strategy. So we obtain PS-BIM (or PS-PGD) and M-BIM (or M-PGD). For SYN and SRNet, the local strategy is natural because the source images only contain text areas. Please note that the proposed mask operation and PS can be also applied to other gradient-based methods.

[1] https://skia.org

Table 1: Configurations for SRNet. "R" means regression

Stage	Inpainting	Parsing		Generation
		Skeleton	Foreground	
Model type	R	R	R	R
$\mathcal{L}_\star$	Eq. (9)	Eq. (8)	Eq. (8)	Eq. (8)

Table 2: Configurations for Derendering. We denote "R" and "C" as the regression and classification, respectively

Stage	Location	Inpainting	Parsing				
			Font	Border-V	Border-E	Shadow-V	Shadow-E
Model type	C	R	C	C	C	C	R
$\mathcal{L}_\star$	Eq. (7)	Eq. (9)	Eq. (7)	Eq. (7)	Eq. (7)	Eq. (7)	Eq. (8)

Metrics. In the subjective evaluations, we use the proportion of images identified as false to assess the defense performance. In objective evaluations, we use PSNR and SSIM to calculate the distance between the source images and their armed version. Besides, we use the recall rate (RT) to measure the performance of hiding texts, which is calculated by $\text{RT} = \frac{N_d}{N_w}$. N_w is the number of the text areas in a source image (we denote the number of areas detected in the source image as N_w) and N_d is the number of detected text areas in the protected source image. To assess the performance of disturbing the text style, we use the accuracy metric (Acc) for the classification sub-models and L_1 for the regression sub-models.

5.2 Subjective Evaluations

We conduct subjective experiments to evaluate the effectiveness of our defense, i.e., determine whether the texts in the scene images can be maliciously edited after arming the texts with our "armors". We recruit 10 observers in the subjective experiments. The observers are first asked to browse 100 source images in each dataset to establish a general impression of the source images. Then we choose another 300 source images and generate the "armors" for the texts in these images. We implement STE attacks to alter the protected texts using Derendering and SRNet. Given the altered images, the observers are asked to observe the style and typesetting of the texts in the images and identify whether our defense was successful. We also provide some subjective guidelines (illustrated in Fig. 2) to assist the assessment of the observers. For example, partial erasure of texts or unnatural text overlap indicates the STE attacks fail to alter texts while maintaining the realistic look. However, inconspicuous changes liking the shadows in the bottom right image of Fig. 2 may be thought as inherent, which means a successful alteration. Note that, the observers are told not judge whether the images are true or false by the content of texts because the content of texts can be modified at will.

We evaluate the effectiveness of both PS-M-BIM and PS-M-PGD and the defense performance of resisting all possible stages. The

Figure 2: STE attack results on four protected source images. Upper left: texts are partially erased and bad typography. Upper right: original text "MANARA" have not been erased cleanly in the inpainting. Bottom left: messy text style and strange typography. Bottom right: inconspicuous shadows.

Table 3: Subjective evaluation of our defense on Derendering

Method	Stage								w/o
	Location	Inpainting	Parsing					Combine	
			Font	Border-V	Border-E	Shadow-V	Shadow-E		
PS-M-BIM	100.00%	67.58%	77.17%	21.01%	89.27%	27.85%	93.15%	95.89%	10.50%
PS-M-PGD	100.00%	64.84%	78.54%	22.83%	89.95%	31.51%	93.61%	97.21%	

average evaluation results are shown in Table 3 and 4, in which the values are the proportion of images identified as false. It is clear that our armors can prevent malicious text editing with a very high probability, especially when we defend against the location in Derendering (100%) and the foreground text generation in SRNet (>95%). Defending against the location in Derendering has two consequences: 1) all texts in an image cannot be detected and the following editing step (e.g., vectorization) cannot be executed (we set the reconstructed image to black in this case); 2) only part of pixels of text areas are recognized and erased, while the remaining are kept in the reconstructed image (see the upper left image of Fig. 2). As shown Fig. 3(a), defending against the foreground text generation in SRNet makes the altered texts distorted. All the above phenomena indicates the failure of the STE attacks.

In addition, we test another cost function for the inpainting in SRNet because we find that $\mathcal{L}_b = -\|i^s - o^b\|_1$ conflicts with the cost function used to distort the generated fake images (denoted as $\mathcal{L}_g$). Specifically, $\mathcal{L}_b = -\|i^s - o^b\|_1$ can maintain the background in the repaired image, but $\mathcal{L}_g$ try to distort the background. Because of this conflict, the performance of the armors generated based on the combined cost function is not good. After using $\mathcal{L}_b = \|t^b - o^b\|_1$, the corresponding scores have been significantly improved.

Table 4: Subjective evaluation of our defense on SRNet

Method	$\mathcal{L}_b$	Stage					w/o
		Inpainting	Parsing		Generation	Combine	
			Skeleton	Foreground			
PS-M-BIM	$-\|i^s - o^b\|_1$	81.67%	80.83%	97.67%	45.33%	50.67%	18.33%
	$\|t^b - o^b\|_1$	88.67%				85.67%	
PS-M-PGD	$-\|i^s - o^b\|_1$	79.00%	80.33%	95.83%	59.50%	66.33%	
	$\|t^b - o^b\|_1$	82.17%				87.33%	

(a) Distorted text (b) Bold text

Figure 3: Visualization samples of the altered images before (top) and after (bottom) using our defense. (a) Defending against the generation of foreground texts makes the altered texts unrecognizable. (b) Defending against the fake image generation leads to the thicker texts.

5.3 Objective Evaluations

Defeating text location. We use PSNR an SSIM to measure the distance between the original and protected source images for assessing the imperceptibility of armors. Besides, we use RT to quantify the performance of hiding text areas (RT without defense is 1 because of the pseudo labels). The experimental results are shown in Table 5. From Table 5, one can observe that both the mask operation and the PS strategy improve the imperceptibility of the armors. Besides, the PS strategy further reduces RT compared the sign function, which indicates that reducing the deviation in each update does enhance the defense effect of hiding texts.

Defeating background inpainting. We show the imperceptibility of our armors and the L_1 distance between the source and repaired images (i.e. the similarity) in Table 6. From Table 6, the PS-based methods achieve approximate even better text preservation performance with fewer perturbations compared with the sign-based methods. However, we find that the defense against the inpainting of Derendering reduces the similarity, which means failed defense from the perspective of the objective metric L_1. However, the defense is successful from the perspective of visual observation shown as Fig. 4(a). The reason for this gap is the perturbations added to the background occupying most of the area seriously increase the L_1 distance. *Therefore, we must emphasize that due to the gap between objective image quality metrics and subjective feelings, the results of all objective evaluations on the similarity even the imperceptibility only partially reflect the defense effect, not accurately.*

Defeating text style parsing. In order to avoid premature termination of iteration of armors generation caused by the wrong location, we modify the locator's output with the corresponding pseudo label of location during the generation process. We do not execute the replacement in the inference phase, and we average the results over all detected real text areas. The experimental results are shown in Table 7 and 8, where the similarity in Table 8 means the L_1 distance between the output of the corresponding substage and its label. From the two tables, we observe that the PS-based methods still performs better than the sign-based methods in most

Table 5: Defenses against text localization

Method	RT ↓	Imperceptibility	
		PSNR ↑	SSIM ↑
BIM	0.017	40.093	0.969
PS-BIM	0.009	42.610	0.981
M-BIM	0.004	42.589	0.996
PS-M-BIM	0.000	43.143	0.997
PGD	0.003	29.706	0.870
PS-PGD	0.000	29.975	0.880
M-PGD	0.001	35.566	0.987
PS-M-PGD	0.001	35.651	0.987

Table 6: Defenses against background inpainting

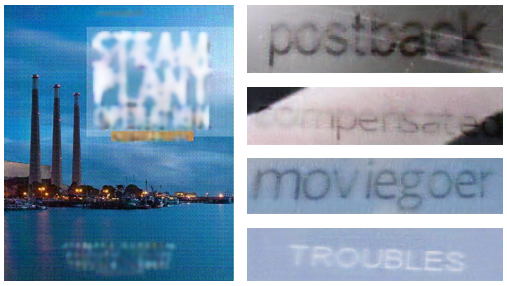
Dataset	Method	Similarity		Imperceptibility	
		w/o	w ↓	PSNR ↑	SSIM ↑
SYN	M-BIM	0.093	0.063	25.879	0.808
	PS-M-BIM		0.061	26.315	0.826
	M-PGD		0.067	23.416	0.628
	PS-M-PGD		0.068	23.629	0.631
BOOK	BIM	0.026	0.031	26.331	0.758
	PS-BIM		0.029	27.127	0.782
	PGD		0.038	24.937	0.724
	PS-PGD		0.039	24.943	0.735

Table 7: Defenses against the text style parsing in Derendering

Method	Acc ↓				L_1 ↑	
	Font	Border-V	Border-E	Shadow-V	Shadow-E	
					Blur	Offset
BIM	0.007	0.267	0.002	0.119	10.424	28.147
PS-BIM	0.008	0.047	0.002	0.026	9.611	29.187
M-BIM	0.006	0.081	0.008	0.008	4.189	10.476
PS-M-BIM	0.003	0.080	0.006	0.008	4.147	11.960
PGD	0.002	0.011	0.005	0.012	13.104	33.431
PS-PGD	0.006	0.001	0.004	0.004	11.145	31.249
M-PGD	0.008	0.007	0.025	0.005	4.331	11.654
PS-M-PGD	0.005	0.004	0.025	0.007	4.547	12.047

Table 8: Defenses against the text style parsing in the SRNet

Stage	Method	Similarity		Imperceptibility	
		w/o	w ↑	PSNR ↑	SSIM ↑
Skeleton	M-BIM	0.163	0.205	29.758	0.868
	PS-M-BIM		0.203	30.468	0.889
	M-PGD		0.256	25.890	0.705
	PS-M-PGD		0.258	26.121	0.713
Foreground	M-BIM	0.049	0.101	27.121	0.795
	PS-M-BIM		0.101	27.843	0.817
	M-PGD		0.106	25.088	0.689
	PS-M-PGD		0.105	25.320	0.696



(a) Derendering (b) SRNet

Figure 4: Preserved texts in the repaired images.

Table 9: Defenses against the fake image generation in SRNet

Method	Similarity ↑	Imperceptibility	
		PSNR ↑	SSIM ↑
M-BIM	0.154	27.056	0.816
PS-M-BIM	0.151	27.947	0.838
M-PGD	0.164	24.478	0.697
PS-M-PGD	0.164	24.735	0.703

Table 10: Defenses against SRNet and Derendering considering all defense goals

Dataset	Method	Similarity		Imperceptibility	
		w/o	w ↑	PSNR ↑	SSIM ↑
SYN	M-BIM	0.066	0.104	28.427	0.848
	PS-M-BIM		0.104	28.894	0.852
	M-PGD		0.121	25.930	0.721
	PS-M-PGD		0.121	25.947	0.734
BOOK	BIM	0.024	0.072	28.431	0.815
	PS-BIM		0.071	30.251	0.860
	M-BIM		0.036	32.913	0.981
	PS-M-BIM		0.036	33.444	0.982
	PGD		0.084	24.855	0.741
	PS-PGD		0.082	25.348	0.759
	M-PGD		0.038	28.869	0.965
	PS-M-PGD		0.037	28.895	0.965

cases. Besides, the defenses do successfully mislead the outputs of the all sub-models used in different parsing stages. Note that, not all misleading in the style parsing stage can resist forgery with a high probability as we have shown in Table 3. Thus, we must carefully choose the target to defend against.

Defeating Fake image generation. We test the defense against the fake image generation in SRNet and list the evaluation results in Table 9, where the similarity is the distance between the generated and desired fake image. Compared with the similarity without defense in Table 10, the increases of the L_1 distances in 9 indicate that the armors successfully distort the generated fake image. However, from Fig 3(b), we can see that the distortion is mainly reflected in bold text and blurred background, which does not destroy the authenticity of the image a lot, liking that in Fig 3(a). This result confirms the gap between the objective metrics and subjective feelings again. All above experimental results show that the defenses deployed for resisting the underlying editing stages (e.g. text location, background inpainting, and text style parsing) are more effective than directly defending against the final stage.

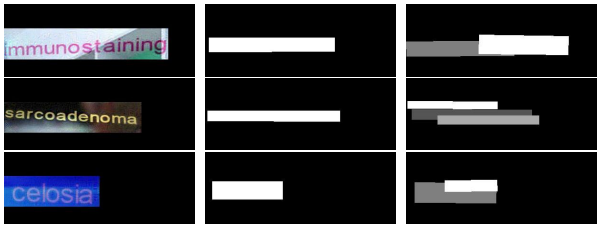
Combine all stages. We finally generate armors by considering all defense goals and exhibit the results in Table 10. From Table 10 and the previous evaluation results (e.g. Table 9 and 8), PGD-based methods can better maintain the visual quality of the source images because of the initial random noises. Thus, we suggest to use BIM-based defense to resist STE attacks. In addition, defending against multiple stages is not necessarily better than defending against only one stage. For example, the combined defense against Derendering is less effective than the defense against the location of Derendering (see Table 3). This is because the defenses against different stages interfere with each other and this interference may be negative, such as we have shown in Table 4.

5.4 Transferability

In this section, we discuss the portrait and landscape transferability of the proposed armors. The portrait transferability refers to the ability that an armor generated for hindering one stage can also work for other stages in the same STE scheme. And the landscape

Table 11: Portrait transferability of different stages

Stage	RT	Acc		L_1
	Location	Font	Border-E	Shadow-E
Location	0.002	0.463	0.043	2.414
Font	0.812	0.003	0.023	2.436
Border-E	0.708	0.080	0.003	3.113
Shadow-E	0.559	0.180	0.004	7.994



(a) $\hat{i}^s$ (b) w/o (c) w

Figure 5: Landscape transferability. (a) Protected source images. (b) Text masks of the original source images. (c) Text masks of protected source images

transferability refers to the ability that an armor generated for an STE scheme can also defeat another one.

Portrait transferability. We evaluate the portrait transferability between the four key stages in Derendering: text location, font parsing, Border-E, and Shadow-E. The evaluation results are shown in Table 11. We replace the outputs of the locator with the corresponding pseudo labels to count Acc and L_1 to accurately evaluate the portrait transferability. The L_1 values of Shadow-E is the sum of the L_1 values of blur and offset. Table 11 confirms that the armors generated for one sub-model can also influence other sub-models in Derendering even these sub-models are parallel (e.g. the parsers of font, Border-E, and Shadow-E). Therefore, even if we only know partial details of STE attacks (i.e. semi-white-box), we can defend against the attack.

Landscape transferability. In this evaluation, we assume that we know the details of SRNet but know nothing about the targeted STE attack (Derendering). We generate the proposed armors thought SRNet and add them to the source images in SYN. Then, we feed these armed source images to Derendering and observe the editing results. To meet the requirement of Derendering about the input size, we place the armed image in a larger black image before feeding it to Derendering. As illustrated in Fig. 5, we observe that these armors can hinder the text location in Derendering. The probability of hindering the location is about 23% in our evaluation. Meanwhile, these armors can also mislead the prediction of fonts as shown in Table 12. All these evidences show that the proposed armors have a strong landscape transferability. This indicates our defense can be applicable to black-box scenarios, where the defender has little prior knowledge about the STE attack.

5.5 Evaluation on ICDAR

Besides SYN and BOOK, we also test our active defense on a real-world dataset ICDAR [8]. In this evaluation, we apply PS-BIM and PS-M-BIM to defend against each stage of Derendering, whose results are shown in Table 13. From the experimental results, we can clearly observe that our method also works for ICDAR.

We also test our method against Derendering on ICDAR using different p and show the results in Table 14. We observe that the setting of p can slightly affect the effectiveness of our defense and our method can always defeat the text location and font recognition process regardless of the value of p. The two sets of experiments can further confirm the practicability of our method.

Table 12: Armors generated against SRNet can reduce the accuracy of the prediction of fonts in Derendering

Inpainting	Skeleton	Foreground	Generation	Combine
0.450	0.630	0.576	0.590	0.606

Table 13: Defenses against Derendering on ICDAR

Method	Location	Inpainting		Parsing					
	RT	L_1		Acc				L_1	
		w/o	w	Font	Border-V	Border-E	Shadow-V	Shadow-E (Blur)	Shadow-E (Offset)
PS-BIM	0.007	0.031	0.022	0.005	0.000	0.000	0.005	16.969	37.848
PS-M-BIM	0.003	-	-	0.004	0.000	0.015	0.000	6.713	19.953

Table 14: Evaluation of different p

Stage	$p = 20$	$p = 30$	$p = 40$	$p = 50$	$p = 60$	$p = 70$
Location (RT)	0.004	0.003	0.003	0.003	0.004	0.005
Font (Acc)	0.010	0.005	0.004	0.004	0.008	0.019

6 CONCLUSION

In this paper, we present a precise active defense against DNN-based STE attacks. We take the adversarial vulnerability of DNNs and generate local perturbations using our gradient normalization strategy to protect texts. Besides, we also systematically analyze and model the STE attacks for providing precise defenses against individual editing stages. Both subjective and objective assessments have demonstrated the superior of our defense on two state-of-the-art STE attacks, even in black-box scenarios.

In the future, we will mainly consider enhancing the practicability and reducing the complexity of our active defense. So we will explore how to improve the transferability, including the portrait and landscape transferabilities, of text armors and how to produce a universal armor for multiple images.

ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China under Grants U20A20176, U21A20463, 62072062, and 62102052, the Natural Science Foundation of Chongqing, China, under Grant cstc2022ycjh-bgzxm0031 and cstc2021jcyj-msxmX0744, Singapore Ministry of Education (MOE) AcRF Tier 2 MOE-T2EP20121-0006 and AcRF Tier 1 RS02/19.

REFERENCES

[1] Lu Chen and Wei Xu. 2020. Attacking optical character recognition (ocr) systems with adversarial watermarks. *arXiv preprint arXiv:2002.03095* (2020).

[2] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. 2018. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 9185–9193.

[3] Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. 2019. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 4312–4321.

[4] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR)*.

[5] Shangwei Guo, Siyuan Geng, Tao Xiang, Hangcheng Liu, and Ruitao Hou. 2021. ELAA: An efficient local adversarial attack using model interpreters. *International Journal of Intelligent Systems* (2021).

[6] Hao Huang, Yongtao Wang, Zhaoyu Chen, Yuheng Li, Zhi Tang, Wei Chu, Jingdong Chen, Weisi Lin, and Kai-Kuang Ma. 2021. CMUA-Watermark: A Cross-Model Universal Adversarial Watermark for Combating Deepfakes. *arXiv preprint arXiv:2105.10872* (2021).

[7] Brian Kenji Iwana, Syed Tahseen Raza Rizvi, Sheraz Ahmed, Andreas Dengel, and Seiichi Uchida. 2016. Judging a book by its cover. *arXiv preprint arXiv:1610.09204* (2016).

[8] Dimosthenis Karatzas, Faisal Shafait, Seiichi Uchida, Masakazu Iwamura, Lluis Gomez i Bigorda, Sergi Robles Mestre, Joan Mas, David Fernandez Mota, Jon Almazàn Almazàn, and Lluís Pere de las Heras. 2013. ICDAR 2013 Robust Reading Competition. In *International Conference on Document Analysis and Recognition (ICDAR)*. 1484–1493.

[9] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. 2017. Adversarial machine learning at scale. In *International Conference on Learning Representations (ICLR)*.

[10] Yuezun Li, Xin Yang, Pu Sun, Honggang Qi, and Siwei Lyu. 2020. Celeb-df: A large-scale challenging dataset for deepfake forensics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 3207–3216.

[11] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations (ICLR)*.

[12] Xin Man, Deqiang Ouyang, Xiangpeng Li, Jingkuan Song, and Jie Shao. 2022. Scenario-Aware Recurrent Transformer for Goal-Directed Video Captioning. *ACM Transactions on Multimedia Computing, Communications, and Applications* 18, 4 (2022), 1–17.

[13] Yisroel Mirsky and Wenke Lee. 2021. The creation and detection of deepfakes: A survey. *Comput. Surveys* 54, 1 (2021), 1–41.

[14] Lokesh Nandanwar, Palaiahnakote Shivakumara, Prabir Mondal, Karpuravalli Srinivas Raghunandan, Umapada Pal, Tong Lu, and Daniel Lopresti. 2021. Forged text detection in video, scene, and document images. *IET Image Processing* 14, 17 (2021), 4744–4755.

[15] Lokesh Nandanwar, Palaiahnakote Shivakumara, Umapada Pal, Tong Lu, Daniel Lopresti, Bhagesh Seraogi, and Bidyut B Chaudhuri. 2020. A new method for detecting altered text in document images. In *International Conference on Pattern Recognition and Artificial Intelligence (ICPRAI)*. 93–108.

[16] Marc Petter, Victor Fragoso, Matthew Turk, and Charles Baur. 2011. Automatic text detection for mobile augmented reality translation. In *IEEE International Conference on Computer Vision Workshops (ICCVW)*. 48–55.

[17] Nataniel Ruiz, Sarah Adel Bargal, and Stan Sclaroff. 2020. Disrupting deepfakes: Adversarial attacks against conditional image translation networks and facial manipulation systems. In *European Conference on Computer Vision (ECCV)*. 236–251.

[18] Wataru Shimoda, Daichi Haraguchi, Seiichi Uchida, and Kota Yamaguchi. 2021. De-rendering Stylized Texts. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 1076–1085.

[19] Congzheng Song and Vitaly Shmatikov. 2018. Fooling OCR systems with adversarial text images. *arXiv preprint arXiv:1802.05385* (2018).

[20] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. In *International Conference on Learning Representations (ICLR)*.

[21] Xinyi Wang, He Wang, and Shaozhang Niu. 2019. An image forensic method for AI inpainting using faster R-CNN. In *International Conference on Artificial Intelligence and Security (ICAIS)*. 476–487.

[22] Liang Wu, Chengquan Zhang, Jiaming Liu, Junyu Han, Jingtuo Liu, Errui Ding, and Xiang Bai. 2019. Editing Text in the Wild. In *Proceedings of the ACM International Conference on Multimedia (MM)*. 1500–1508.

[23] Tao Xiang, Hangcheng Liu, Shangwei Guo, Yan Gan, and Xiaofeng Liao. 2022. EGM: An Efficient Generative Model for Unrestricted Adversarial Examples. *ACM Transactions on Sensor Networks* (2022).

[24] Tao Xiang, Ying Yang, Shangwei Guo, Hangcheng Liu, and Hantao Liu. 2021. PRNet: a progressive recovery network for revealing perceptually encrypted images. In *Proceedings of the ACM International Conference on Multimedia (MM)*. 3537–3545.

[25] Cihang Xie, Zhishuai Zhang, Yuyin Zhou, Song Bai, Jianyu Wang, Zhou Ren, and Alan L Yuille. 2019. Improving transferability of adversarial examples with input diversity. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2730–2739.

[26] Xing Xu, Jiefu Chen, Jinhui Xiao, Lianli Gao, Fumin Shen, and Heng Tao Shen. 2020. What machines see is not what they get: Fooling scene text recognition models with adversarial text images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 12304–12314.

[27] Xing Xu, Jiefu Chen, Jinhui Xiao, Zheng Wang, Yang Yang, and Heng Tao Shen. 2020. Learning optimization-based adversarial perturbations for attacking sequential recognition models. In *Proceedings of the ACM International Conference on Multimedia (MM)*. 2802–2822.

[28] Jiabin Yan and Changsheng Chen. 2021. Cross-Domain Recaptured Document Detection with Texture and Reflectance Characteristics. In *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. 1708–1715.

[29] Chaofei Yang, Leah Ding, Yiran Chen, and Hai Li. 2021. Defending against gan-based deepfake attacks via transformation-aware adversarial faces. In *International Joint Conference on Neural Networks (IJCNN)*. 1–8.

[30] Qiangpeng Yang, Jun Huang, and Wei Lin. 2020. Swaptext: Image based texts transfer in scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 14700–14709.

[31] Shuai Yang, Jiaying Liu, Wenhan Yang, and Zongming Guo. 2018. Context-aware unsupervised text stylization. In *Proceedings of the ACM international conference on Multimedia (MM)*. 1688–1696.

[32] Boxi Yu, Yong Xu, Yan Huang, Shuai Yang, and Jiaying Liu. 2021. Mask-guided GAN for robust text editing in the scene. *Neurocomputing* 441 (2021), 192–201.

[33] Jiaming Zhang, Jitao Sang, Kaiyuan Xu, Shangxi Wu, Xian Zhao, Yanfeng Sun, Yongli Hu, and Jian Yu. 2021. Robust CAPTCHAs Towards Malicious OCR. *IEEE Transactions on Multimedia* 23 (2021), 2575–2587.

[34] Lin Zhao, Changsheng Chen, and Jiwu Huang. 2021. Deep Learning-Based Forgery Attack on Document Images. *IEEE Transactions on Image Processing* 30 (2021), 7964–7979.