Latest updates: https://dl.acm.org/doi/10.1145/3581783.3612076

RESEARCH-ARTICLE

ProTegO: Protect Text Content against OCR Extraction Attack

YANRU HE, University of Science and Technology of China, Hefei, Anhui, China

KEJIANG CHEN, University of Science and Technology of China, Hefei, Anhui, China

GUOQIANG CHEN, University of Science and Technology of China, Hefei, Anhui, China

ZEHUA MA, University of Science and Technology of China, Hefei, Anhui, China

KUI ZHANG, University of Science and Technology of China, Hefei, Anhui, China

JIE ZHANG, Nanyang Technological University, Singapore City, Singapore

View all

Open Access Support provided by:

Nanyang Technological University

University of Science and Technology of China

National University of Singapore

Aerospace Information Research Institute

ProTegO: Protect Text Content against OCR Extraction Attack

Yanru He
heyanru@mail.ustc.edu.cn
University of Science and Technology
of China

Kejiang Chen*
chenkj@ustc.edu.cn
University of Science and Technology
of China

Guoqiang Chen
ch3nye@mail.ustc.edu.cn
University of Science and Technology
of China

Zehua Ma
mzh045@ustc.edu.cn
University of Science and Technology
of China

Kui Zhang
zk19@mail.ustc.edu.cn
University of Science and Technology
of China

Jie Zhang
jie_zhang@ntu.edu.sg
Nanyang Technological University

Huanyu Bian
bianhy@aircas.ac.cn
Aerospace Information Research
Institute, Chinese Academy of
Sciences

Han Fang
fanghan@nus.edu.sg
National University of Singapore

Weiming Zhang
Nenghai Yu
zhangwm@ustc.edu.cn
ynh@ustc.edu.cn
University of Science and Technology
of China

ABSTRACT

Online documents greatly improve the efficiency of information interaction but also cause potential security hazards, such as the ability to copy and reuse text content without authorization readily. To address copyright concerns, recent works have proposed converting reproducible text content into non-reproducible formats, making digital text content observable but not duplicable. However, as the Optical Character Recognition (OCR) technology develops, adversaries can still take screenshots of the target text region and use OCR to extract the text content. None of the existing methods can be well adapted to this kind of OCR extraction attack. In this paper, we propose "ProTegO", a novel text content protection method against the OCR extraction attack, which generates adversarial underpaintings that do not affect human reading but can interfere with OCR after taking screenshots. Specifically, we design a text-style universal adversarial underpaintings generation framework, which can mislead both text recognition models and commercial OCR services. For invisibility, we take full advantage of the fusion property of human eyes and create complementary underpaintings to display alternatively on the screen. Experimental results demonstrate that ProTegO is a one-size-fits-all method that can ensure good visual quality while simultaneously achieving a high protection success rate on text recognition models with different architectures, outperforming the state-of-the-art methods. Furthermore, we validate the feasibility of ProTegO on a wide range of popular commercial OCR services, including Microsoft, Tencent,

Alibaba, Huawei, Baidu, Apple, and Xiaomi. Codes will be available at https://github.com/Ruby-He/ProTegO.

CCS CONCEPTS

• **Security and privacy** → *Digital rights management.*

KEYWORDS

Optical Character Recognition (OCR), text protection, adversarial examples

1 INTRODUCTION

The widespread usage of smart devices, including laptops, tablets, and mobile phones, has made it easier to browse digital text content online. However, the convenience of digital access also raises copyright concerns such as duplication, manipulation, and redistribution of copyrighted documents [33]. Existing protection methods [4, 23] normally employ access restrictions to prevent unauthorized copying or use Unicode encoding technology to modify the content to make the copied text significantly different from the original. But adversaries can easily circumvent these restrictions with OCR technology. As shown in the top half of Figure 1, the attacker can take screenshots of the target text area and utilize OCR to extract text content, which is defined as the OCR extraction attack here. Therefore, it is imperative to develop an effective protection method that is adaptable to this new attack scenario on text content.

OCR technology greatly benefits from the rapid evolution of deep neural networks (DNNs), and most commercial services apply DNNs-based text recognition models for better performance. Despite DNNs having superiority in various computer vision tasks,
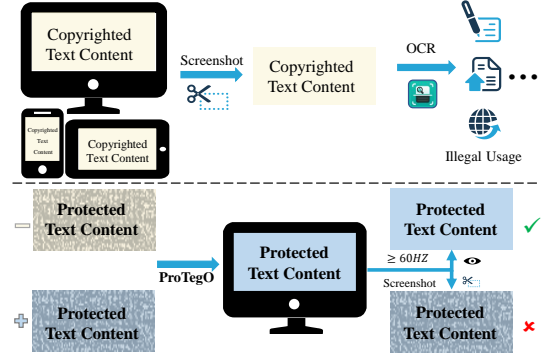
*Corresponding author.

they are susceptible to adversarial attacks. From the protection standpoint on text content, we can turn weaknesses into strengths and provide an initiative defense. Prior works [22, 31, 37, 40, 41] have extensively demonstrated the vulnerability of DNNs-based text recognition models to various types of adversarial perturbations, which are under constraints of the L_p norm. However, these methods are primarily designed for general scene text recognition (STR) tasks and cannot be directly applied to document text recognition (DocTR) tasks due to the following nontrivial challenges.

First, STR tasks typically involve text images with complex backgrounds, such as billboards and posters with rich textures or natural images. While text images in DocTR tasks tend to have simple backgrounds, e.g., e-books and scheme documents, which include pure colors or similar pattern underpaintings. The limited magnitude of modification makes adding imperceptible perturbation harder, so the contradiction between visual quality and protection strength in DocTR tasks is more prominent. Second, when in regard to blackbox settings, there also exist a significant number of commercial OCR services, such as the text recognition service offered by the application "WeChat", that do not return any intermediate probability information except for the final text content. Third, all existing methods are designed to generate adversarial perturbations for specific text content. From a practical perspective, they are timeconsuming and not suitable for large-scale documents efficiently.

To address the above limitations, we adopt the idea of transferbased methods to solve the rigorous black-box scenario, which does not require any internal knowledge of the target model. Specifically, adversarial perturbations are first generated against a surrogate model, and these perturbations are expected to work for the other models as well. We may ask: "*can we leverage the unique characteristics of text recognition models to make adversarial perturbations more transferable?*" Now we are facing this opportunity already, inspired by the fact perturbations style similar to the text appearance, defined as "text-style", can mislead text region positioning of DNN-based models, we propose ProTegO, a universal transferbased black-box method to generate adversarial underpaintings with the text-style guided. To mitigate the visual impact of adversarial underpaintings, two complementary adversarial frames are generated and alternately rendered in a high-frequency mode. In this way, the human can see the benign-looking fusion text image of complementary frames due to the flicker fusion effect of the human vision system (HVS), while the screenshots operation will only get one of the two adversarial frames, which can fool the text recognition models. To enhance the robustness of ProTegO in real-world scenarios, we investigate the pipeline of commercial OCR services and design an enhancement layer for simulating operations that potentially be involved in the OCR extraction attack.

We summarize the main contributions as follows:
1) Universal transfer-based black-box method. We propose a textstyle guided-based universal adversarial underpaintings generation method, ProTegO, which is independent of text content. Unlike state-of-the-art methods, ProTegO relies only on a local surrogate model and requires no feedback information from the target models and commercial services.
2) Good visual quality and high robustness. We explore the flicker fusion property of HVS and employ a two-frame decompositionbased visual compensation strategy for better visual quality. We

Figure 1: The schematic diagram of the OCR extraction attack (top) and the proposed method "ProTegO" (bottom).

also design a differentiable enhancement layer by mimicking common operations that occur in the OCR extraction attack, which makes our method more robust in black-box scenarios.
3) Practical protection with better performance. We thoroughly evaluate ProTegO on different models and the widespread commercial OCR services, including text recognition capabilities built into smart devices and applications, as well as the text recognition APIs of cloud platforms. Extensive experiments demonstrate that our method has superior protection performance and practicability.
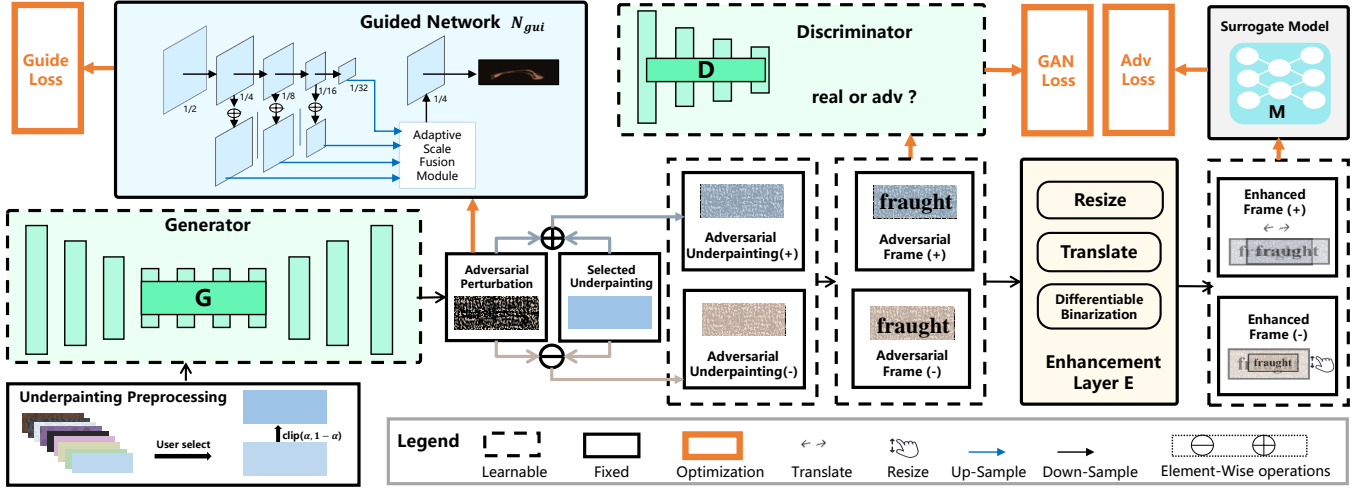
2 RELATED WORK

2.1 Optical Character Recognition (OCR)

The OCR pipeline mainly consists of text detection and text recognition. The detection models take the text images as input and locate the regions containing text. Then the recognition models output the corresponding characters from the regions. Currently, DNNbased recognition models are segmentation-free and can recognize entire character sequences in variable-sized text image input. These models are always treated as the sequential labeling task[13], first using DNN to extract features from text images, and then sequence recognition technology is used to convert the features into corresponding characters. The Connectionist Temporal Classification (CTC)[29] and the attention[30] mechanisms are two promising solutions, so recognition models are categorized into CTC-based and attention-based models. More details for readers can be found in the surveys[8, 21]. In this paper, we focus on the CTC mechanism and generalize our method to attention-based models.

2.2 Flicker Fusion Effect of Human Vision

Human eyes perceive temporal changes with light intensity in a low-pass manner. When time-variant fluctuations in light intensity exceed the lowest frequency, termed the critical flicker frequency (CFF for short hereafter), human eyes perceive only the average luminance instead of the flicker, and we call this phenomenon as *flicker fusion effect* [10, 34]. We summarize some important characteristics of the flicker fusion as follows: 1) Human eyes are more sensitive to changes in luminance than chromatic. Flickers will appear when the illuminance change of two frames is over the CFF threshold [26]. 2) Humans cannot perceive high-frequency flickers, but the screenshot operation (same as camera exposure) is enabled.

Figure 2: The overview of ProTegO. It consists of four main parts: preprocessing of underpainting, adversarial underpainting generation, robustness enhancement, and visual compensation. The whole pipeline can be trained end-to-end.

3) Thanks to the progress of screen display technology, currently used screen devices can generally support a refresh rate of 60 Hz, which is enough to exceed the CFF threshold and can effectively avoid visible flickers [24]. Inspired by the above properties, we propose a two-frame decomposition-based visual compensation strategy to add more perturbations into the single frame. Ideally, as long as the two complementary frames are shown alternately at a high frequency ($\geq$ 60 Hz), humans will perceive the fusion frame with a good visual quality like the original frame.

2.3 Adversarial Examples

Early pioneering works on adversarial examples focused on the non-sequential vision task represented by image classification [6, 12, 32, 36]. Recently, a few studies[37, 38, 41] have started to pay attention to the scene text recognition (STR) task, but the document text recognition (DocTR) task has rarely been explored, which is a more challenging problem. Since scene text images generally have complex backgrounds, they provide more relaxed conditions for adversarial attacks. But document text images leave limited operation space due to their simple backgrounds, which hinders the good performance of the method implementations. Chen *et al.* [7] first attempted a new watermark-style attack method, FAWA, which hid adversarial perturbations into watermarks. However, there are still several obvious shortcomings. First, they generate specific perturbations for each text image, which is difficult to apply to large-scale documents effectively. Second, watermarking-style perturbations are still noticeable, and to maintain good visual quality, it is hard to add more perturbations. Third, FAWA achieved remarkable white-box attack results but performed poorly in the black-box attack, such as unknown models and powerful commercial services. Xu *et al.* [39] recently presented a black-box method that was heavily based on knowledge of prediction/confidence scores and could not satisfy the rigorous black-box assumption that provides only output sequence labels. Prior works [19, 28, 35] of image classification have shown that the adversarial examples generated for one model may also be misclassified by another model. Such property is referred

to as transferability, which can be leveraged to perform black-box attacks. In this paper, we focus on the more challenging DocTR task in the rigorous black-box setting and propose a transfer-based universal adversarial underpainting generation method to achieve better performance and higher efficiency simultaneously.

3 THE PROPOSED METHOD

3.1 Threat Model

In this section, our purpose is to protect text content against the OCR extraction attack, that is, to mislead various text recognition models and commercial OCR services not extracting text content accurately. Our method is based on untargeted and black-box settings, meaning that we make the recognition results different from the original text content without any internal knowledge of the target models. We assume that the adversary can directly access the text image of the viewed document, which can be captured by taking screenshots and then performing the OCR extraction attack. We take the screenshot as our target protection scenario because of the easiest way to obtain high-quality text images.

3.2 Overview of ProTegO

Solving the OCR extraction attack in a rigorous black-box setting is demanding, especially adding invisible perturbations in DocTR tasks is even more challenging. As the defender, we could add the appropriate underpainting, which widely appears in popular software (e.g., Microsoft Office), to documents to enhance protection performance. To this end, we propose an initiative protection method, ProTegO, based on universal adversarial underpaintings and a two-frame decomposition visual compensation strategy. The overview of ProTegO is shown in Figure 2. First, an initial underpainting δ is selected based on the user's preferences, and the underpainting will be preprocessed to meet the requirements in cases that are not suitable. Next, δ is fed to the conditional generator G to produce the adversarial perturbations ϵ. Then we add and subtract the adversarial perturbations ϵ to the underpainting δ, obtaining the complementary frames of the adversarial underpainting $\delta'_{\pm}$. Last,

we can directly apply $\delta'_\pm$ to arbitrary text content x, receiving complementary adversarial text images $x'_\pm$, each of which can fool the surrogate model. Several modules give feedback to the generator to create adversarial perturbations with better performance (i.e., higher transferability and robustness), including guide network N_{gui}, discriminator D, surrogate model $\mathcal{M}$, and enhancement layer E. The details will be elaborated in the following subsections.

3.3 Preprocessing of Underpainting

We propose a principle to select and preprocess the underpainting. According to the W3C Accessibility Guidelines (WCAG) [27], the visual presentation of text and background needs to have a contrast ratio of at least 4.5:1 to ensure readability for the naked human eyes. However, higher contrast is not always better in our method, as we also need to reserve the magnitude for adding adversarial perturbations to create complementary underpaintings. We attribute underpainting preprocessing as an optimization problem and use the luminance value Y (derived from the RGB colorspace) of the underpainting as our constrained object, which is formulated by:

$$Y = 0.222485R + 0.716905G + 0.060610B \tag{1}$$

Given the original underpainting δ, we adopt the Accessible Perceptual Contrast Algorithm (APCA) [25] to acquire the contrast ratio, which is a new way to predict contrast for text and non-text content on self-illuminated displays. Once the text color is certain, the entire optimization process can be described by:

$$\mathscr{L}_c(Y(\gamma), Y(\delta)) \geq 4.5$$
$$s.t. \quad Y(\delta) \in [\alpha, 255 - \alpha] \tag{2}$$

where α is the budget for adversarial perturbations, γ is the text color, and $\mathscr{L}_c$ represents APCA.

3.4 Adversarial Underpainting Generation

To generate adversarial underpainting, we employ the conditional generative adversarial network architecture, which is widely used in the image-to-image translation literature [15]. The generator G contains three down-sampling blocks, four residual blocks, three up-sampling blocks and ends with a function $tanh$. The generator G takes the preset underpainting δ as input and outputs universal adversarial perturbations ϵ with the magnitude bound of L_2 norm. Then, a couple of complementary adversarial underpaintings are created in terms of adversarial perturbations. Notably, the adversarial underpaintings are universal, that is, independent of text content. To optimize the generator G, there are several loss functions for collaboration.

3.4.1 Hinge Loss. To bound the magnitude of the perturbation ϵ, we introduce a $\mathcal{L}_{hinge}$ loss on the L_2 norm:

$$\mathcal{L}_{hinge} = max(\| \epsilon \|_2 - c, 0) \tag{3}$$

where c denotes a user-specified bound and is set as 0.1 by default. This soft operation can also stabilize the GAN's training, as shown in prior work [36].

3.4.2 Adversarial Loss. Unlike the image classification task, the DNNs-based text recognition model deals with the input of variable size. In this paper, we take a CTC-based text recognition

model as our surrogate model $\mathcal{M}$, so we adopt the CTC mechanism to handle this sequential labeling task. Given a raw input sequence l of x, the model outputs a sequential probability distribution $y = \{y_1, y_2, ..., y_M\}$ for each character $\{l_i\}_{i=1}^T$ in l, where $M \geq T$. Then CTC operation removes blanks and redundant duplicate characters in the sequence l. Generally, the probability of one valid alignment path π can be written as $p(\pi \mid x) = \prod_{t=1}^T y_{\pi_t}^t$, where y_{π_t} is the probability of a valid character in π. Hence, for a complete prediction, CTC-based models calculate the negative log probability on all possible valid alignment paths of a given sequence l, and this process can be formulated as the CTC loss:

$$\mathcal{L}_{ctc}(x, l) = -\log \sum_{\pi \in S(l)} p(\pi|x) \tag{4}$$

where S is the set of all possible valid alignments for sequence l.

Applying complementary adversarial underpaintings to the text content, we can get the corresponding two frames of protected text images $x'_\pm$. To ensure these two frames can fool unknown target models, we let the surrogate model $\mathcal{M}$ receive two adversarial frames $x'_\pm$ and output two prediction sequences that differ from the ground-truth sequence l of the original frame. By maximizing the CTC loss of both adversarial frames to change the valid output paths, we finally obtain the adversarial loss $\mathcal{L}_{adv}$ as follows:

$$\mathcal{L}_{adv} = max(\mathcal{L}_{ctc}(x'_-, l) + \mathcal{L}_{ctc}(x'_+, l)) \tag{5}$$

3.4.3 GAN Loss. To mitigate the impact of perturbations on visual quality, we also introduce a discriminator for feedback. For the discriminator D, we adopt three "Conv-BatchNorm-ReLu" blocks for binary classification. Two complementary frames of the adversarial text images are fed to the discriminator D, and we obtain the GAN loss for the generator as:

$$\mathcal{L}_{GAN} = \log D(x) + \log(1 - D(x'_\pm)) \tag{6}$$

3.4.4 Style-guided Loss. However, we find it difficult to train a naive generator G to transfer the perturbations to other different architecture text recognition models, such as the attention-based models, and let the perturbations further work for commercial OCR services successfully. Intuitively, if we can guide G to generate perturbations similar to text appearance, it will improve transferability across various models. After careful analysis of existing text recognition models, we find that the human visual system and DNNs-based models have different sensitivities to text. Humans prioritize foreground text information and automatically filter the background that does not affect the reading. However, the models usually distinguish text and background by setting a threshold through binarization or other processing operations. This phenomenon inspires us to take a detection model as our guided network, forcing the model to recognize the generated perturbations as text content, which we define as "text-style" perturbations.

Segmentation-based detection methods have superiority in the scene text detection field, profiting from pixel-level manipulation. Generally, for a given text image, the backbone network of the detection model will predict a probability map $P_{i,j} \in [0, 1]^n$, indicating whether a pixel belongs to the text or not. In this paper, we choose a popular segmentation-based detection model as our guided network N_{gui}, but we only use the feature-pyramid backbone with the Adaptive Scale Fusion (ASF) module to obtain the probability

Table 1: The visual quality of protected text images with different methods.

Methods	Protected	Prediction	MOS	Methods	Protected	Prediction	MOS
Original		ACM	5	FAWA[7]		Acm	2.05
Xu et.al [37]		ACm	3.45	ProTegO (S-1)		Ireronving/Irercouving	3.45
AD^{2}E [39]		_CM	2.5	ProTegO (S-4)		Laong/Laroving	4.45
SINIFGSM[19]		_cm	1.5	ProTegO (S-5a)		Pencimicilis/Pencinigili	4.45
VMIFGSM[35]		Acm	1.6	ProTegO (S-5b)		Bexitbicniliate/Besitbicilite	4

Note that, the red font indicates the error recognized characters; '_' means the corresponding character for this position is not recognized; Prediction refers to the recognition results of model "STAR-Net "; The higher MOS means the better visual quality.

map. In order to force the generated perturbation containing more text features, we apply the nonlinear activation function *tanh* to the output of the N_{gui} probability map, giving the text region more weight. We then incorporate the detection result as the guide loss into our generator's optimization, which is formulated as follows:

$$\mathcal{L}_{guide} = -\sum_{i,j \in n} \tanh(\eta \times P_{i,j}) \tag{7}$$

where η is the amplification factor and is set as 1000 by default.

In summary, the final loss of ProTegO consists of the aforementioned hinge loss, style-guide loss, GAN loss, and adversarial loss, which can be formulated by:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{hinge} + \lambda_2 \mathcal{L}_{guide} + \lambda_3 \mathcal{L}_{GAN} + \lambda_4 \mathcal{L}_{adv} \tag{8}$$

where λ_1, λ_2, λ_3, and λ_4 are set as 0.001, 2, 1, and 10 by default.

3.5 Robustness Enhancement

To improve the robustness of our method in the real-world scenario, we design the *enhancement layer E* for better performance. After carefully investigating the pipeline of the OCR extraction attack, we conclude three main distortions, i.e., translation, resize, binarization, and mix them as a combined enhancement layer, which can be incorporated into the end-to-end training of ProTegO.

The first translation distortion generally occurs in taking screenshots of the text images, and users may only get an incomplete part of the underpaintings. To ensure that the enhancement design for translation does not affect the text region, we bound the translation range to four pixels in the four directions (top, bottom, left, and right) of our dataset, and this setting can also be changed according to the size of different text images. The latter two kinds of distortion resize and binarization come from the preprocessing operations commonly used in OCR. Text recognition models with specific architectures need different input sizes, so resizing the text image to a suitable size is a necessary preprocessing step for the feature extraction network. For resize operation, the text image in our dataset is first randomly resized to $h \times w \times 3$ and then resized to $32 \times 100 \times 3$, where h and w represent the height and width of the text image, respectively, and $h \in [32, 50]$, $w \in [100, 160]$. Considering that binarization is not differentiable, we define the

differentiable approximation function $t(\cdot)$ instead to simulate the effect of binarization as follows:

$$t(x'_{i,j}) = \frac{1}{1 + e^{-k(x'_{i,j}-\tau)}} \tag{9}$$

where (i, j) indicates the coordinate point in the complementary protected images x', k denotes the amplifying factor and is set as 50 by default. τ is the optimal binarization threshold found by the OTSU algorithm of OpenCV [5].

3.6 Visual Compensation

Increasing the magnitude of the perturbations can improve protection performance but reduce visual quality. Inspired by the flicker fusion effect of the human vision system, we propose the *two-frame decomposition-based visual compensation strategy* to circumvent this self-contradictory problem. As shown in the bottom half of Figure 1, after creating the two frames of protected text images $x'_{\pm}$ by the complementary adversarial underpaintings $\delta'_{\pm}$, we can alternatively display these two adversarial frames $x'_{\pm}$ on the screen at 60 Hz. Supporting by the flicker fusion characteristic, humans can only perceive the fusion text image, which is viewed as close to the original text image. However, once an attacker takes screenshots operation, they can only obtain one of the complementary frames with the adversarial underpainting, making it impossible to accurately extract the text content by text recognition models or commercial OCR services. Therefore, we can maintain better visual quality while achieving a strong strength of protection.

4 EXPERIMENTS

4.1 Experiment Settings

4.1.1 Experiment Design. To evaluate the performance of ProTegO comprehensively, we design three sets of experiments: protection against various DNNs-based text recognition models, protection against commercial OCR services, and human perception of protected text images (visual quality). For various DNNs-based text recognition models, following [37, 39], we choose four advanced models based on multistage pipelines as our black-box models. We divide them into two categories: CTC-based models CRNN [29]

Table 2: Experimental results on DNNs-based models with different architectures.

Method / Model	FAWA [7]		Xu *et al.* [37]		AD^{2}E [39]		SINIFGSM [19]		VMIFGSM [35]		ProTegO (S-5a/5b)	
	PSR(%)	ED_{avg}	PSR(%)	ED_{avg}	PSR(%)	ED_{avg}	PSR(%)	ED_{avg}	PSR(%)	ED_{avg}	PSR(%)	ED_{avg}
CTC-based STAR-Net [20]	99.67	1.54	**100**	1	**100**	1.04	93.25	1.1	99.92	1.08	**100/100**	9.39/10.21
CRNN [29]	39.68	1.17	5.25	1.08	40.79	1.1	6.43	1.07	3.84	1.13	**95/96**	4.61/3.52
Rosetta [3]	27.56	1.26	2.75	1.18	17.31	1.04	4.56	1.08	2.34	1.18	**96/93**	3.98/3.75
Attention-based RARE [30]	30.72	1.61	2.67	1.09	21.62	1.24	3.49	1.08	1.5	1.17	**91/99**	5.64/6.26
TRBA [1]	22.58	1.48	3	1.19	18.92	1.2	4.38	1.14	1.67	1.2	**97/93**	6.67/6.41
Runtime (s)	562.97		2659.06		2466.71		3072.56		1911.35		**490.22/519.74**	**(65.05)**

Note that, "S-5a/5b" represents the underpainting *Style 5* with white font and black font respectively.

and Rosetta [3], as well as attention-based models RARE [30] and TRBA [1]. For commercial OCR services, we test seven targets comprising three types. The first type is the general OCR API offered by four commercial cloud platforms: Microsoft Azure, Huaweicloud, Aliyun, and Baidu. The second type is the built-in text recognition tool provided by smartphones "iPhone 14" and "MI 10 Pro". Finally, we choose the text recognition capability provided by the "WeChat" application as the third test type. For the visual quality evaluation, we perform a comprehensive user study that involved various experimental settings, such as underpainting style, refresh rate, distance, and perspective. We use monitor "AOC-G2770PF" as the screen to display the protected text images, which supports the Variable Refresh Rate (VRR).

4.1.2 Implementation Details. Since our method focuses mainly on the DocTR task, we follow the data setting in literature [7] and use the tool *Text Recognition Data Generator* [2] to generate text images. The training dataset comprises 180 samples, and 1200 samples are used to test the models. For commercial OCR services, we randomly selected 100 text images to form a large-size test sample, and more details can be found in Appendix A.1.

For the underpainting style, we handpick eight styles (four pure colors and four textures) and preprocess them on the basis of our principle in Section 3.3. The selection of these styles also refers to the design concept of background in popular software (e.g., Microsoft Office, Notability, etc.). For example, pale green is designed to relieve visual fatigue [17], light Khaki is suitable for night mode [14], and more styles can be found in Appendix B.1. We also can adjust the text color to match the color gradation of underpaintings. We designate *Style 5* as the default underpainting, which can be used simultaneously for both black font and white font. For the surrogate model, we use the CTC-based text recognition model STAR-Net [20] in our method, which has less inference time and superior performance. The segmentation-based text detection model DBNet++[18] is picked as the prototype of our guided network.

We compare ProTegO with three state-of-the-art methods, that is, FAWA [7], Xu *et al.* [37] and AD^2E [39]. Furthermore, we also compare two general transfer-based methods, SINIFGSM [19] and VMIFGSM [35]. For a fair comparison, we set the maximum magnitude of adversarial perturbations α with the clip norm to 40/255, and all compared methods are implemented with the default settings in their papers, except for the two transfer-based methods, where we introduced early-stop mechanisms into the iterations. We train ProTegO for 60 epochs with a batch size of 16 and use Adam for the generator and discriminator with a learning rate of 10^{-3} as

the default hyperparameters. All the experiments are implemented by PyTorch [9] and executed on a single NVIDIA RTX 3090.

4.1.3 Evaluation metric. We evaluate ProTegO in three main aspects: visual quality, protection effectiveness, and efficiency.

We utilize the mean opinion score (MOS) as our evaluation metric of visual quality, which follows the same settings as in previous works [11, 16]. Specifically, we randomly ask 20 volunteers to score the visual quality of the protected text images, which are selected from the same test dataset for baseline methods and ProTegO. The score ranges from 1 (poor quality) to 5 (excellent quality), and the score increment is 1. In addition, we conduct a more extensive user study to analyze how different experimental settings affect human perception and present the detailed results in Appendix B.

We conduct two levels of evaluation for protection effectiveness. We use the protection success rate (PSR) as the word-level metric, which is designed based on the principle that the OCR services with "NONE MANUAL PARTICIPATION". The OCR extraction attack aims to copy text losslessly, even if a single character that has changed in the word needs to be manually corrected, which is time-consuming and defeats the attacker's goal. Because each sample contains a separate word, the PSR is equivalent to the success rate of word recognition. Hence, for the target word, whether the recognition result is null (no valid word recognized) or error (different from the ground-truth word), we classified it as a successful protection against the OCR extract attack. Thus, PSR can be defined as:

$$PSR = \frac{N_e}{N_{\text{total}}} \tag{10}$$

where N_e indicates the error recognition numbers and N_{total} represents the total number of test examples. For character-level evaluation with more rigorous criteria, we adopt the edit distance (ED) to measure the difference between the prediction sequence and the ground-truth sequence. The design idea behind this metric is based on the usability of text content after recognition, and the greater the editing distance, the less informative the text content will be. The average edit distance can be defined as:

$$ED_{\text{avg}} = \frac{\sum_{i=0}^{N_{\text{total}}} Dist(l_i, p_i)}{N_e} \tag{11}$$

where $Dist$ indicates the edit distance between the recognition sequence p and ground-truth sequence l.

Efficiency is also an important consideration when evaluating the practicality of methods in large-scale document scenarios. We measure the running time taken to generate protected text images

Table 3: Experimental results on a variety of commercial OCR services.

Method \\ Commercial Services	FAWA [7]		Xu *et al.* [37]		AD^{2}E [39]		SINIFGSM [19]		VMIFGSM [35]		ProTegO (S-5a/5b)	
	PSR(%)	ED_{avg}	PSR(%)	ED_{avg}	PSR(%)	ED_{avg}	PSR(%)	ED_{avg}	PSR(%)	ED_{avg}	PSR(%)	ED_{avg}
WeChat (8.0.34)	2	2.00	0	-	0	-	0	-	0	-	**100/87**	**8.19/7.94**
iPhone 14 (iOS 16.4)	16	2.50	0	-	7	1.75	1	1.00	1	1.00	**100/100**	**7.78/7.77**
MI 10 Pro (MIUI 13)	24	1.63	0	-	16	1.00	1	1.00	0	-	**87.5/89.5**	**5.55/6.50**
Baidu OCR	24	2.46	4	1.00	6	1.17	7	1.43	1	1.00	**60.5/91.5**	**4.93/6.16**
Alibaba OCR	40	2.45	3	1.67	18	2.44	5	2.00	5	2.20	**88/91.5**	**4.13/3.88**
Huawei OCR	51	3.06	2	2.70	27	2.52	19	2.47	18	2.67	**98/100**	**5.05/6.38**
Microsoft OCR	28	1.64	12	2.33	13	2.31	9	2.22	1	2.00	**96/100**	**5.82/6.31**
Average	26.43	2.25	3	1.10	12.43	1.60	6	1.45	3.86	1.41	**90/94.21**	**5.92/6.42**

Note that, "-" means that the average edit distance is not calculated due to invalid protection, and "S-5a/5b" represents the underpainting *Style 5* with white font and black font respectively.

to evaluate efficiency. Note that our method is based on the fusion text images, and the time taken for each example actually refers to the generation time of a pair of adversarial text images.

4.2 Visual Quality Evaluation

We compare the visual quality of ProTegO with the five methods mentioned in Section 4.1.2. Specifically, we prepare a large-size text image containing 100 words for each method and ask 20 volunteers to rate the protected text images. As our method is based on the fusion of two frames, we write a script to alternate display the two frames of the protected text images at the current fresh rate of the monitor with C++ and Python. Meanwhile, we also provide raters with the original text images as a reference. To demonstrate the generalization of our method, we show the effect of different underpaintings, including two pure color styles (light and dark) and a texture style. The results are presented in Table 1, and we can find that the MOS of ProTegO is much better than that of the other five methods. From the locally enlarged image block, we can see that other methods will cause visual distortion more or less, but our method will maintain a high visual quality attribute to the two-frame decomposition-based visual compensation strategy. According to Nyquist's sampling law, if the refresh rate of the screen is twice larger than the human eyes, the human will observe the average of two adjacent frames. Since we have ensured that the two adversarial frames are completely complementary, the visual artifacts from perturbations can be eliminated. The fusion text image shown on the screen will be perceived by human eyes, which greatly improves visual quality. In addition, experiments indicate that high refresh rates have a positive influence, that is, the higher refresh rate means the better visual quality. More visual quality evaluations can be found in Appendix B.2.

4.3 Protection Effectiveness Evaluation

4.3.1 Effectiveness on DNNs-based Models. Table 2 shows the results of all methods on DNNs-based models with different architectures. With regard to word-level evaluation, ProTegO and [37, 39] reach the PSR of 100% on STAR-Net, but ProTegO performs better on the character-level metric ED_{avg} of 9.39 and 10.21. This suggests that our method is more protective and the adversary obtains less useful information from the protected text content. We can also see more intuitively from the prediction results in Table 1. The five baseline methods can only make small changes in the prediction results of the target model, for instance, recognizing the capital character

to the corresponding lowercase ("M" → "m"), but this does not hinder us from learning useful information from the text. While ProTegO enables the model to output text content significantly different from the originals ("ACM" → "Laong"), which provides more powerful protection against the OCR extraction attack.

Moreover, the PSR and the ED_{avg} results of the other four models in Table 2 show that our method has better transferability and can still achieve considerable protection performance (over 90% of PSR) on unknown black-box models, covering both CTC-based and attention-based models. Due to the design of the guided network, various text recognition models are all susceptible to text-style perturbations. In particular, the average edit distance can be up to 6.35 and 6.41 on the TRBA [1] model, which has a disparate sequence decoding mechanism from our surrogate model.

4.3.2 Effectiveness on Commercial OCR services. We test the effectiveness of ProTegO on seven commercial OCR services, and the results are given in Table 3. In order to evaluate the real OCR extraction attack scenario, we test each method with the large-size protected text image that contains 100 words. It is worth noting that our method is based on the two-frame decomposition design, so we take random manual screenshots twice for fairness and take the average of the two test results.

Overall, ProTegO achieves an average PSR of 90% and 94.21%, as well as ED_{avg} of 5.92 and 6.42, which are obtained on the underpainting of *Style 5* with different font colors. In particular, only our method can provide powerful protection against OCR services built into smart devices and applications. In other words, only ProTegO can satisfy this rigorous black-box assumption that requires no access to the targets, which is very practical since some commercial OCR services, like "WeChat", do not provide a programmable API. For example, AD^{2}E [39] failed to carry out effective protection when confronted with devices that do not return confidence scores. Among all six methods, we also find that our method performs better on the OCR API services of four famous cloud platforms. This indicates that ProTegO is more effective in leveraging the useful transferability of the text-style perturbations with minimal information from the targets. Furthermore, the ED_{avg} implies that our method provides more significant advantages in terms of protection capability, thereby making it difficult for OCR to extract useful information from the given text images.

We also evaluate the impact of the magnitude of the perturbation on the effectiveness, and the maximum magnitude tolerated by our method is 50/255, according to the MOS. To provide stronger

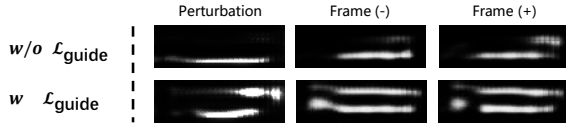**Table 4: The contribution of guided network and enhancement layer on protection effectiveness.**

Methods	STAR-Net [20] (S-5a/5b)		CRNN [29] (S-5a/5b)		Rosetta [3] (S-5a/5b)		RARE [30] (S-5a/5b)		TRBA [1] (S-5a/5b)	
	PSR(%)	ED_{avg}	PSR(%)	ED_{avg}	PSR(%)	ED_{avg}	PSR(%)	ED_{avg}	PSR(%)	ED_{avg}
base	**100/100**	8.37/8.46	93/87	4.20/3.50	84/84	3.93/3.42	84/97	4.67/6.11	90/89	5.27/5.14
w $\mathcal{L}_{guide}$	**100/100**	**9.39/10.21**	**94/88**	**4.48/3.52**	**95/89**	**3.97/3.75**	**89/98**	**5.13/6.26**	**95/93**	**6.35/6.12**
$*$ $base$	56/89	7.28/8.71	53/86	4.17/4.72	51/78	3.13/4.53	41/82	3.62/6.04	49/77	5.78/5.74
$*$ w E	**57/99**	**9.32/8.18**	**54/89**	**4.60/5.97**	**54/90**	**3.85/5.06**	**42/94**	**4.27/6.92**	**54/86**	**5.35/5.99**

Note that, "S-5a/5b" represents the underpainting *Style 5* with white font and black font respectively, "base" means the naive generator trained without $\mathcal{L}_{guide}$ and enhancement layer E, and "*" means test the protected text images with distortion operations (translation, resize and binarization) in a random way.

protection, we have to sacrifice some visual quality, and more results can be found in Appendix C.1.

4.4 Protection Efficiency Evaluation

Whether the method can be applied practically to large-scale documents is also a key point. To better evaluate the efficiency of ProTegO, we calculate the running time of generating 1200 protected text images. The last row in Table 2 shows that our method has superior efficiency compared with the other five methods. It is worth noting that the total running time of our method includes the training time, so the actual generation time of the complementary protected images is only 65.05 seconds. In other words, once the generator has finished training, we can quickly add adversarial underpaintings for documents to be protected. Hence, our method is ever so applicable in large-scale document protection scenarios, and the larger scale will highlight our strengths more.

Figure 3: The probability map with/without guide network.

4.5 Ablation Study

4.5.1 Importance of Guided Network. To justify the contribution of our guided network, we train two generators with and without the guided network, which are respectively denoted w $\mathcal{L}_{guide}$ and w/o $\mathcal{L}_{guide}$ to simplify the expression. As shown in Table 4, we find that the effectiveness of most models has improved more or less after adopting the guided network. The metric ED_{avg} better reflects this trend, especially on the model TRBA [1] with a different attention-based decoding mechanism from our surrogate model. To more intuitively demonstrate the effects of the guided network, we provide a visual illustration of the results generated by the two generators. We take the generated perturbations and two frames of protected text images as the inputs of the detection model DBNet++ [18], then get the corresponding binary probability map, where pixels with a value of 1 are considered valid text areas. Figure 3 shows that with the help of $\mathcal{L}_{guide}$, the white region of the probability map becomes larger, which means that the perturbations are more easily recognized as the text and can interfere with different text recognition models effectively.

4.5.2 Effectiveness of Enhancement Layer. To verify the robustness of the enhancement layer E in practical use, we train two versions of the generator with or without E. We randomly apply the three distortions mentioned before to the samples generated from both models and evaluate their effectiveness with the metrics PSR and ED_{avg}. Table 4 indicates that training with the enhancement layer will improve the robustness of all models when suffering from the three main distortions. We also observe that when we choose *Style 5* as our underpainting, white fonts are more vulnerable to distortion processing than black fonts. This can be interpreted as the adaptive binarization algorithm calculating an appropriate threshold based on the pixel values of the input images. Compared to black font, the contrast between white font and underpainting is relatively smaller, so the generated perturbations are closer to the underpainting, which is susceptible to the adaptive binarization algorithm. Therefore, this also provides an important reference for selecting underpainting and its contrast ratio setting with text. Moreover, all the average editing distances ED_{avg} are increased with the combined enhancement layer, indicating that effective perturbations remained after the distortions.

5 CONCLUSION

In this paper, we propose a text content protection method, ProTegO, which is the first work to protect copyrighted documents against the OCR extraction attack in a powerful, efficient, and practical manner. The key components of ProTegO are the universal adversarial underpaintings generation framework and a two-frame decomposition-based visual compensation strategy. To generate two complementary underpaintings, we design a text-style guided loss to improve the transferability and an enhancement layer to ensure robustness. In addition, the alternating display of the complementary frames achieves high invisibility by human eyes. The experimental results show that ProTegO can achieve good visual quality and effective protection simultaneously in the black-box scenario, including both different models and commercial OCR services. Moreover, we can employ ProTegO as a novel robust text-based CAPTCHAs scheme. Compared to static text CAPTCHAs, ProTegO is a dynamic solution that can perform better with the cooperation of user behavior analysis.

REFERENCES

[1] Jeonghun Baek, Geewook Kim, Junyeop Lee, Sungrae Park, Dongyoon Han, Sangdoo Yun, Seong Joon Oh, and Hwalsuk Lee. 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.

[2] Belval. 2020. TextRecognitionDataGenerator. https://github.com/Belval/TextRecognitionDataGenerator.

[3] Fedor Borisyuk, Albert Gordo, and Viswanath Sivakumar. 2018. Rosetta: Large scale system for text detection and recognition in images. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 71–79.

[4] Nicholas Boucher, Ilia Shumailov, Ross Anderson, and Nicolas Papernot. 2022. Bad characters: Imperceptible nlp attacks. In *2022 IEEE Symposium on Security and Privacy (SP)*. IEEE, 1987–2004.

[5] Gary Bradski. 2000. The openCV library. *Dr. Dobb's Journal: Software Tools for the Professional Programmer* 25, 11 (2000), 120–123.

[6] Nicholas Carlini and David Wagner. 2017. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*. Ieee, 39–57.

[7] Lu Chen, Jiao Sun, and Wei Xu. 2020. FAWA: fast adversarial watermark attack on optical character recognition (OCR) systems. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 547–563.

[8] Xiaoxue Chen, Lianwen Jin, Yuanzhi Zhu, Canjie Luo, and Tianwei Wang. 2021. Text recognition in the wild: A survey. *ACM Computing Surveys (CSUR)* 54, 2 (2021), 1–35.

[9] Ronan Collobert, Koray Kavukcuoglu, and Clément Farabet. 2011. Torch7: A matlab-like environment for machine learning. In *BigLearn, NIPS workshop*.

[10] Hao Cui, Huanyu Bian, Weiming Zhang, and Nenghai Yu. 2019. Unseencode: Invisible on-screen barcode with image-based extraction. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 1315–1323.

[11] Han Fang, Dongdong Chen, Feng Wang, Zehua Ma, Honggu Liu, Wenbo Zhou, Weiming Zhang, and Nenghai Yu. 2021. TERA: Screen-to-Camera Image Code with Transparency, Efficiency, Robustness and Adaptability. *IEEE Transactions on Multimedia* 24 (2021), 955–967.

[12] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572* (2014).

[13] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*. 369–376.

[14] Shawn Lawton Henry. 2012. Developing text customisation functionality requirements of PDF reader and other user agents. In *Computers Helping People with Special Needs: 13th International Conference, ICCHP 2012, Linz, Austria, July 11-13, 2012, Proceedings, Part I 13*. Springer, 602–609.

[15] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. 2017. Image-to-Image Translation with Conditional Adversarial Networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), 5967–5976.

[16] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. 2017. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4681–4690.

[17] Zhi Jian Li and Nuo Li. 2013. Investigation of reading background colour based on visual fatigue. In *Applied Mechanics and Materials*, Vol. 295. Trans Tech Publ, 536–538.

[18] Minghui Liao, Zhisheng Zou, Zhaoyi Wan, Cong Yao, and Xiang Bai. 2022. Real-Time Scene Text Detection with Differentiable Binarization and Adaptive Scale Fusion. *IEEE transactions on pattern analysis and machine intelligence* PP (2022).

[19] Jiadong Lin, Chuanbiao Song, Kun He, Liwei Wang, and John E Hopcroft. 2019. Nesterov Accelerated Gradient and Scale Invariance for Adversarial Attacks. In *International Conference on Learning Representations*.

[20] Wei Liu, Chaofeng Chen, Kwan-Yee K Wong, Zhizhong Su, and Junyu Han. 2016. Star-net: a spatial attention residue network for scene text recognition.. In *BMVC*, Vol. 2. 7.

[21] Xiyan Liu, Gaofeng Meng, and Chunhong Pan. 2019. Scene text detection and recognition with advances in deep learning: a survey. *International Journal on Document Analysis and Recognition (IJDAR)* 22, 2 (2019), 143–162.

[22] Yanhong Liu, Fengming Cao, and Yuqi Zhang. 2022. Generative Adversarial Examples for Sequential Text Recognition Models with Artistic Text Style.. In *ICPRAM*. 71–79.

[23] Ian Markwood, Dakun Shen, Yao Liu, and Zhuo Lu. 2017. PDF mirage: content masking attack against information-based online services. In *Proceedings of the 26th USENIX Conference on Security Symposium*. 833–847.

[24] Marino Menozzi, F Lang, U Naepflin, C Zeller, and H Krueger. 2001. CRT versus LCD: Effects of refresh rate, display technology and background luminance in visual performance. *Displays* 22, 3 (2001), 79–85.

[25] Myndex. 2022. Accessible Perceptual Contrast Algorithm. https://github.com/Myndex/apca-w3.

[26] Viet Nguyen, Yaqin Tang, Ashwin Ashok, Marco Gruteser, Kristin Dana, Wenjun Hu, Eric Wengrowski, and Narayan Mandayam. 2016. High-rate flicker-free screen-camera communication with spatially adaptive embedding. In *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*. IEEE, 1–9.

[27] Visual Contrast of Text Subgroup. 2021. Visual Contrast Whitepaper. https://www.w3.org/WAI/GL/task-forces/silver/wiki/Visual_Contrast_of_Text_Subgroup/Whitepaper.

[28] Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. 2016. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint arXiv:1605.07277* (2016).

[29] Baoguang Shi, Xiang Bai, and Cong Yao. 2017. An End-to-End Trainable Neural Network for Image-Based Sequence Recognition and Its Application to Scene Text Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39 (2017), 2298–2304.

[30] Baoguang Shi, Xinggang Wang, Pengyuan Lyu, Cong Yao, and Xiang Bai. 2016. Robust scene text recognition with automatic rectification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4168–4176.

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

[32] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. 2019. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation* 23, 5 (2019), 828–841.

[33] Milad Taleby Ahvanooey, Qianmu Li, Hiuk Jae Shim, and Yanyan Huang. 2018. A comparative analysis of information hiding techniques for copyright protection of text documents. *Security and Communication Networks* 2018 (2018).

[34] Anran Wang, Zhuoran Li, Chunyi Peng, Guobin Shen, Gan Fang, and Bing Zeng. 2015. Inframe++ achieve simultaneous screen-human viewing and hidden screen-camera communication. In *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*. 181–195.

[35] Xiaosen Wang and Kun He. 2021. Enhancing the transferability of adversarial attacks through variance tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1924–1933.

[36] Chaowei Xiao, Bo Li, Jun-Yan Zhu, Warren He, Mingyan Liu, and Dawn Song. 2018. Generating adversarial examples with adversarial networks. *arXiv preprint arXiv:1801.02610* (2018).

[37] 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/CVF Conference on Computer Vision and Pattern Recognition*. 12304–12314.

[38] Yikun Xu, Pengwen Dai, and Xiaochun Cao. 2021. Less Is Better: Fooling Scene Text Recognition with Minimal Perturbations. In *Neural Information Processing: 28th International Conference, ICONIP 2021, Sanur, Bali, Indonesia, December 8–12, 2021, Proceedings, Part VI 28*. Springer, 537–544.

[39] Yikun Xu, Pengwen Dai, Zekun Li, Hongjun Wang, and Xiaochun Cao. 2023. The Best Protection is Attack: Fooling Scene Text Recognition With Minimal Pixels. *IEEE Transactions on Information Forensics and Security* 18 (2023), 1580–1595.

[40] Mingkun Yang, Haitian Zheng, Xiang Bai, and Jiebo Luo. 2021. Cost-effective adversarial attacks against scene text recognition. In *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 2368–2374.

[41] Xiaoyong Yuan, Pan He, Xiaolin Lit, and Dapeng Wu. 2020. Adaptive adversarial attack on scene text recognition. In *IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 358–363.

A MORE IMPLEMENTATION DETAILS

A.1 Dataset Setup

The details for our dataset are shown in Table 5. We select six commonly used fonts in documents, including three ans-serif fonts (Helvetica, Arial, Verdana) and three serif fonts (Times New Roman, Courier, Georgia), as well as their bold variants, totaling up to 12 fonts. In our experiment, the synthetic images are resized to a height of 32 and a width of 100 (other heights and widths can also be used in our method). We adopt 62 characters composed of Alphabet (A~Z, a~z) and Arabic numerals (0~9) to generate 180 training samples, and each text image contains four letters or five numbers. We use the "Oxford Dictionary" as the corpus to generate 1200 test samples. The test samples consist of 100 samples for each font, and each text image contains one word. It is worth noting that both the surrogate model and the black-box test models maintain 100% accuracy in clean text images, which can eliminate the deviation caused by the model performance during evaluation.

Table 5: Details of the dataset.

Dataset	Quantity	Font	Corpus
Train	$180(15 \times 12)$	All$^\sharp$	Alphabet and Arabic numerals
Test-M	$1200(100 \times 12)$	All$^\sharp$	Oxford Dictionary
Test-C	100	Times-base	Oxford Dictionary

Note that, (a) "Test-M" means the test dataset of various models. (b) "Test-C" means the test dataset of commercial OCR services. (c) All$^\sharp$ means that include all of the twelve fonts (six basic mode and six bold variants). (d) "Times-base" represents the basic mode of the "Times New Roman" font.

A.2 Combined Training Strategy of the Enhancement layer

As described in Section 3.5, we take resize, translation, and differentiable binarization as three common distortion operations in the OCR extraction attack. In order to avoid the warm-up issue, we also add an enhancement-free layer (called Identity) to the three distortion layers above, thus forming the final combined enhancement layer E. After that, we use a mini-batch training strategy from scratch and randomly choose one layer from E in each mini-batch. Finally, our combined training strategy encourages ProTegO not only to work for a specific distortion but also to be effective for multiple different distortions at the same time. In addition, we can adjust the ratio of each enhancement layer to the target protection scenario to achieve better protection performance. Specifically, we set the ratio of each enhancement layer as

$$\sum_{i=1}^{4} q_i = 1 \quad (12)$$

where q_i represents the ratio of one of the enhancement layers. In our experiment, we heuristically set "*Identity, Resize, Translation, and Differentiable Binarization*" to be equal, that is, $q_1 = q_2 = q_3 = q_4 = 0.25$, but the ratio q_i can be changed arbitrarily based on actual protection requirements.

B MORE VISUAL ANALYSIS

For test samples, we take 100 text images generated by font "Times New Roman" to form a large-size text image and alternately display two complementary frames of the text image with different styles of adversarial underpaintings. Unless specified otherwise during the experiment, there are some default settings as follows.

- The *Style 5* is selected as our default underpainting because of its adaptability for both black and white font colors.
- The default refresh rate of the screen is 60 *Hz*.
- The default perturbation is set to $\epsilon = 40/255$.
- The default viewing distance setting is 60 cm.
- The default angle of the horizontal perspective is set to $0°$.

B.1 Different Underpainting Styles

To verify the generalization of ProTegO, we demonstrate 8 different styles of underpainting. These styles cover different modes, such as pure color, texture, dark and light modes, and each style is matched with a font color to meet contrast requirements. As shown in Table 6, we find that the MOS results of all styles are

Table 6: MOS results of different underpainting styles.

Style	Example	MOS
1		3.45
2		3.55
3		3.60
4		4.45
5a		4.45
5b		4.00
6		3.70
7		3.75
8		3.85

above 3.45, which means that our method can maintain good visual quality under various styles. Out of the test of 8 styles, *Style 4* and *Style 5a* provide relatively better MOS, as the human eyes are often sensitive to luminance. This suggests that choosing the dark style of underpainting can achieve better visual quality. We also find that *Style 3* has the best MOS among all three light styles. This is due to the fact that HVS is more sensitive toward red (R) and blue (B) channels as compared to the green (G) channel. Therefore, for better visual quality, we can set the underpainting with relatively larger values of the R and B channels, like the purple color in *Style 3*. Furthermore, we also observe that the MOS of the texture style is generally better than that of the pure color style. Compared to texture, it is easier for human eyes to perceive the artifacts of adversarial perturbations in pure color underpainting, which again shows that it is difficult to achieve the invisibility of adversarial perturbations in the DocTR task.

B.2 Different Refresh Rates of Screen

We tested the refresh rate of the monitor on the visual quality effect. Apart from the default style, we also respectively choose a light style and a dark style of underpainting for a comprehensive evaluation. Table 7 illustrates that the visual quality of different underpaintings improves as the refresh rate increases. When the refresh rate reaches 100 *Hz*, the MOS results are almost 5.00 for all three test styles, indicating that the artifacts of adversarial perturbations are invisible to human eyes.

Table 7: MOS results under different refresh rates.

Refresh Rate (Hz)	60	75	100	120
Style 1	3.45	3.75	4.60	5.00
Style 4	4.45	5.00	5.00	5.00
Style 5a	4.45	5.00	5.00	5.00
Style 5b	4.00	4.50	5.00	5.00

B.3 Different Distances

Based on the common distances for humans reading electronic documents on monitors, we evaluate the visual quality of our method at five distances from 40 *cm* to 80 *cm*. As shown in Table 8, the visual quality improves somewhat with increasing distance. At distances below 60 *cm*, the MOS is lower than 4.00, that is, the flicker can be perceived slightly by human eyes but does not affect the normal reading of the text content. However, when the distance reaches to 60 *cm*, the visual quality is significantly improved, and human eyes barely perceive the adversarial perturbations.

Table 8: MOS results of different distances.

Distance (*cm*)	40	50	60	70	80
Style 5a	3.55	3.90	4.45	5.00	5.00
Style 5b	3.45	3.85	4.00	5.00	5.00

B.4 Different Angles

We shoot five different angles ranging from the left 30° to the right 30° of the horizontal perspective. Table 9 shows the MOS results obtained with the same shooting distance of 60 *cm* but different shooting angles. We find that different angles of MOS are slightly different, but they remain close to the score obtained at the original angle 0°. Overall, ProTegO can preserve visual quality well at a variety of angles, and increasing the reading angle can mitigate the flicker effect to some extent.
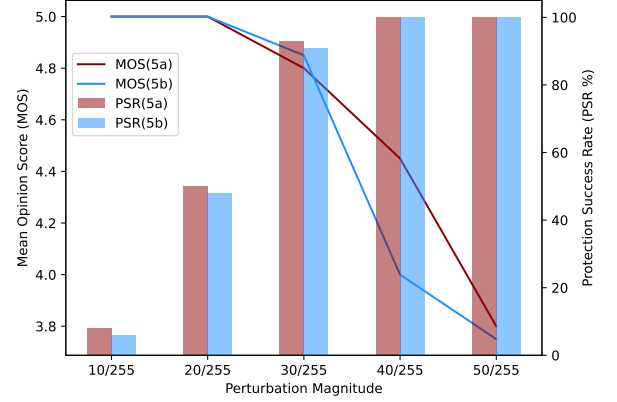
Table 9: MOS results of different angles.

Horizontal (°)	Left 30	Left 15	0	Right 15	Right 30
Style 5a	4.80	4.35	4.45	4.55	4.85
Style 5b	4.85	4.50	4.00	4.65	4.85

C MORE RESULTS OF PROTECTION PERFORMANCE

C.1 Different Magnitudes of Perturbation

There is an inherent trade-off between protection strength and visual quality. Taking our surrogate model STAR-Net [20] as an example, we plot the trade-off figure of the protection success rate (PSR) and mean opinion score (MOS) at different perturbation magnitudes. As shown in Figure 4, we find that human eyes are more likely to perceive adversarial artifacts with increasing magnitude of perturbations. When the perturbations are less than 30/255, our method can almost achieve the same visual quality as the initial underpainting. As the perturbations reach 40/255, the MOS results of *Style 5a/5b* still stay above 4.0, suggesting that our method can tolerate more perturbation intensity while maintaining good visual quality. When the perturbations fall below 30/255, ProTegO obtains the best visual quality with MOS up to 5.00, but the protection performance remains poor. When the perturbations reach 40/255, the MOS results are more than 3.45, and all the PSR results of different

Figure 4: The trade-off between the MOS and PSR.

underpainting styles are more than 90%, which can achieve strong protection. However, when the perturbations increase to 50/255, the visual quality deteriorates, but the improvement in protection performance is not significant. In conclusion, to achieve a comprehensive balance of visual quality and protection performance, we suggest setting the perturbation at 40/255.

C.2 Different Fonts

We also investigate the impact of different fonts on protection performance. Each of the 12 fonts mentioned in Appendix A.1 is tested under the perturbation magnitude of 30/255, and we choose *Style 5a* as the test underpainting. As presented in Table 10, we find that different fonts require different protection strengths but not much difference in general. Additionally, we can also observe that the PSR results of bold fonts are all lower than those of normal fonts. This is not surprising, as bold fonts contain more useful pixels for text recognition models, and they naturally need stronger protection. However, when the perturbations reach 40/255, ProTegO can achieve effective protection on all fonts.

Table 10: PSR results of different fonts.

Font	PSR	Font	PSR
Times New Roman	97	Times New Roman Bold	95
Courier	100	Courier Bold	95
Georgia	94	Georgia Bold	85
Helvetica	97	Helvetica Bold	88
Arial	94	Arial Bold	89
Verdana	99	Verdana Bold	89