Universal Adversarial Watermarking for Text Image Protection via Template-based Underpainting

Fangjun Yan, Yutong Huang, Li Dong, *Member, IEEE*, Jiacheng Deng, Qiuping Jiang, *Senior Member, IEEE*, Zhibo Wang, *Senior Member, IEEE*, Xin Liao, *Senior Member, IEEE*

***Abstract*—Text images are images that contain textual content. They are vulnerable to content piracy through Optical Character Recognition (OCR) and unauthorized distribution. Existing methods based solely on adversarial examples or digital watermarking are insufficient to address both threats effectively. In this work, we propose Universal Adversarial Watermarking (UAW), a dual protection that misleads OCR systems and enables watermarking for traceability simultaneously. Specifically, we first generate two templates for representing watermark bits while attaining adversarial capability against OCR. Then, the watermark message is encoded via structured tiling of the bit templates into the underpainting of text images. Finally, the embedded watermark can be reliably extracted with the aid of watermark block synchronization and a majority voting mechanism. Experimental results demonstrate the effectiveness of the proposed method in terms of adversarial capability and watermarking functionality. We also validate the transferability of UAW across various font sizes, colors and languages, and evaluate the practical usage through real-world applications. The code is publicly available at https://github.com/yfjn/UAW.**

***Index Terms*—Optical character recognition, adversarial watermark, text image.**

I. Introduction

TEXT images refer to images that contain textual content and are widely used to present information in documents, certificates, web pages, chat logs, and internal reports. They carry sensitive or valuable content, making them vulnerable to unauthorized use such as copyright violations and private information leaks. Existing protection mechanisms include copy prevention [1], access control [2], and anti-crawling [3]. However, they remain inadequate against advanced extraction threats. In particular, modern neural network-based Optical Character Recognition (OCR) systems have achieved substantial improvements in text recognition accuracy, enabling malicious actors to capture screenshots at scale. Malicious actors may use automation tools such as Selenium [4] and then apply OCR to extract and redistribute the textual content without authorization. Therefore, practical protection should preserve readability for legitimate users while discouraging OCR-based reconstruction and supporting post-leak tracing.

To address this issue, two existing solutions can mitigate the risks of unauthorized text extraction or redistribution. The first one focuses on disrupting OCR systems through generating adversarial examples, which refers to specially crafted inputs with imperceptible perturbations that cause OCR models to make errors. As illustrated in Fig. 1, pirates can leverage advanced neural network-based OCR to illegally acquire and distribute textual content for profit. To combat this, several studies have proposed adversarial example-based protection methods targeting OCR systems [5]–[7]. By exploiting the intrinsic vulnerability of neural networks, these methods can prevent malicious users from obtaining accurate textual content from text images. Nevertheless, adversarial perturbations are mostly explored in academic studies and are not commonly available as public commercial products.

The second solution addresses the problem of screenshots and can utilize digital watermarking [8], [9] for copyright protection and traceability. As deep learning-based watermarking techniques become more robust, they are used to embed identifiers into images to assert ownership and prevent unauthorized use. Robust watermarking has already been adopted in industry for copyright protection and traceability (*e.g.*, xSecuritas [10], Fasoo [11], and Huawei Cloud [12]). As shown in Fig. 1, watermarks can be embedded in text images, and when the images suffer unauthorized distribution, the embedded watermark can be extracted by content creators or copyright holders as evidence of infringement. However, digital watermarking fails to protect against attacks that utilize OCR to extract textual content from text images.

It is natural to combine the merits of adversarial examples and watermarking into a unified defensive framework. A straightforward strategy is to apply these two components sequentially. However, this leads to degraded performance due to mutual interference. For instance, when the watermark is embedded after adversarial perturbations, it would interfere with the carefully crafted adversarial perturbation, weakening the adversarial effectiveness. Conversely, enforcing adversarial noise after watermark embedding may distort the encoded watermark signal, affecting the watermark reliability in extraction. Essentially, this mutual interference arises from treating the two processes independently and ignoring their interactions. This limitation highlights the need for a

Fangjun Yan, Yutong Huang, Li Dong, and Qiuping Jiang are with the Faculty of Electrical Engineering and Computer Science, Ningbo University, Ningbo 315211, China (e-mail: 2311100096@nbu.edu.cn; 2411100086@nbu.edu.cn; dongli@nbu.edu.cn; jiangqiuping@nbu.edu.cn). *(Fangjun Yan and Yutong Huang contributed equally to this work.) (Corresponding author: Li Dong.)*

Jiacheng Deng is with the School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China (e-mail: 1462492739@qq.com).

Zhibo Wang is with the College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China (e-mail: zhibowang@zju.edu.cn).

Xin Liao is with the College of Cyber Science and Technology, Hunan University, Changsha 410082, China (e-mail: xinliao@hnu.edu.cn).

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2026.3695161

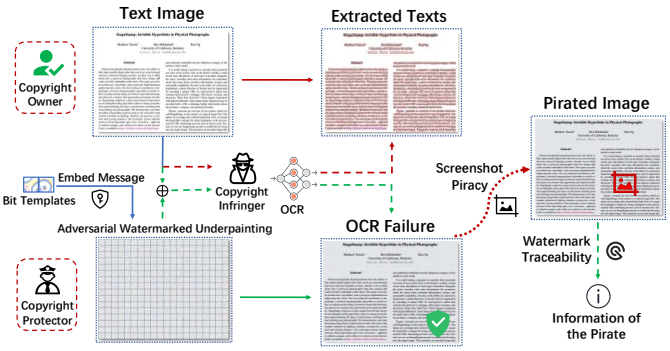


Fig. 1. **Application scenario of the proposed method.** An unprotected text image is vulnerable to OCR and screenshots, which may lead to the extraction and unauthorized distribution of its textual content. By embedding an adversarial watermarked underpainting into the background of the text image, OCR is disrupted, while a robust watermark is preserved for reliable extraction when the protected image is redistributed.

unified design that jointly considers adversarial robustness and watermark reliability. Quiring *et al.* [13] revealed that adversarial perturbations and watermarking distortions share common principles, allowing techniques to be transferred and adapted between the two domains. Later on, Jia *et al.* [14] and Feng *et al.* [15] employed watermarking techniques to generate adversarial examples by treating watermarks as a form of perturbation. However, these methods encounter challenges in practical applications. First, most existing methods are image content-adaptive, requiring the generation of unique perturbations tailored to each text image. This leads to high computational costs and limits scalability when handling large-scale text images. Second, these methods lack scalability across different image sizes. The perturbations are not reusable for images of various dimensions, hindering applicability in the real world.

To address the aforementioned challenges, template-based underpainting is chosen because of the following reasons. It is content-agnostic and works across images without per-image optimization [6]. Its regular pattern also helps keep alignment after distortions, which supports stable watermark decoding [16]. Meanwhile, both OCR resistance and watermark support could be met by template-based underpainting. In this work, we propose Universal Adversarial Watermarking (UAW), a dual-protection framework that unifies adversarial examples with traceable watermarking for text image protection. First, we train a pair of bit templates that degrade the performance of text detection models, preventing OCR extraction. Next, the watermark is modulated into the underpainting of text images by using the learned bit templates, enabling image traceability without compromising readability. Finally, the watermark can be reliably extracted with the aid of watermark block synchronization and a majority voting mechanism, supporting reliable source identification under various distortions. This unified design ensures dual protection by simultaneously resisting OCR and enabling robust traceability. The main contributions of our work are as follows.

- We propose a framework that unifies template-based watermarking with adversarial perturbations to generate an underpainting, which can be readily tiled as the background of text images.
- The proposed method is transferable and scalable, enabling application to diverse text images of varying content and sizes without the need to generate perturbations uniquely for each instance.
- Extensive experiments demonstrate the practical usage of UAW against real-world OCR systems, confirming its robustness and effectiveness in real-world text image protection scenarios.

The rest of this paper is organized as follows. We briefly review the related work in Section II. Section III elaborates the proposed method. Section IV presents the experimental results and analysis, and finally, Section V concludes this work.

II. Related Work

A. Adversarial Attacks on OCR

A typical OCR system is comprised of two key steps: scene text detection (STD) and scene text recognition (STR). First, scene text detection localizes text regions, and then the text recognition component transcribes the content of these regions into readable characters. In this work, we focus on attacking the STD model in OCR systems because accurate STD text localization is crucial for reliable text recognition.

Generally, STD methods can be classified into two research lines, geometry-based and pixel-level classification based methods. Geometry-based methods directly predict the bounding boxes for text regions and excel at handling texts with regular structures, *e.g.,* EAST [17] and TextBoxes++ [18]. Pixel-level classification methods employ pixel-level classification followed by post-processing to generate detection boxes, making them well-suited for irregular or curved text, such as PSENet [19] and PAN++ [20]. One representative method in this category is CRAFT [21], which could localize arbitrarily-shaped text by predicting individual character positions and their affinities. DBNet [22] and DBNet++ [23] employ differentiable binarization for adaptive thresholding, enhancing both accuracy and computational efficiency. Note that the commercial WeChat OCR system [24] is built upon DBNet, demonstrating its practical effectiveness.

Adversarial attacks on OCR can be grouped into image-adaptive attacks and universal attacks, according to how the perturbation is generated. Image-adaptive attacks optimize a dedicated perturbation for each text image. Xu *et al.* [25] proposed a black-box method that leverages an adaptive-discrete differential evolution algorithm to protect privacy in STR. Xiang *et al.* [5] applies localized adversarial perturbations to achieve fine-grained control across different stages of scene text editing. Per-image optimization often achieves strong attack efficacy, but it relies on repeated optimization steps, which increases computation and limits large-scale, plug-and-play deployment.

In contrast, universal attacks learn a reusable perturbation that can be applied to various images. Wu *et al.* [26] proposed a universal adversarial method based on probability maps and threshold loss. UDUP [6] could protect text images from unauthorized text extraction by modifying the underpainting with

fixed-size adversarial patches tiled in text image backgrounds. A single perturbation enables plug-and-play protection without per-image optimization. However, the attack behavior can be sensitive to the target OCR detector and its training pipeline. Since the learned perturbation is coupled to the training detector, it may reduce transferability when the target changes.

Importantly, this line of work is designed for OCR disruption. Even when the OCR attack succeeds, it does not provide an extractable watermark for traceability, and once the adversarial effect fails, it offers no additional mechanism to deter or trace screenshot-based redistribution.

B. Template-based Image Watermarking

Watermarking techniques can be categorized into universal and non-universal approaches. Non-universal methods [27] require image-specific watermark design, which limits efficiency when processing large-scale image instances. In contrast, universal methods enable efficient deployment at scale, with template-based watermarking serving as a representative approach that embeds predefined or pretrained templates into images, enjoying high efficiency and scalability.

Template-based watermarking methods improve robustness through template-based synchronization and distortion compensation. Fang *et al.* [28] introduced a two-stage learning framework for template-based watermarking. Ma *et al.* [29] proposed a symmetry-based watermark synchronization scheme based on autocorrelation function. This method can resist both local geometric distortions of random bending and global geometric distortions of rotation and scaling. Fang *et al.* [16] applied template-based underpainting watermarking to text images. RA Code [30] and RU Code [31] introduced bit templates and modulation strategies to aesthetically enhance QR codes through block-wise modulation. To resist screen photography, Fang *et al.* [9] proposed an algorithm based on scale-invariant feature transformation for accurately locating watermark embedding regions.

Template-based pipelines are scalable to provide robustness through synchronization and decoding stability. However, they are designed for watermark recovery and do not address the joint constraints required when OCR disruption and watermark traceability must hold simultaneously for text image protection.

C. Adversarial Watermarking

Adversarial watermarking refers to watermarking schemes that provide traceability while also exhibiting adversarial effects on target models. Existing studies can be grouped by their application setting into watermarking against image classification and watermarking against image forgery.

For watermarking against image classification, Feng *et al.* [15] optimized an invisible watermark as a perturbation that alters the prediction of a neural classifier while preserving visual quality. Wang *et al.* [32] trained a fusion model to generate invisible adversarial watermark images for discouraging unauthorized identification. Zhang *et al.* [33] proposed a one-time embedding strategy that misleads unauthorized face recognition while retaining verification for authorized users. Adv-watermark [14] adopted visible marks to influence image classification. These methods are tailored to the characteristics and decision boundaries of classification or face recognition models, and the same design assumptions are not aligned with text image pipelines that depend on reliable text localization. FAWA [7] moved closer to the text domain by overlaying a semi-transparent logo and optimizing perturbations in the watermark region to affect STR. Nevertheless, its design introduces visible identifiers and requires per-image optimization inside a mask, which limits scalability across diverse text images.

For watermarking against image forgery, Zhang *et al.* [34] embedded perceptual adversarial codes to maintain attribution under face swapping. Xu *et al.* [35] incorporated adversarial watermarks in the DCT domain to defend against GAN-based synthesis under JPEG compression. Qiao *et al.* [36] developed a scalable expansion scheme to combat forgery models. These approaches focus on resisting generative manipulation and are evaluated in face media settings. Their objectives and threat models differ from the STD scenario, in which preventing reliable text extraction is the primary concern.

Overall, adversarial watermarking oriented to STR remains underexplored at the scale and diversity of real-world text images. Most of existing methods are designed for natural images, restricted to one-bit watermarks, or rely on per-image optimization that makes universal deployment difficult. Motivated by these gaps, we adopt a template-based strategy that embeds adversarial watermarks into the background of text images.

III. Proposed Universal Adversarial Watermarking

The overall framework is illustrated in Fig. 2. In general, the proposed framework consists of three stages. First, in the bit templates generation stage, two bit templates are trained to disrupt the text detection model, preventing the text detection via OCR. Second, in the watermark modulation stage, the given message bitstream is modulated to a watermark unit using bit templates, and then watermark unit is fused with the text image as underpainting. Third, in the watermark extraction stage, the message is decoded via watermark block synchronization and state determination from redistributed text images. In the following, we elaborate each stage in detail.

A. Bit Templates Generation

The primary goal of this stage is to create two distinct templates, which serve both as information carriers (representing bit 0 and bit 1, respectively) and as universal adversarial templates against OCR systems. When deployed as the background of text images, they can interfere with OCR text detection while simultaneously preserving text readability.

The process of the bit template generation is illustrated in Fig. 3, which is an iterative optimization procedure. Initially, the two bit templates are initialized with uniform noise. Then for each optimization iteration, it involves three steps: randomly tiling the bit templates to construct an adversarial underpainting, fusing the underpainting with the background

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2026.3695161

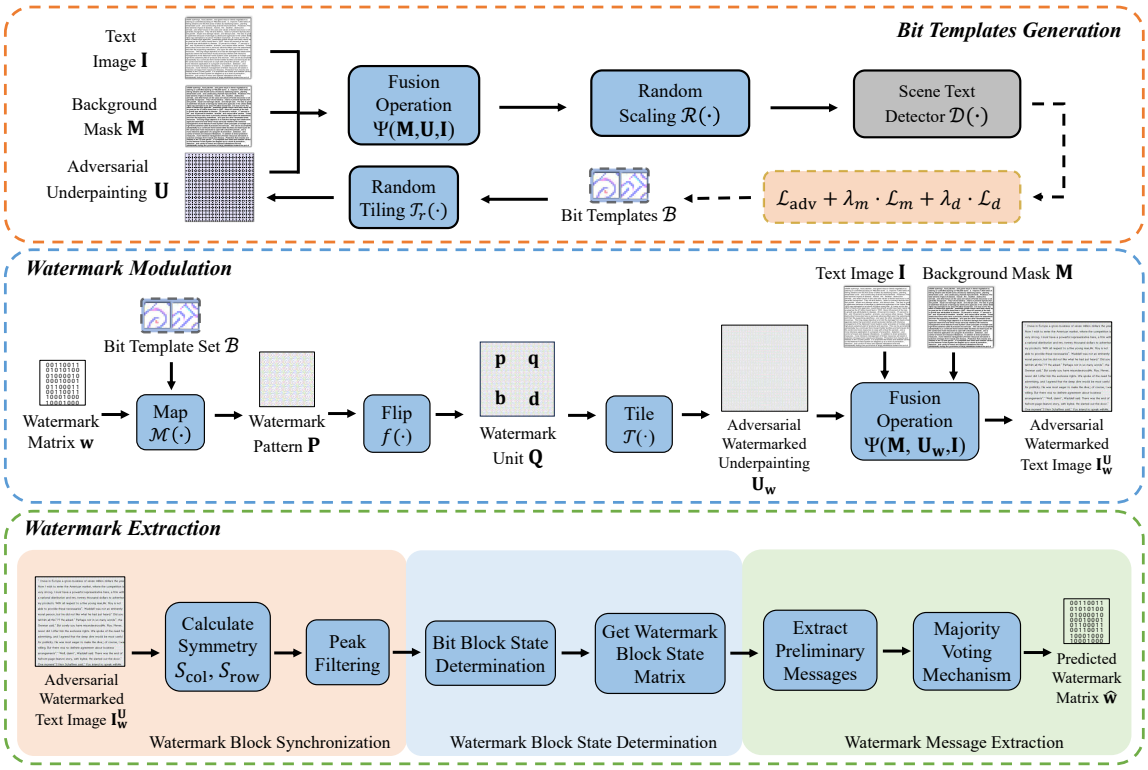


Fig. 2. **Overview of the proposed UAW framework.** In bit templates generation, bit templates are generated and optimized to disrupt text detection models. In watermark modulation, binary watermark message is encoded into watermark unit using bit templates and fused into the background of the text image, while preserving the readability of the foreground. The watermark extraction stage involves watermark block synchronization and watermark block state determination, enabling the recovery of the watermark message.

regions of the text image, and applying a random scaling to simulate potential resizing in practice. The resulting image is then fed into a text detection model to evaluate adversarial effectiveness. The loss is back-propagated through a gradient-based optimizer to update the bit templates, progressively enhancing both adversarial transferability and robustness.

Specifically, let $\mathcal{I} = \{\mathbf{I}_i, \mathbf{M}_i\}_{i=1}^{N}$ denote a text image dataset, where $\mathbf{I}_i \in [0,1]^{h\times w}$ is a text image. $\mathbf{M}_i \in \{0,1\}^{h\times w}$ is the binary mask of the text image, in which 0 indicates a text pixel and 1 denotes a background pixel. Let the STD model be denoted by $\mathcal{D}(\cdot)$. Given an input text image $\mathbf{I}$, it outputs a probability map $\mathcal{D}(\mathbf{I}) \in [0,1]^{h\times w}$, where each value indicates the predicted probability of a pixel belonging to a text region. The two bit templates to be learned are denoted as the set $\mathcal{B} = \{\mathbf{B}^0, \mathbf{B}^1\}$, where each template $\mathbf{B}^i$ $(i \in 0,1)$ encodes a distinct binary watermark bit. These templates are confined to the intensity range $[1-\epsilon, 1]^{s\times s}$, where $s\times s$ is the template size and $\epsilon \in [0,1]$ controls the maximum perturbation magnitude.

The overall workflow of bit template deployment proceeds as follows. A batch of text images $\{\mathbf{I}_i\}$ and their corresponding background masks $\{\mathbf{M}_i\}$ are first sampled from the dataset $\mathcal{I}$. To construct the adversarial underpainting $\mathbf{U} \in [1-\epsilon, 1]^{h\times w}$, a random tiling operation $\mathcal{T}_r(\cdot)$ is applied to the bit template set $\mathcal{B}$, such that $\mathbf{U} = \mathcal{T}_r(\mathcal{B})$. This process adapts fixed-size $s\times s$ templates to arbitrary image dimensions and ensures their universality across various text layouts. More specifically, the text image is divided into $h' \times w'$ non-overlapping blocks, where $h' = h/s$ and $w' = w/s$. Each block $\mathbf{U}_{[x,y]}$ is randomly assigned one of the two bit templates from $\mathcal{B}$, allowing arbitrary combinations of bit templates to maintain adversarial effectiveness against STD models. Once all blocks are filled, the final underpainting $\mathbf{U}$ is fused with the original image $\mathbf{I}$ through a fusion operation $\Psi(\cdot)$,

$$\mathbf{I}^{\mathbf{U}} = \Psi(\mathbf{M}, \mathbf{U}, \mathbf{I}) = \mathbf{M} \odot \mathbf{U} + (\mathbf{1} - \mathbf{M}) \odot \mathbf{I}, \tag{1}$$

where $\odot$ denotes element-wise multiplication. To enhance the image size-resilience of the bit templates, the adversarial text image $\mathbf{I}^{\mathbf{U}}$ is subjected to random scaling $\mathcal{R}(\cdot)$ before being fed into the target STD model. This strategy was also adopted in the work [6].

The key purpose of adversarial image $\mathbf{I}^{\mathbf{U}}$ is to degrade the text detection of OCR system while retaining the text readability, with the bit templates encoding the watermark bits. In this light, to obtain two distinctive bit templates to fulfill this purpose, we formulate the following constrained optimization problem,

$$\begin{aligned} \mathcal{B}^* = \{\mathbf{B}^{0*}, \mathbf{B}^{1*}\} = \underset{\{\mathbf{B}^0, \mathbf{B}^1\}}{\arg\min}\, \mathbb{E}_{\mathbf{I}\sim\mathcal{I}}[\mathcal{L}_{\text{adv}} + \lambda_m \cdot \mathcal{L}_m + \lambda_d \cdot \mathcal{L}_d] \\ \text{s.t.: } \|\mathbf{1} - \mathbf{B}^i\|_\infty \le \epsilon, \quad i \in \{0,1\} \end{aligned} \tag{2}$$

where the loss function includes three parts: the adversarial loss $\mathcal{L}_{\text{adv}}$, the multi-intermediate layer loss $\mathcal{L}_m$, and the discriminative loss $\mathcal{L}_d$. The hyperparameter λ_m and λ_d are used to balance among the losses. The constraint ensures that the perturbations introduced by the bit templates are limited to

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2026.3695161

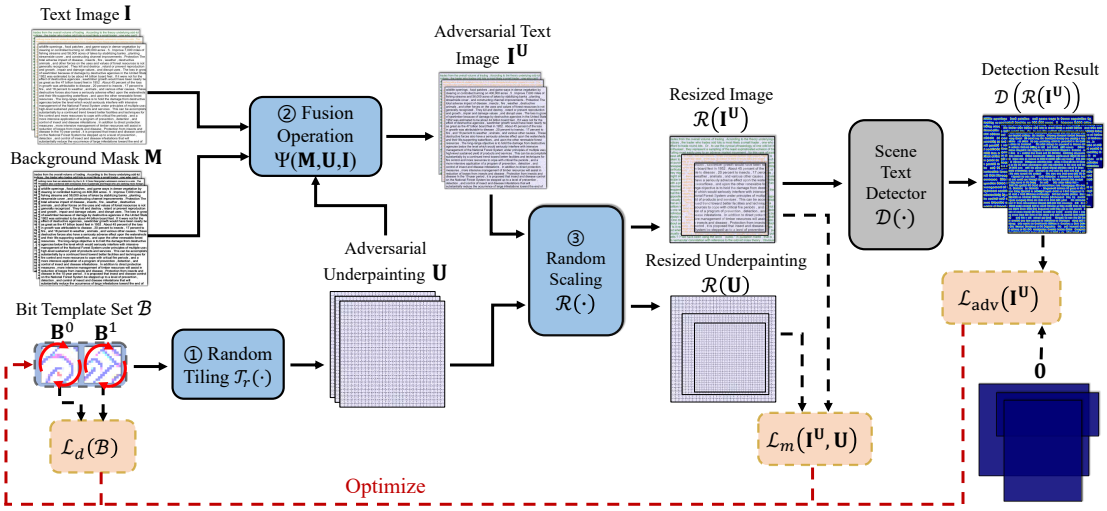


Fig. 3. **Framework for bit templates generation.** It begins by randomly tiling two bit templates, $\mathbf{B}^0$ and $\mathbf{B}^1$, to construct an adversarial underpainting $\mathbf{U}$. The fusion operation $\Psi(\cdot)$ then combines this underpainting with the text image to preserve foreground text while replacing the background. This fused image is applied random scaling to simulate potential distortion. Finally, adversarial loss, multi-intermediate layer loss, and discriminative loss are computed. The bit templates are iteratively optimized, ensuring both robustness and transferability.

preserve the readability of the foreground text. In the sequel, we discuss each loss function in more detail.

Adversarial Loss: The bit templates are supposed to protect textual content from detection by STD model of an OCR system. This means that, when the randomly-resized adversarial image $\mathcal{R}(\mathbf{I}^{\mathbf{U}})$ is fed into the target model $\mathcal{D}(\cdot)$, the output probability map should assign low probabilities to text regions. We thus define the adversarial loss $\mathcal{L}_{\text{adv}}$ as

$$\mathcal{L}_{\text{adv}}(\mathbf{I}^{\mathbf{U}}) = \frac{1}{hw}\|\mathcal{D}(\mathcal{R}(\mathbf{I}^{\mathbf{U}})) - \mathbf{0}\|_2^2. \tag{3}$$

Multi-Intermediate Layer Loss: Previous works [37]–[39] have found that the adversarial examples generated based on final predicted probability maps may overfit to the architecture or feature representation of the source model. To address this issue, a multi-intermediate layer loss is introduced as part of the optimization objective to improve transferability. Specifically, the $\mathcal{L}_m$ loss is designed to minimize the distance between the adversarial text image and the adversarial underpainting in intermediate layers of the target STD model. This loss could reduce the impact of different texts on the defensive effect of underpainting, which can be expressed by

$$\mathcal{L}_m(\mathbf{I}^{\mathbf{U}},\mathbf{U}) = \frac{1}{K}\sum_{k=1}^{K}\|\mathcal{D}_k(\mathcal{R}(\mathbf{I}^{\mathbf{U}})) - \mathcal{D}_k(\mathcal{R}(\mathbf{U}))\|_2^2, \tag{4}$$

where K is the number of intermediate layers, and $\mathcal{D}_k(\cdot)$ is the output of the k-th intermediate layer of the STD model.

Discriminative Loss: Each bit requires a unique bit template. This loss enhances watermark extraction robustness by minimizing similarity between templates computed by

$$\mathcal{L}_d(\mathcal{B}) = -\frac{1}{s^2}\|\mathbf{B}^0 - \mathbf{B}^1\|_1. \tag{5}$$

To solve optimization problem (2), we employ a gradient-based optimizer to determine the bit template set $\mathcal{B}$. The gradient with momentum is computed as

$$\mathbf{g}_{t+1} = \mu\cdot\mathbf{g}_t + \frac{\nabla_{\mathcal{B}}\mathcal{L}^*}{\|\nabla_{\mathcal{B}}\mathcal{L}^*\|_1}, \tag{6}$$

where $\mathcal{L}^* = \mathcal{L}_{\text{adv}} + \lambda_m\cdot\mathcal{L}_m + \lambda_d\cdot\mathcal{L}_d$ is the total loss, and μ is the momentum decay parameter. The gradient is normalized to ensure consistent scale across different mini-batches, which helps stabilize the optimization. Then, the bit template set $\mathcal{B}$ is updated using Projected Gradient Descent (PGD) [40],

$$\mathcal{B}_{t+1} = \text{Clip}_\epsilon\{\mathcal{B}_t - \alpha\cdot\text{sign}(\mathbf{g}_{t+1})\}, \tag{7}$$

where $\text{Clip}_\epsilon\{\cdot\}$ is a function that clips the input variable to ensure that $\|\mathbf{1}-\mathbf{B}^i\|_\infty \le \epsilon$ for $i\in\{0,1\}$, and α is the learning rate. Note that the $\text{Clip}_\epsilon(\cdot)$ operation could enforce the constraint in (2), bounding the perturbation magnitude induced by the bit templates to ϵ.

As can be seen, after T iterations, adversarially optimized bit template set $\mathcal{B}^*$ is resistant to STD models. These templates will be deployed in the next watermark modulation stage to encode binary watermark bits.

B. Watermark Modulation

In this stage, we construct the adversarial watermarked underpainting $\mathbf{U}_{\mathbf{w}}$ by systematically combining the bit templates with the watermark message. First, we generate a watermark pattern $\mathbf{P}$ by mapping the binary watermark matrix $\mathbf{w}$ to the corresponding bit templates. Second, we apply flipping operations to $\mathbf{P}$ to obtain a watermark unit $\mathbf{Q}$. Third, we perform tiling on $\mathbf{Q}$ to assemble the adversarial watermarked underpainting $\mathbf{U}_{\mathbf{w}}$. Finally, we fuse $\mathbf{U}_{\mathbf{w}}$ with the text image to produce the adversarial watermarked text image $\mathbf{I}_{\mathbf{w}}^{\mathbf{U}}$.

1) Bit-template Mapping: Given a binary watermark matrix $\mathbf{w}$ of size $a\times b$, a watermark pattern $\mathbf{P}$ is constructed by mapping each bit in $\mathbf{w}$ to its corresponding bit template. As illustrated in Fig. 4, a bit-template mapping mechanism

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2026.3695161

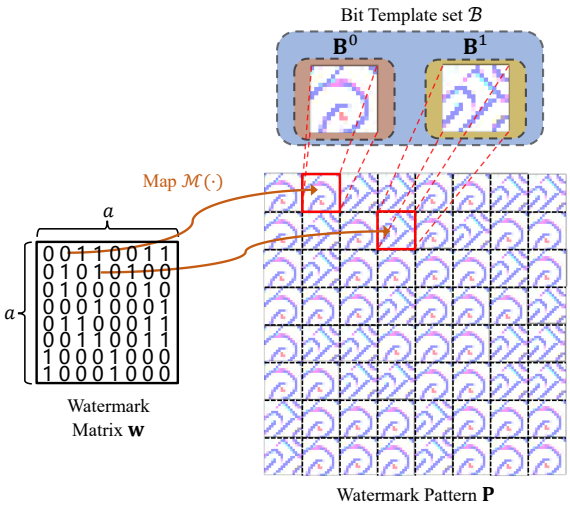


Fig. 4. **Watermark mapping.** The watermark matrix $\mathbf{w}$ is used to construct the watermark pattern $\mathbf{P}$ by applying bit templates $\mathbf{B}^0$ and $\mathbf{B}^1$. The example uses an 8×8 watermark block for illustration only.

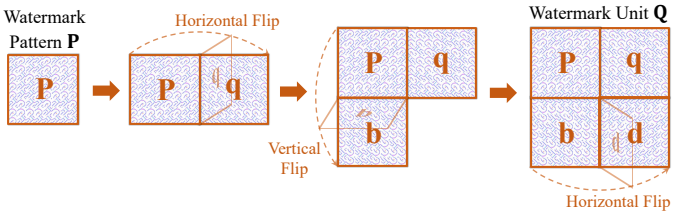


Fig. 5. **Flip-based watermark unit construction.** The horizontal and vertical flips are applied to the watermark pattern $\mathbf{P}$ to create a watermark unit $\mathbf{Q}$.

is adopted to embed the watermark message. The mapping process can be expressed concisely as

$$\mathbf{P} = \mathcal{M}(\mathbf{w}, \{\mathbf{B}^0, \mathbf{B}^1\}). \tag{8}$$

where $\mathcal{M}(\cdot)$ denotes the mapping function that replaces each bit in $\mathbf{w}$ with the corresponding bit template.

Specifically, the bit templates $\mathbf{B}^0$ and $\mathbf{B}^1$, obtained in the template generation stage, are used to represent watermark bit values 0 and 1, respectively. Each watermark bit in $\mathbf{w}$ is mapped to a corresponding $s \times s$ bit template. The entire watermark matrix $\mathbf{w}$ is expanded into a watermark pattern $\mathbf{P}$ of size $as \times bs$ according to the following rule,

$$\mathbf{P}_{[r,c]} = \begin{cases} \mathbf{B}^0, & \text{if } \mathbf{w}_{r,c} = 0 \\ \mathbf{B}^1, & \text{if } \mathbf{w}_{r,c} = 1 \end{cases}, \tag{9}$$

where $\mathbf{P}_{[r,c]}$ denotes the $s \times s$ block in $\mathbf{P}$ corresponding to the r-th row and c-th column, *i.e.,* $\mathbf{P}[rs : rs+s-1,\ cs : cs+s-1]$, and $\mathbf{w}_{r,c}$ represents the bit value at the r-th row and c-th column of the watermark matrix. Through this construction, the watermark pattern not only embeds the watermark information but also ensures the independence and recognizability of each bit. This lays the foundation for the identification of each bit in the watermark extraction stage.

2) Flip-based Unit Construction: In real-world applications, text images may undergo various geometric distortions such as cropping, rotation, and flipping, which can severely affect the accuracy of watermark extraction. To address this, we adopt a flip-based construction mechanism to enhance the structural robustness of the watermark.

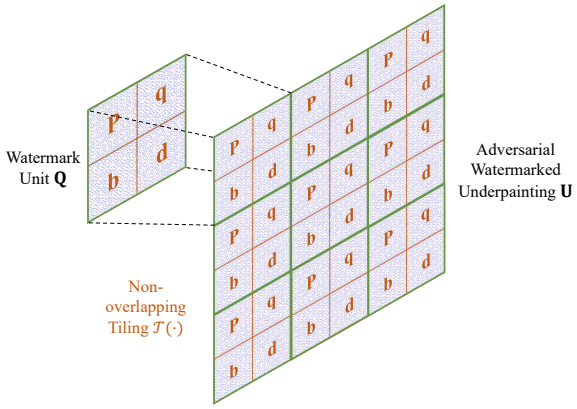


Fig. 6. **Tiling for adversarial watermarked underpainting generation.** The watermark unit $\mathbf{Q}$ is tiled non-overlappingly across the image, followed by cropping to fit the dimensions of the target image, resulting in an adversarial watermarked underpainting $\mathbf{U}_\mathbf{w}$.

Flipping operations are applied to the watermark pattern $\mathbf{P}$ to generate symmetrical variants. There are two types of flipping operations: horizontal flipping $\mathrm{hflip}(\cdot)$ and vertical flipping $\mathrm{vflip}(\cdot)$. As illustrated in Fig. 5, by applying these operations, four variants of $\mathbf{P}$ are produced, including the original pattern without flip, a horizontally flipped version $\mathrm{hflip}(\mathbf{P})$, a vertically flipped version $\mathrm{vflip}(\mathbf{P})$ and a version with both horizontal and vertical flips $\mathrm{vflip}(\mathrm{hflip}(\mathbf{P}))$. These four variants are arranged into a 2×2 grid in row-major order to construct the watermark unit $\mathbf{Q}$, which can be expressed as

$$\mathbf{Q} = f(\mathbf{P}, \{\mathrm{hflip}, \mathrm{vflip}\}) \tag{10}$$

where $f(\cdot)$ denotes the flipping operation. This design encodes symmetrical features into the watermark unit, providing reliable anchors for alignment detection during extraction.

3) Tiling for Underpainting Generation: To apply the watermark to text images of arbitrary size, we tile the watermark unit $\mathbf{Q}$ non-overlappingly across both height and width dimensions. As illustrated in Fig. 6, this operation replicates $\mathbf{Q}$ until the entire background area is covered, and the tiled result is subsequently cropped to match the dimensions of the target image. The tiling process is expressed as

$$\mathbf{U}_\mathbf{w} = \mathcal{T}(\mathbf{Q}) \tag{11}$$

where $\mathcal{T}(\cdot)$ denotes the tiling operation. The resulting adversarial watermarked underpainting $\mathbf{U}_\mathbf{w}$ inherits both the adversarial capability from the bit templates and the semantic information from the embedded watermark message. Moreover, this tiling strategy enhances robustness against local occlusion and partial distortion during watermark extraction.

4) Fusion with Text Image: The goal of this step is to produce adversarial watermarked text images that preserve the visual appearance of the original content while embedding a traceable watermark and maintaining adversarial robustness. To this end, the adversarial underpainting $\mathbf{U}_\mathbf{w}$ is fused with the original image $\mathbf{I}$ using the following operation,

$$\mathbf{I}_\mathbf{w}^\mathbf{U} = \Psi(\mathbf{M}, \mathbf{U}_\mathbf{w}, \mathbf{I}) = \mathbf{M} \odot \mathbf{U}_\mathbf{w} + (\mathbf{1} - \mathbf{M}) \odot \mathbf{I}, \tag{12}$$

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2026.3695161

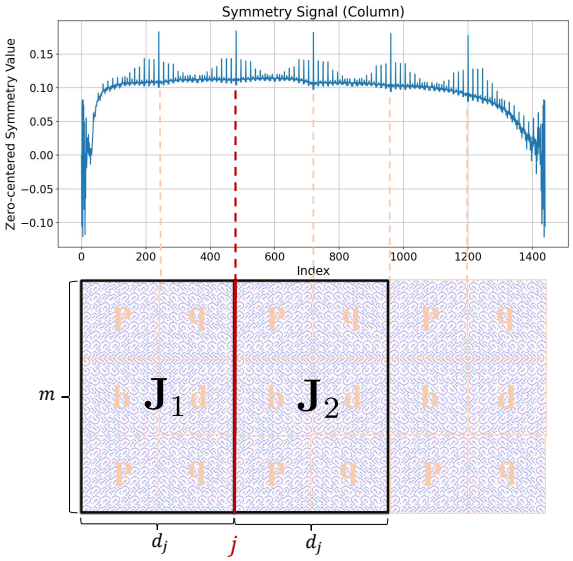


Fig. 7. **Symmetry signal and calculation diagram.** The upper plot shows the zero-centered symmetry values computed along column indices, where prominent peaks indicate potential boundaries of watermark blocks. The lower illustration depicts how the text image is split at column j into two parts $\mathbf{J}_1$ and $\mathbf{J}_2$, each of width d_j and height m, for symmetry calculation. Candidate boundaries are highlighted with dashed lines.

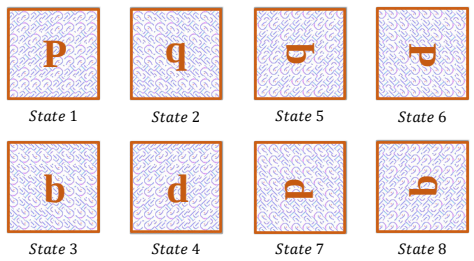


Fig. 8. **Eight possible states of watermark blocks.** As only two bit templates are stored, it is necessary to compare each bit block in a watermark block with the templates for each state. The entire watermark block is assumed to be in one of eight possible states (combinations of flipping and rotation). The state with the highest correlation is selected as the most likely state for each watermark block.

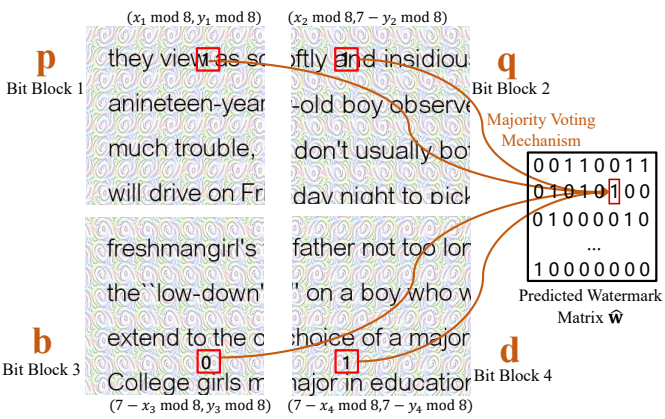


Fig. 9. **Majority voting mechanism for watermark extraction.** The red lines indicate the aggregation of corresponding bits across multiple watermark blocks based on majority voting. The example uses an 8×8 watermark block for illustration only. When a watermark block is partially truncated at the image boundary, any contained full $s \times s$ bit block is still used in voting.

where background pixels in $\mathbf{I}$ are replaced with those from $\mathbf{U}_\mathbf{w}$, and foreground text regions are kept intact. The resulting image $\mathbf{I}_\mathbf{w}^\mathbf{U}$ combines adversarial characteristics with embedded watermark message, enabling dual protection.

It is worth noting that this stage differs fundamentally from the bit templates generation phase. There, the underpainting $\mathbf{U}$ is formed by randomly tiling bit templates, aiming solely to induce adversarial perturbations. In contrast, the current process introduces semantic information through the structured tiling of the bit templates based on the watermark matrix $\mathbf{w}$, which is further operated with flipping and tiling to support reliable watermark extraction.

C. Watermark Extraction

The extraction process can be divided into three parts: watermark block synchronization, watermark block state determination and watermark message extraction. To distinguish from the bit template $\mathbf{B}$ and the watermark pattern $\mathbf{P}$ defined during the modulation stage, we refer to the subdivisions identified in an adversarial watermarked text image $\mathbf{I}_\mathbf{w}^\mathbf{U}$ during extraction as bit block $\mathbf{B}^\mathbf{I}$ and watermark blocks $\mathbf{P}^\mathbf{I}$, respectively.

1) Watermark Block Synchronization: In the watermark modulation stage, flipping and tiling operations introduce periodic symmetry. As illustrated in Fig. 7, these symmetrical features provide useful cues for locating the boundaries of watermark blocks during extraction. Taking column symmetry detection as an example, inspired by the method [16], we calculate the symmetry signal S_{col} at each column index as follows,

$$S_{\text{col}}(j) = \frac{\sum_{x=1}^{m} \sum_{y=1}^{d_j} C(\mathbf{J}_1) C(\Phi_{\text{col}}(\mathbf{J}_2))}{m \times d_j}, \tag{13}$$

where $\mathbf{J}_1$ and $\mathbf{J}_2$ denote the left and right parts of the image separated by column index j, Φ_{col} represents the column flipping operation, $C(\cdot)$ denotes zero-centered normalization, m and d_j are the height and width of the analyzed region, respectively. To avoid interference from text, we calculate symmetry only in the background areas. Similarly, the symmetry signal in the row direction S_{row} can be obtained using the same way. Some peak points can be observed from the symmetry signal, which need to be extracted to identify the boundaries of watermark blocks.

To analyze the boundaries of watermark blocks, the symmetry waveform is divided into segments according to the width of the watermark unit as. For each test point L_c in the waveform, we compute its score using the following formula,

$$V_c = \frac{\sum_{j=0}^{1} |S_c(L_c + j) - S_c(L_c + j - 1)|}{2} - \sqrt{\frac{1}{k-1} \sum_{j=1}^{k-1} (S_c(j) \setminus S_c(L_c) - \mathbb{E}(S_c \setminus S_c(L_c)))^2}, \tag{14}$$

where S_c is the segment that contains L_c, k denotes the number of points in S_c, and $\mathbb{E}(S_c \setminus S_c(L_c))$ represents the expectation of all values in S_c excluding the test point itself. The score V_c is computed by measuring the average absolute

TABLE I
COMPARISON OF DIFFERENT TEXT IMAGE PROTECTION METHODS. *Plug-and-play* REFERS TO WHETHER IT IS SUITABLE FOR ARBITRARY CHARACTERS. *Arbitrary size* REFERS TO WHETHER IT IS VALID FOR DIFFERENT TEXT IMAGE SIZES.

Category	Method	Functionality		Universality		Robustness	
		OCR resistance	Watermark support	Plug-and-play	Arbitrary size	Scaling	JPEG
Traditional text image watermarking	[42]	✗	✓ 1 bit / curved letter	✗	✓	✗	✗
	[43]	✗	✓ 1 bit / word group	✗	✓	✓	✗
	[44]	✗	✓ 1 bit / letter	✗	✓	✓	✗
Template-based watermarking	[9]	✗	✓ 64 bits	✓	✓	✓	✓
	[28]	✗	✓ 64 bits	✓	✓	✓	✓
	[29]	✗	✓ 64 bits	✓	✓	✓	✓
	[16]	✗	✓ 128 bits	✓	✓	✓	✓
Adversarial attacks against OCR	[5]	✓	✗	✗	✗	✗	✗
	[45]	✓	✗	✗	✗	✓	✗
	[46]	✓	✗	✗	✗	✓	✗
	[25]	✓	✗	✗	✗	✗	✗
Universal adversarial attacks against OCR	[6]	✓	✗	✓	✓	✓	✓
	[26]	✓	✗	✗	✗	✗	✗
Adversarial watermark attack on OCR (FAWA)	[7]	✓	✓ visible watermarking	✗	✓	✗	✗
The Proposed UAW		✓	✓ 64 bits	✓	✓	✓	✓

difference between the test point and its immediate neighbors, which captures the sharpness of the local peak. Then, the standard deviation of other values in the segment is calculated to estimate the background fluctuation level. By subtracting the background variability from the local sharpness, the score emphasizes points that are prominent from their surroundings.

All points are ranked according to their scores, and the top-ranked ones are selected as candidates for symmetry axes. The valid watermark blocks are subsequently segmented from the watermarked text image $\mathbf{I}_\mathbf{w}^\mathbf{U}$ according to the symmetry axes.

2) Watermark Block State Determination: After the watermark blocks are obtained, we can divide them into bit blocks according to the size a. Each bit block originates from either $\mathbf{B}^0$ or $\mathbf{B}^1$ possibly in a flipped form. As illustrated in Fig. 8, there are eight transformation states resulting from flipping and rotation. To determine the state of a watermark block, we analyze the states of its constituent bit blocks.

We adopt normalized cross-correlation (NCC) metric [41] to measure the similarity between candidate bit blocks and the two bit templates under different transformation states, as NCC is robust to variations in brightness and contrast,

$$\mathrm{NCC}(\mathbf{B}, \mathbf{B}^\mathbf{I}) = \frac{\sum_{i=1}^{n} \left(\mathbf{B}_i - \bar{\mathbf{B}}\right)\left(\mathbf{B}_i^\mathbf{I} - \bar{\mathbf{B}^\mathbf{I}}\right)}{\sqrt{\sum_{i=1}^{n} \left(\mathbf{B}_i - \bar{\mathbf{B}}\right)^2} \sqrt{\sum_{i=1}^{n} \left(\mathbf{B}_i^\mathbf{I} - \bar{\mathbf{B}^\mathbf{I}}\right)^2}}, \quad (15)$$

where bit block $\mathbf{B}^\mathbf{I}$ is segmented from the adversarial watermarked text image $\mathbf{I}_\mathbf{w}^\mathbf{U}$ and $\mathbf{B}$ denotes a reference bit template, which can be either $\mathbf{B}^0$ or $\mathbf{B}^1$. The variable n refers to the number of elements in the bit block. For each candidate state, we calculate the NCC with both $\mathbf{B}^0$ and $\mathbf{B}^1$ and assign to each bit block the state with the highest NCC.

After determining the states of all bit blocks in a given watermark block, we assign the most frequently occurring bit block state to the overall state of the watermark block. This procedure is applied to all detected watermark blocks in the adversarial watermarked text image $\mathbf{I}_\mathbf{w}^\mathbf{U}$, resulting in a state matrix of all the watermark blocks.

3) Watermark Message Extraction: After obtaining the matrix of watermark block states, we first apply an inverse

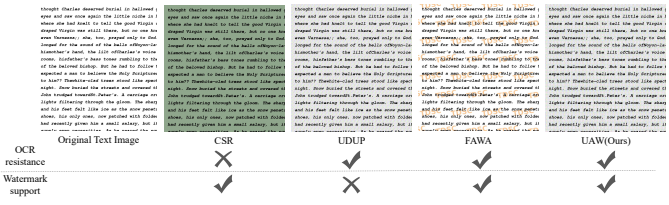


Fig. 10. **Visualization of different text image protection methods.** Only FAWA and the proposed method UAW can provide OCR resistance and watermark tracing capability simultaneously. However, they differ in watermarking form and in the intended usage scenario.

transformation to each watermark block based on its state. For each bit block $\mathbf{B}^\mathbf{I}$ at position (r, c) in the watermark block $\mathbf{P}^\mathbf{I}$, we compute the NCC with both bit templates $\mathbf{B}^0$ and $\mathbf{B}^1$. The preliminary extraction process is as follows,

$$\widehat{\mathbf{w}}_{r,c} = \begin{cases} 1, & \text{if } \mathrm{NCC}(\mathbf{B}^\mathbf{I}, \mathbf{B}^1) \geq \mathrm{NCC}(\mathbf{B}^\mathbf{I}, \mathbf{B}^0) \\ 0, & \text{if } \mathrm{NCC}(\mathbf{B}^\mathbf{I}, \mathbf{B}^1) < \mathrm{NCC}(\mathbf{B}^\mathbf{I}, \mathbf{B}^0) \end{cases}. \quad (16)$$

In this manner, the preliminary watermark for each watermark block $\mathbf{P}^\mathbf{I}$ is determined based on the higher NCC value between the two bit templates.

To obtain the final watermark, we use the majority voting mechanism that aggregates the corresponding bits across all watermark blocks, as shown in Fig. 9. This voting mechanism improves the robustness of watermark extraction. Additionally, complete bit blocks located near the image boundaries are also included in the voting. Here, a complete bit block refers to an $s \times s$ block that fully matches the size of one bit template, where s is the template size. Although some of these bit blocks may belong to incomplete watermark blocks near the image boundary, they still contain valid local template information and are therefore included in majority voting to maximize the use of available evidence. The relative position $[r, c]$ of one $a \times b$ bit block in the watermark block $\mathbf{P}^\mathbf{I}$ is as follows,

$$\begin{aligned} r &= \begin{cases} a - 1 - x \bmod a, & \text{if vflip occurs} \\ x \bmod a, & \text{otherwise} \end{cases}, \\ c &= \begin{cases} b - 1 - y \bmod b, & \text{if hflip occurs} \\ y \bmod b, & \text{otherwise} \end{cases}, \end{aligned} \quad (17)$$

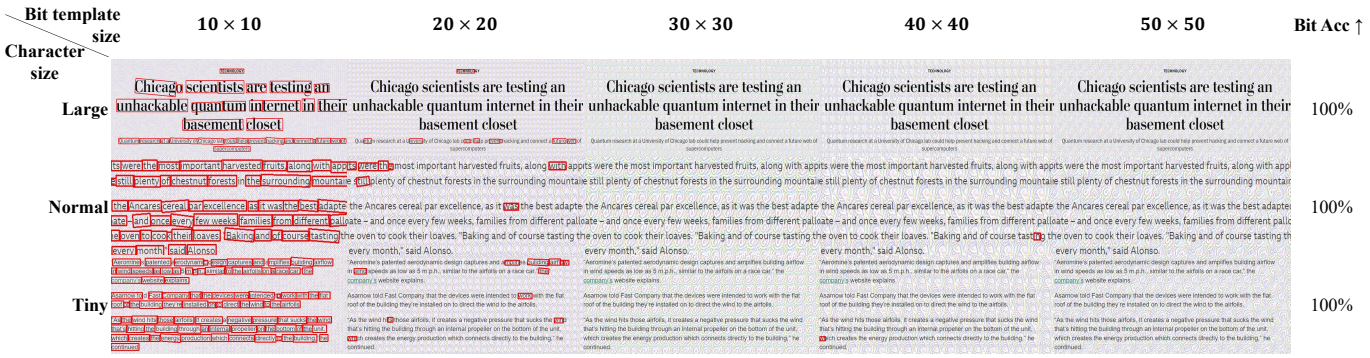


Fig. 11. **Text detection and watermark extraction results with different font sizes.** The left column shows font sizes of tiny (15px), normal (30px), and large (45px). The upper row represents different bit template sizes. In the center, we present the text detection results from CRAFT. The adversarial effect remains stable across all font sizes when the bit template size is at least 20×20. The right column demonstrates that watermark extraction using 30×30 bit templates achieves 100% accuracy over 100 test images.

where $[x, y]$ represents the absolute position of the bit block in the adversarial watermarked text image $\mathbf{I}_{\mathbf{w}}^{\mathbf{U}}$. The flags vflip and hflip specify whether a bit block has undergone vertical or horizontal flipping, respectively.

IV. Experiments

This section evaluates the proposed Universal Adversarial Watermark (UAW) method. We first detail the experimental setup, including the metrics, datasets, and implementation. Then, we qualitatively compare UAW with 14 text protection methods and conduct quantitative effectiveness assessments. Further analysis examines the universality of UAW across various text styles, fonts, colors, and languages, as well as the robustness against common image distortions. Finally, we demonstrate the practical usage of UAW with two real-world applications, followed by hyperparameter discussion.

A. Experimental Setup

STD and OCR Models: For STD, we train the bit template set targeting the CRAFT [21] STD model in a white-box setting. Transferability is evaluated on black-box STD models: DBNet [22], PSENet [19], and PAN++ [20]. For OCR systems, we assess UAW on commercial APIs: EasyOCR, WeChat OCR [24], QQ OCR, and ChatGPT-4o.

Evaluation Metrics: For scene text detection task, the *recall* (R) and *precision* (P) are two metrics for evaluating detection accuracy. The recall is defined as the ratio of correctly detected text regions to all ground-truth text regions, while precision is defined as the ratio of correctly detected text regions to all predicted text regions. They are defined as

$$R = \frac{\text{TP}}{\text{TP} + \text{FN}}, P = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{18}$$

where TP (True Positive) denotes the area of correctly detected text regions, FN (False Negative) represents the area of ground-truth text regions that are missed by the detector, and FP (False Positive) indicates the area of background regions that are erroneously classified as text. Furthermore, we calculate the F_1-score, which is the harmonic mean of precision and recall, *i.e.,* $F_1 = 2PR/(P + R)$.

However, we shall highlight that merely reporting the post-attack recall, precision, and F_1-score may misrepresent adversarial effectiveness. This is because identical post-attack metrics can correspond to different degrees of performance degradation depending on the original accuracy of STD models. To quantify such differential impact across models, we recommend using the relative metrics,

$$R^r = \frac{R^d}{R^c}, \; P^r = \frac{P^d}{P^c}, \; F_1^r = \frac{F_1^d}{F_1^c}, \tag{19}$$

where $*^c$ and $*^d$ denote the metric values *before* and *after* applying the adversarial attacks, respectively.

To evaluate watermark robustness, watermark bit accuracy (Bit Acc) is used, which is defined as follows,

$$\text{Bit Acc}\left(\mathbf{w}, \widehat{\mathbf{w}}\right) = 1 - \frac{\sum_{r,c} \mathbf{w}_{r,c} \oplus \widehat{\mathbf{w}}_{r,c}}{\texttt{numel}(\mathbf{w})}, \tag{20}$$

where $\mathbf{w}$ and $\widehat{\mathbf{w}}$ denote the original and extracted binary watermark matrix, respectively. $\oplus$ indicates XOR operation. $\texttt{numel}(\cdot)$ returns the cardinality (number of elements) of its input. Unless specified for a particular image, all metrics in the experiments are computed over 100 test images.

To evaluate the perturbation introduced by the adversarial underpainting, we adopt the Mean Underpainting Intensity (MUI) metric [6]. The MUI for an underpainting generated from the bit template set $\mathcal{B}^*$ is computed as

$$\text{MUI} = \frac{1}{2s^2}\left(\left\|\mathbf{1} - \mathbf{B}^0\right\|_1 + \left\|\mathbf{1} - \mathbf{B}^1\right\|_1\right). \tag{21}$$

Clearly, a lower MUI indicates better text readability.

Dataset: We construct a text image dataset containing various font type, font size, and font color. The general workflow for synthesizing a text image is as follows. First, the height and width are randomly selected in the range of $[360, 720]$ in training and are set as 960×960 in testing. The image background is initialized as all white. Then, a piece of text is sampled from the Brown Corpus [47]. The font types are randomly drawn from the system font library, and font sizes are randomly selected in the range of $[15, 45]$ pixels, covering typical text sizes in practice. Considering that black is dominant text color in real-world text images, the font color

TABLE II
QUANTITATIVE COMPARISON OF DIFFERENT TEXT IMAGE PROTECTION METHODS. THE SUPERSCRIPT ($*^r$) DENOTES THE RATIO OF THE POST-ATTACK RECALL, PRECISION, AND F_1-SCORE TO THE CORRESPONDING PRE-ATTACK METRIC.

Method	Adversarial Performance against STD (recall ratio $R^r \downarrow$ / precision ratio $P^r \downarrow$ / F_1-score ratio $F_1^r \downarrow$)					Watermark Recovery
	CRAFT	DBNet	EasyOCR	PAN++	PSENet	
CSR [16]	/	/	/	/	/	99.74%
UDUP [6]	0.0001/0.1667/0.0001	0.7778/0.2452/0.3729	0.7461/0.7533/0.7497	0.4442/0.7501/0.5580	0.6173/0.8448/0.7133	/
FAWA [7]	0.9380/0.9305/0.9342	0.9744/0.9690/0.9717	0.7120/0.7106/0.7113	0.9079/0.8782/0.8928	0.9761/0.9574/0.9667	100.00%
UAW (Ours)	0.0012/0.2727/0.0023	0.7666/0.2814/0.4117	0.6763/0.7293/0.7018	0.5652/0.8845/0.6897	0.6015/0.8630/0.7089	100.00%

is set to black with an 80% probability, and to a randomly selected color with a 20% probability.

Hyperparameter Setting: Unless otherwise stated, the watermark matrix composed of $a \times b$ bits is configured as 8×8. In (7), the step size is set to $\alpha = 3/255$, with a batch size of 100 samples per iteration. The momentum decay coefficient is $\mu = 0.1$, and the maximum perturbation magnitude is constrained by $\epsilon = 100/255$. The target STD model used for adversarial example generation is CRAFT. The bit template size is fixed at 30×30. Unless otherwise stated, the MUI is set to 0.09. The loss balancing factors λ_m and λ_d are set to 0.01 and $1e{-}5$, respectively. The experimental setup employed comprises an NVIDIA GeForce RTX 3090 GPU and an Intel Core i9 12900 CPU. For screen-shooting, we display the text images on an ASUS VE248H monitor. We then recapture them using a Redmi Note 11T Pro smartphone under office lighting. For print-shooting, we print the text images on A4 paper using a Brother MFC-T4500DW printer and recapture them with the same smartphone.

Fig. 12. **Text detection and watermark extraction results with different colors.** Adversarial watermark is applied using 30×30 bit templates. Watermark extraction achieved 100% bit accuracy across six text colors, averaged over 100 test images per color. These results confirm that UAW maintains both adversarial and watermarking universality with respect to character color.

B. Qualitative Comparison in Text Image Protection

As shown in Table I, we provide the qualitative comparison among the related image protection methods across three dimensions, *i.e.,* functionality, universality, and robustness. All the related methods are generally categorized into five groups: (1) Traditional text image watermarking methods [42]–[44], (2) Template-based watermarking methods [9], [16], [28], [29], (3) Adversarial attacks against OCR [5], [25], [45], [46], (4) Universal adversarial attacks against OCR [6], [26], and (5) Adversarial watermark attack on OCR (*i.e.,* FAWA) [7].

First, from the perspective of functionality, traditional text image watermarking and template-based watermarking methods provide watermarking capabilities but fail to resist OCR. In contrast, adversarial attacks against OCR and universal adversarial attacks against OCR can effectively thwart OCR systems, but they do not offer watermarking functionality. The work FAWA combines the adversarial perturbations and visible watermarking to achieve dual functionality.

Second, in terms of universality, we consider two properties, where *plug-and-play* refers to whether the method is suitable for arbitrary characters (different font, style, color *etc.*), and *arbitrary size* refers to whether the method is valid for different text image sizes. As can be seen, traditional text image watermarking methods require generating unique watermark per image, but they can handle different image sizes. Template-based watermarking exhibit better universality, accommodating various image sizes. Adversarial attacks against OCR generate perturbations that lack universality. Although the method [26] generates adversarial perturbations that can transfer across different STD models, it lacks adaptability to various text images and image sizes. The work FAWA, despite supporting different image sizes, requires the generation of new visible adversarial watermark for each image.

Third, for the robustness, the template-based watermarking-generated perturbations remain effective under typical resizing and compression distortions. On the other hand, many adversarial methods struggle with maintaining robustness under such degradation conditions.

In conclusion, each category of methods has its own merits. However, the proposed UAW method is the *only* one that achieves all desired functionalities. UAW enforces adversarial perturbations resisting OCR onto text images, with the watermark functionality-supporting meantime, ensuring transferability across different text images and image sizes. Moreover, the proposed UAW maintains adversarial effectiveness and watermark extraction accuracy under certain image distortions.

C. Quantitative Comparison in Text Image Protection

To offer a comprehensive evaluation of text image protection schemes, we conducted a quantitative analysis under a unified experimental setting. First, regarding functional integration,

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2026.3695161

Fig. 13. **UAW applied to web pages in different languages.** Despite being trained only on English, UAW remains effective on Chinese, Japanese and Arabic. Watermark bit accuracy remains at 100% across all languages.

the proposed UAW method demonstrates versatile functionality and protection effectiveness. As shown in Fig. 10, the typical template-based watermarking CSR [16] offers traceability but no adversarial protection. The exemplar adversarial patch-based attacking method UDUP [6] offers adversarial capability but no traceability. FAWA [7] uses visible textual watermarks, while the proposed UAW employs semantically invisible underpaintings, making the watermark visually imperceptible while maintaining high watermark extraction accuracy.

Second, regarding quantitative performance, the proposed UAW achieves desired watermark bit accuracy (reaching 100% in the absence of attacks) and demonstrates the unique ability to balance watermark traceability with adversarial defense. As illustrated in Table II, the watermarking method CSR yields high extraction accuracy but offers no resistance to STD. While the adversarial method UDUP degrades text detection performance, it lacks the capability to embed forensic information. A critical comparison with FAWA reveals that UAW provides stronger adversarial protection. The F_1-score ratios for FAWA are generally above 0.7, suggesting it has a negligible impact on text detection. Conversely, UAW suppresses these metrics to levels comparable to UDUP (F_1-score ratio < 0.7) across multiple detectors.

D. Universality Evaluation of the Proposed Method

UAW is devised to be universal across a variety of scenarios. To validate this, we test the performance of the proposed method under different conditions.

Character Size: In real-world scenarios, font size in text images often varies due to formatting, stylistic choices or contextual differences such as headings and body text. We test font sizes of tiny (15px), normal (30px), and large (45px) to evaluate CRAFT text detection performance across different bit template sizes. As demonstrated in Fig. 11, for a specific bit template size, the detection results are similar for all three different font sizes. When the bit template size is 10×10, the protection is insufficient as almost all the text can be correctly detected. When the bit template size is larger than 20×20, almost no text can be identified, showing strong adversarial effectiveness across all font sizes. In addition to adversarial performance, UAW also preserves watermark extraction reliability for different font sizes. The experiment using 30×30 bit templates shows that UAW achieves 100% bit accuracy across all font sizes.

TABLE III
EXPERIMENTAL RESULTS FOR SIX BACKGROUND CATEGORIES

Case	Background Type	Example	$R^r \downarrow / P^r \downarrow / F_1^r \downarrow$	Bit Acc. ↑
C1	Solid color	Light gray	0.0503/0.4998/0.0913	100.00%
C2	Simple texture	Low-frequency	0.0631/0.5682/0.1136	100.00%
C3	Weak complex	Particle	0.1325/0.6696/0.2213	100.00%
C4	Weak textual WM	High-transparency WM	0.1113/0.6446/0.1895	100.00%
C5	Strong textual WM	Low-transparency WM	0.1108/0.6374/0.1891	100.00%
C6	High complex	Texture superposition	0.1487/0.6863/0.2444	100.00%

Character Color: While black dominates text imagery (*e.g.*, web pages, documents), other colors occasionally serve emphasis or stylistic purposes. To verify the color agnosticism of UAW, we apply 30×30 bit templates to text in six colors: black, blue, gray, green, red, and yellow. As shown in Fig. 12, we conduct both adversarial and watermarking assessments. For watermarking, UAW achieves 100% bit accuracy for all text colors. In terms of adversarial effectiveness, near-zero text regions are detected by the STD across all colors. This result stems from the multi-intermediate layer loss, which facilitates the decoupling of the adversarial effect from textual content.

Language: Fig. 13 shows the effectiveness of UAW on multilingual web pages. UAW maintains strong adversarial performance across the evaluated languages, indicating good cross-language transferability. For watermarking evaluation, UAW achieves 100% bit accuracy in the four cases, demonstrating the robustness of watermark extraction across the evaluated languages. Even with complex layouts, watermark messages remain fully recoverable. These results indicate that the proposed method maintains stable performance across the evaluated languages and layouts. Since different languages have similar high-contrast and elongated structures, periodic template patterns in the background disrupt local features in a similar way. These results suggest that the proposed method has good cross-language transferability.

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2026.3695161

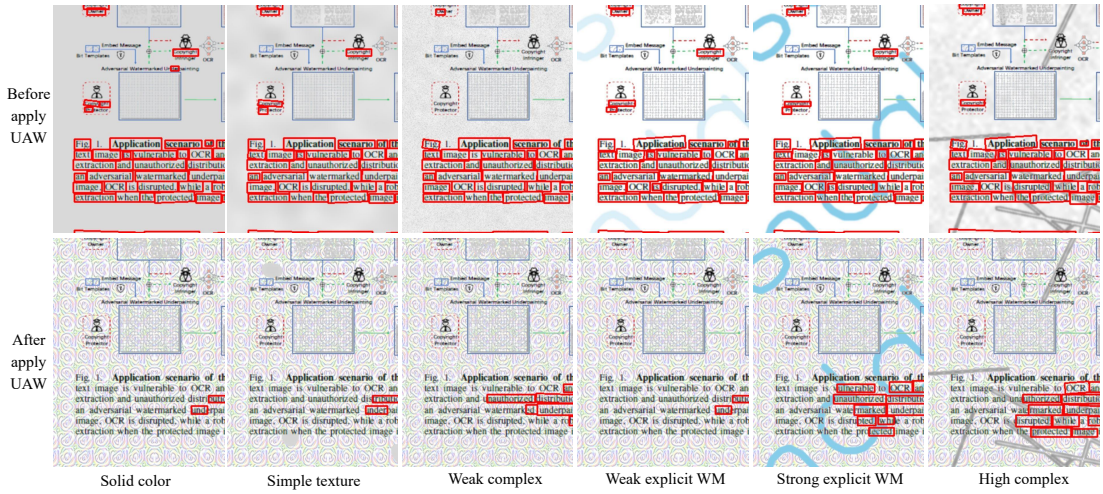


Fig. 14. **Representative examples of six categories of background complexity backgrounds.** The first row shows the text image detection results before applying UAW, and the second row shows the results after applying UAW. Each column represents a different background texture.

TABLE IV
ROBUSTNESS EVALUATION OF ADVERSARIAL PROPERTY UNDER COMMON DISTORTIONS

Distortion	Brightness		Contrast		Saturation		Hue		Gaussian Noise		Gaussian Blur		JPEG Compression			Resize	
	0.85	1.15	0.85	1.15	0.85	1.15	-0.1	0.1	0.05	0.1	0.5	1.0	90	70	50	(0.8,0.9)	(1.5,1.4)
$R^r \downarrow$	0.0023	0.1658	0.0037	0.0021	0.0045	0.0027	0.0095	0.0045	0.0117	0.2051	0.0099	0.5216	0.0064	0.0134	0.0337	0.0177	0.1261
$P^r \downarrow$	0.3429	0.5780	0.4524	0.3929	0.4035	0.5600	0.5213	0.4600	0.4317	0.6327	0.4513	0.6840	0.4459	0.4759	0.5406	0.5714	0.4837
$F_1^r \downarrow$	0.0046	0.2577	0.0073	0.0043	0.0089	0.0054	0.0187	0.0089	0.0227	0.3098	0.0194	0.5919	0.0127	0.0261	0.0634	0.0342	0.2001

Complexity of Background: We conducted experiments with complex backgrounds to evaluate both OCR resistance and watermark extraction. Six exemplar background categories were presented. They included solid color, simple texture, weak complex texture, weak textual watermark, strong textual watermark, and highly complex texture. The visualization of representative background examples are summarized in Fig. 14. The quantitative results are reported in Table III. The results show that the proposed method remains effective under complex backgrounds. Bit accuracy stays at 100% for all six categories, including the highly complex case C6. Adversarial performance becomes weaker as background complexity increases (F_1-score ratio=0.2444). This trend is expected because dense textures introduce additional edges and repeated patterns. These patterns partially dilute the adversarial effect and also reduce the effective background area for embedding. Even under strong textual watermark or highly-complex settings, the proposed UAW method maintains reliable watermark decoding and still degrades text detection.

E. Robustness Evaluation of the Proposed Method

We evaluate the robustness of UAW in terms of both adversarial effectiveness and watermark reliability. The evaluation covers common image processing operations.

Adversarial Robustness: The results in Table IV demonstrate that UAW exhibits high adversarial robustness across a variety of common image distortions. In particular, it resists color transformations including changes in brightness, contrast, saturation, and hue, with all recall ratios R^r below 0.2. The addition of Gaussian noise also has limited impact on adversarial performance, with recall ratio remaining low even at a noise level of 0.1. However, high-intensity Gaussian blur degrades adversarial robustness, with the recall ratio increasing to 0.5216. This is because blurring suppresses fine-grained image details crucial to adversarial features. Nevertheless, many text regions still remain undetected. JPEG compression and resizing introduce moderate challenges, with R^r values ranging from 0.0064 to 0.1261, indicating that UAW remains effective even under compression artifacts and scale changes. These results confirm the resilience of UAW to practical image degradation scenarios.

Watermarking Robustness: The results are given in Table V. UAW achieves high robustness in watermark extraction under a wide range of common image distortions. For all tested categories, including color transformations, noise, blur and geometric operations, the watermark bit accuracy remains 100%, except under heavy blurring. Specifically, for color transformations, the bit accuracy reaches 100% at all intensity levels. This confirms the stability of UAW in text images where color properties may shift due to image enhancement filters or format conversions. In terms of blur and noise, UAW maintains 100% bit accuracy for Gaussian blur levels up to 1.0, and all tested Gaussian noise levels. For an extreme case when the blur level is 1.5, the bit accuracy drops to 85.42%. However, such heavy blurring is quite destructive for text images, which rarely happens in real-world scenarios. Under JPEG compression, even at aggressively reduced quality settings, the bit accuracy remains to be 100%, showing good compression robustness.

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2026.3695161

Fig. 15. **Text detection results on real-world certificates.** Top: Most text regions are correctly detected on original text images. Bottom: After applying UAW, almost no text is detected, while the watermark is recovered and visual quality remains intact.

TABLE V
ROBUSTNESS EVALUATION OF WATERMARK UNDER DIFFERENT DISTORTIONS

Distortion	Brightness			Contrast		
	0.7	0.85	1.3	0.7	0.85	1.3
Bit Acc. ↑	100%	100%	100%	100%	100%	100%
Distortion	Saturation			Hue		
	0.7	0.85	1.3	-0.2	-0.1	0.2
Bit Acc. ↑	100%	100%	100%	100%	100%	100%
Distortion	Gaussian Noise			Gaussian Blur		
	0.1	0.15	0.2	0.5	1.0	1.5
Bit Acc. ↑	100%	100%	100%	100%	100%	85.42%
Distortion	JPEG Compression			Resize		
	90	50	10	0.6*0.7	0.8*0.9	1.5*1.4
Bit Acc. ↑	100%	100%	100%	100%	100%	100%
Distortion	Rotate			Crop		
	90	180	270	0.6	0.7	0.8
Bit Acc. ↑	100%	100%	100%	100%	100%	100%

Finally, geometric distortions such as resizing, cropping, and rotations do not affect watermark extraction performance. The bit accuracy remains at 100% under all such transformations, confirming the spatial invariance and adaptability of UAW.

F. Evaluation on Real-world Applications

To further verify the real-world effectiveness of our method, we conducted experiments on the practical utility of UAW.

Privacy Protection for Sensitive Certificate Images: Certificates often contain sensitive or legally binding information (*e.g.*, identity cards, award certificates), making them vulnerable to unauthorized duplication. Fig. 15 demonstrates the protection efficacy of UAW. While CRAFT detects most text in original certificates images (top row), the protected versions

Fig. 16. **Text detection results on black-box models and practical applications.** The top row shows detection outputs from five representative STD models. The bottom row presents OCR results from real-world applications. All systems exhibit detection errors after applying UAW, demonstrating its strong transferability in both open-source and practical OCR scenarios.

TABLE VI
ADVERSARIAL EXPERIMENTAL RESULTS FOR DIFFERENT STD TYPES WITH UAW EMBEDDING

STD Category	STD Model	$R^r \downarrow$	$P^r \downarrow$	$F_1^r \downarrow$
Pixel-level Classification-based	CRAFT [21]	0.0012	0.2727	0.0023
	DBNet [22]	0.7666	0.2814	0.4117
	EasyOCR	0.6763	0.7293	0.7018
	PAN++ [20]	0.5652	0.8845	0.6897
	PSENet [19]	0.6015	0.8630	0.7089
Geometry-based	EAST [17]	0.6368	0.3874	0.4817
	TextBoxes++ [18]	0.7050	0.8470	0.7695
VLM-based	TCM [48]	0.4895	0.2978	0.3703

(bottom row) prevent nearly all text detection, confirming strong resistance to content extraction. Beyond adversarial effectiveness, UAW guarantees watermark robustness. The watermarks are extracted with 100% accuracy from all protected certificates. Simultaneously, the method preserves document legibility and maintains visual integrity.

Transferability to Unknown OCR Systems: In real-world scenarios, it is often difficult to predict which OCR system may be employed by unauthorized users. Therefore, UAW is designed to exhibit transferability, meaning that it remains effective against unknown or black-box STD models. We evaluate other open-source STD systems and report the results in Table VI. We have observed two distinct failure modes of OCR system. The first mode involves a reduction in recall ratio where the model fails to detect existing text. The proposed UAW triggers this behavior in models such as EasyOCR, PAN++, and PSENet. The second mode involves a reduction in precision ratio where the model incorrectly identifies background as text. This behavior is more prominent in DBNet. The F_1-score represents the harmonic mean of precision and recall. Consequently, both of these modes lead to a low F_1-score ratio.

Geometry-based detectors regress oriented bounding boxes, which differs from pixel level methods that rely on post processing over classification maps. EAST makes dense predictions over the whole image, so it is disturbed when local contrast and textures change. Our periodic underpainting alters background statistics and adds text-like patterns, which breaks

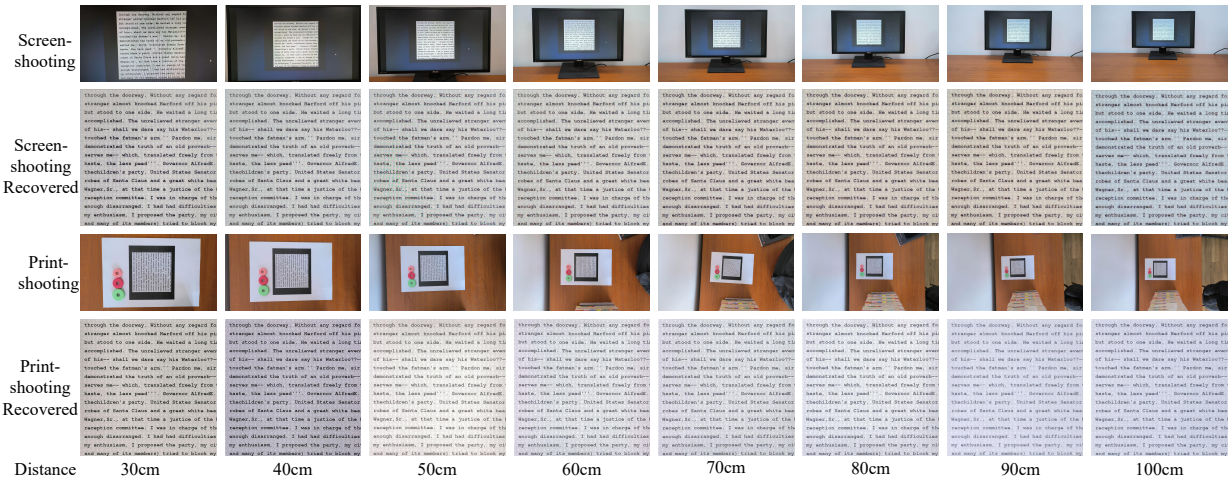


Fig. 17. **The example of shooting documents with different distances.** The camera is placed facing the target. The shooting distance is set from 30 to 100 cm under both screen-shooting and print-shooting scenarios.

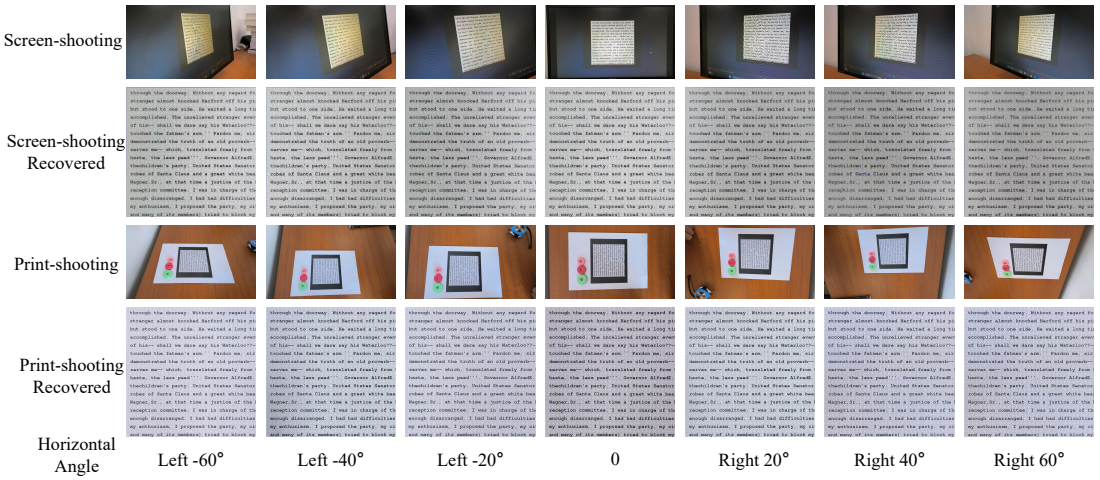


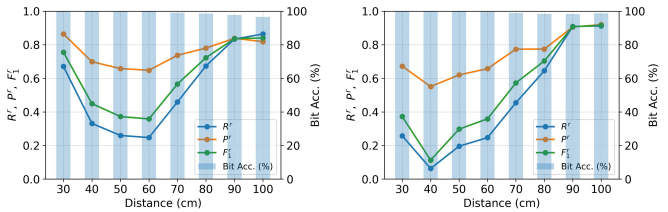
Fig. 18. **The example of shooting documents with different horizontal angles.** The shooting distance is fixed at 40 cm. The camera is rotated horizontally to simulate oblique views, with angles of $+20°$, $+40°$, $+60°$, 0, $-20°$, $-40°$, and $-60°$. The plus sign denotes a rightward rotation, while the minus sign denotes a leftward rotation.

its score maps and box regression and leads to more misses and shifted boxes. TextBoxes++ relies on anchor boxes and strong features inside candidate regions, so it depends less on small background changes. Because our fusion keeps the foreground text intact, it still sees stable cues and keeps a high F_1-score ratio. Despite this inherent resistance, UAW could also degrade the detection quality to a certain extent. TCM uses CLIP features with prompt alignment and cross-modal similarity, so it can be confused when patch-level representations are distorted. Our periodic textures interfere with these representations, weaken the similarity response, and pull attention away from text regions.

In addition to open-source models, we also test UAW on the black-box commercial OCR component that integrated in real-world applications, including WeChat and QQ, as well as the latest multimodal foundation model ChatGPT-4o. As shown in Fig. 16, these three systems fail to detect or correctly locate text in adversarial watermarked text images. This demonstrates that UAW maintains strong adversarial property across both traditional OCR systems and the cutting-edge AI models, ensuring robust protection in practical scenarios.

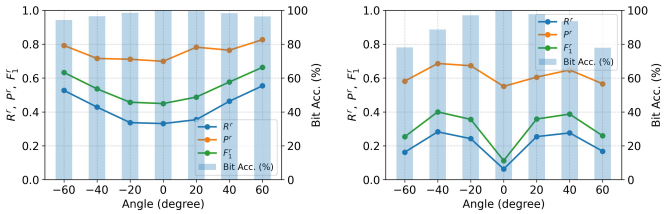
Robustness Tests under Shooting Scenarios: For the shooting experiments, we increase the MUI from 0.09 to 0.12. The reason is that print-shooting and screen-shooting introduce additional distortions, such as capture blur, display or printing deviation, and resolution loss, which weaken the adversarial underpainting. A slightly larger MUI therefore helps preserve OCR resistance and watermark recovery while maintaining acceptable readability. Except for the MUI adjustment and the grayscale constraint adopted for physical shooting scenarios, the other experimental settings remain unchanged. As shown in Fig. 17, for the distance study, the camera is placed facing the target. The shooting distance is set from 30 to 100 cm under both screen-shooting and print-shooting scenarios. As shown in Fig. 18, for the angle study, the shooting distance is fixed at 40 cm. The camera is rotated horizontally to simulate oblique views.

As shooting distance affects image sharpness and pixel



(a) Screen-shoot (b) Print-shoot

Fig. 19. **Adversarial performance and bit accuracy at different distances.** Shooting distance affects image sharpness and pixel density.



(a) Screen-shoot (b) Print-shoot

Fig. 20. **Adversarial performance and bit accuracy at different horizontal angles.** A larger shooting angle introduces projective distortion and non-uniform blur.

density, at short distances ($\leq$ 30cm), camera defocus effects and optical blur could reduce adversarial effectiveness (F_1-score ratio $>$ 0.4). As shown in Fig. 19, at long distances ($\geq$ 70cm), the resolution loss in the document region weakens background texture patterns, which degrades attack strength (F_1-score ratio $>$ 0.4), but watermark decoding reliability remains (bit accuracy $> 95\%$). Within a practical range where the text remains legible, UAW continues to suppress text detection while preserving reliable watermark extraction in both print-shooting and screen-shooting scenarios.

A larger shooting angle introduces projective distortion and non-uniform blur. As shown in Fig. 20, near frontal views ($\leq$ 20°) produce stable results in both adversarial performance and bit accuracy (F_1-score ratio $<$ 0.4 and bit accuracy $> 95\%$). When the viewing angle increases (40°), both metrics gradually deteriorate due to stronger geometric deformation. Under extreme oblique views($\geq$ 60°), for the screen-shooting scenario, substantial angular deviations lead to minor deterioration in both metrics (F_1-score ratio $<$ 0.7 and bit accuracy $> 90\%$). For the print-shooting scenario, where watermark extraction suffers the most severe impact (bit accuracy $< 80\%$), the detection performance of the STD model degrades more rapidly than the adversarial effect, thereby preserving strong adversarial effectiveness (F_1-score ratio $<$ 0.3).

Time Consumption for Real-time Requirements: The training of bit templates constitutes a one-time offline cost completed during the bit templates generation phase. The online embedding phase involves bit-template mapping, flip-based unit construction, tiling for underpainting generation (Pattern Gen), and fusion with text image (Fusion). The extraction phase consists of watermark block synchronization (Sync), watermark block state determination (Decide), and watermark message extraction (Get WM). As shown in Table VII, offline template training takes 347.89 seconds on the GPU. This cost is incurred only once during the bit template generation stage. The optimized templates are then reused for arbitrary text images, so the offline overhead does not affect deployment latency.

TABLE VII
TIME CONSUMPTION OF THE PROPOSED METHOD (SECONDS). TEMPLATE TRAINING IS PERFORMED ONCE OFFLINE TO GENERATE BIT TEMPLATES AND CAN BE REUSED FOR ARBITRARY TEXT IMAGES. WATERMARK MODULATION AND EXTRACTION ARE EXECUTED ONLINE.

Processing Unit	**Offline** Template Training	**Online** WM Modulation		**Online** WM Extraction		
		Pattern	Fusion	Sync	Decide	Get WM
GPU	347.89	0.0018	0.0884	0.3881	0.0787	0.4942
CPU	/	0.0131	0.1407	1.7863	0.0775	0.1754

TABLE VIII
EFFECT OF MUI AND BIT TEMPLATE SIZE ON F_1-SCORE RATIO $F_1^r \downarrow$ AGAINST CRAFT

Patch Size	MUI						
	0.06	0.07	0.08	0.09	0.10	0.11	0.12
10	0.7573	0.7319	0.7124	0.6610	0.6007	0.5471	0.4563
20	0.9040	0.8625	0.7907	0.5224	0.0699	0.0064	0.0007
30	0.1235	0.0258	0.0104	0.0023	0.0012	0.0004	0.0004
40	0.8771	0.7474	0.4023	0.1066	0.0108	0.0006	0.0000
50	0.7588	0.4805	0.0968	0.0028	0.0019	0.0000	0.0000

TABLE IX
EFFECT OF MUI AND BIT TEMPLATE SIZE ON BIT ACCURACY ↑

Patch Size	MUI						
	0.06	0.07	0.08	0.09	0.10	0.11	0.12
10	89.84%	97.45%	97.47%	100.00%	100.00%	100.00%	100.00%
20	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
30	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
40	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
50	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%

Online watermark modulation is lightweight. The total runtime is 0.0902 seconds on the GPU and 0.1538 seconds on the CPU. This efficiency indicates that embedding introduces negligible delay in the text image rendering and publishing pipeline. Online watermark extraction is more computationally intensive. The total runtime is 1.4686 seconds on the GPU and 2.2256 seconds on the CPU. Synchronization is the dominant cost on the CPU and takes 1.7863 seconds. GPU acceleration reduces synchronization to 0.3881 seconds by speeding up correlation computations via batch processing. Overall, the measured latency remains practical for real-world use, and GPU support further improves throughput.

G. Discussion on the Hyperparameter Settings

MUI and Size of Bit Template for Adversarial Effectiveness: Table VIII shows the effect of different bit template sizes and MUI values on the adversarial performance against the CRAFT model. For each template size, increasing MUI value leads to a lower F_1-score, indicating stronger adversarial effects. However, a higher MUI tends to incur more noticeable perturbations, which degrades text readability. For a fixed MUI

TABLE X
EFFECT OF HYPERPARAMETER λ_m IN LOSS FUNCTION ON F_1-SCORE RATIO $F_1^r \downarrow$ AGAINST CRAFT (MUI= 0.09)

Patch Size	λ_m					
	0	0.001	0.01	0.1	1	10
10	0.6651	0.7325	0.6744	0.6857	0.7123	**0.6610**
20	0.8925	0.7937	0.8515	0.6643	**0.5224**	0.7640
30	0.0267	0.0276	**0.0023**	0.0138	0.3248	0.0389
40	0.4080	**0.1066**	0.3794	0.3675	0.2071	0.8344
50	0.2413	0.0833	**0.0029**	0.0057	0.0732	0.5904

TABLE XI
ABLATION STUDY OF LOSS FUNCTIONS

$\mathcal{L}_{adv}/\mathcal{L}_m/\mathcal{L}_d$	CRAFT [21]			PAN++ [20]			Bit Acc. ↑
	$R^r \downarrow$	$P^r \downarrow$	$F_1^r \downarrow$	$R^r \downarrow$	$P^r \downarrow$	$F_1^r \downarrow$	
✓ / ✗ / ✗	0.0138	0.4104	0.0267	0.6882	0.9363	0.7933	100.00%
✓ / ✓ / ✗	0.0012	0.2727	0.0023	0.5652	0.8845	0.6897	100.00%
✓ / ✓ / ✓	0.0037	0.4750	0.0073	0.5849	0.9149	0.7136	100.00%

value, the adversarial effectiveness would be enhanced at first and then declines with the increment of template size. This phenomenon can be attributed to the fact that smaller templates lack sufficient capacity to form adversarial patterns, while excessively large templates will introduce complexity that impedes training bit templates. In experiments, the observed balancing point of bit template size is at 30. To balance visual quality and adversarial robustness without harming watermark extraction, we recommend setting the template size as 30, and MUI value as 0.09.

MUI and Size of Bit Template for Watermark Reliability: We assess watermark extraction accuracy under different MUI values and bit template sizes. Table IX shows that nearly all the parameter combinations achieve 100% accuracy and the performance drop is observed for small sizes (< 20) with lower MUI values (< 0.09). This suggests that the watermarking remains robust and reliable across various perturbation intensities and bit template sizes. The extraction process remains unaffected by adversarial perturbations, highlighting the functional compatibility between adversarial example generation and watermarking in the proposed framework.

Hyperparameter λ_m: Table X illustrates the effect of λ_m in loss function on the adversarial effectiveness. The parameter λ_m balances between adversarial loss and multi-intermediate layer loss during optimization. The results show that the optimal λ_m differs across template sizes, with the 30×30 size achieving the best performance at $\lambda_m = 0.01$. It is worth noting that, compared to the case that uses no multi-intermediate layer loss ($\lambda_m = 0$), the optimal setting of λ_m improves adversarial effectiveness, demonstrating its complementary role to the adversarial loss.

Ablation Study of Loss Functions: We evaluate the impact of $\mathcal{L}_m$ and $\mathcal{L}_d$ on performance in both white-box and black-box settings. As shown in Table XI, the inclusion of $\mathcal{L}_m$ improves the overall adversarial effectiveness against the CRAFT and PAN++ models, and also contributes to enhancing the transferability of adversarial perturbations. The presence

TABLE XII
IMPACT OF ADVERSARIAL AND WATERMARKING INTERACTION

Method	CRAFT [21]			PAN++ [20]			Bit Acc. ↑
	$R^r \downarrow$	$P^r \downarrow$	$F_1^r \downarrow$	$R^r \downarrow$	$P^r \downarrow$	$F_1^r \downarrow$	
Only Adversarial	0.0014	0.3043	0.0027	0.5642	0.8858	0.6893	/
Only Watermarking	0.9058	0.9533	0.9289	0.9776	0.9842	0.9809	100.00%
Ours	0.0012	0.2727	0.0023	0.5652	0.8845	0.6897	100.00%

TABLE XIII
COMPARISON OF ADVERSARIAL PERFORMANCE AND WATERMARK BIT ACCURACY UNDER DIFFERENT PAYLOAD SETTINGS

Payload (bits)	Matrix Size ($h \times w$)	$R^r \downarrow / P^r \downarrow / F_1^r \downarrow$	Bit Acc. ↑
64	8×8	0.0012/0.2727/0.0023	100.00%
81	9×9	0.0039/0.3721/0.0078	100.00%
100	10×10	0.0020/0.3368/0.0039	100.00%
128	8×16	0.0040/0.3869/0.0079	100.00%
128	16×8	0.0047/0.3889/0.0093	100.00%

or absence of $\mathcal{L}_d$ has a minimal impact on adversarial performance, with negligible differences observed between these conditions. Both configurations ensure the reliable extraction of the watermark. However, to guarantee clear distinguishability between the two bit templates, the discriminative loss has been incorporated into the framework.

Impact of Adversarial and Watermarking Interaction: We investigate the effects of embedding only the watermark and adding only adversarial perturbations to analyze the interaction between adversarial perturbations and watermark signals. In the experiment, the underpainting was optimized to an MUI of 0.09, with CRAFT in the white-box setting and PAN++ in the black-box setting. The bit templates used for watermarking in the *Only Watermarking* case were randomly generated to follow a uniform distribution. As shown in Table XII, the adversarial effects of the *Only Adversarial* method and the proposed method are similar in both the white-box and black-box settings, indicating that the watermarking module has minimal impact on the adversarial effectiveness.

On the other hand, both the *Only Watermarking* and the proposed methods achieve 100% watermark bit accuracy, showing that the adversarial module has little effect on watermarking performance. Furthermore, from the perspective of adversarial effectiveness, it is clear that if the underpainting is not carefully designed, it fails to achieve a meaningful adversarial effect, thus highlighting the importance of the adversarial module.

Watermark Matrix Size: To conduct experiments to evaluate whether the proposed method can support more payload, we increased the watermark matrix size from 8×8 to larger layouts. We kept the bit template set and the fusion strategy unchanged. Payload extension changes the arrangement and count of bit blocks, but it does not increase perturbation amplitude. As shown in Table XIII, bit accuracy remains 100% for all payload settings. Adversarial property remains with payload length changes (F_1-score ratio < 0.1). Capacity scaling only changes the dimensions of the binary watermark matrix. The pipeline does not impose a structural constraint on the watermark matrix dimensions. The main trade-off is spatial

redundancy. Given an image size, a larger payload reduces the number of repeated watermark blocks produced by tiling. This reduction can weaken majority voting and may lower extraction stability under stronger distortions.

V. Conclusion

In this work, we propose Universal Adversarial Watermarking (UAW), a dual-protection framework to paralyze text detection and implement watermarking simultaneously. The proposed method first trains two bit templates to represent watermark bits and then constructs the adversarial watermarked underpainting. The underpainting is readily tiled as background texture to text images without tedious per-image customization. Experiments demonstrate the practical utility of UAW in privacy protection and defense against real-world commercial black-box OCR systems. We believe these capabilities position UAW as a promising solution for protecting text images against unauthorized extraction and distribution.

References

[1] W. De Groef, "Client-and server-side security technologies for javascript web applications," 2016.

[2] F. W. Dingledy and A. B. Matamoros, "What is digital rights management?" *Digital Rights Management: The Librarian's Guide*, pp. 1–25, 2016.

[3] X. Zhang, Y. Chen, J. Liang, J. Zhang, T. Lu, and C. Wang, "Research on anti-crawler and anti-anti-crawler technology," in *Proc. Int. Conf. Inform., Netw. Comput. (ICINC)*. IEEE, 2022, pp. 35–39.

[4] Selenium: Web browser automation. Accessed: Apr. 1, 2026. [Online]. Available: https://www.selenium.dev/

[5] T. Xiang, H. Liu, S. Guo, H. Liu, and T. Zhang, "Text's armor: optimized local adversarial perturbation against scene text editing attacks," in *Proc. 30th ACM Int. Conf. Multimedia (ACM MM)*, 2022, pp. 2777–2785.

[6] J. Deng, L. Dong, J. Chen, D. Yan, R. Wang, D. Ye, L. Zhao, and J. Tian, "Universal defensive underpainting patch: Making your text invisible to optical character recognition," in *Proc. 31st ACM Int. Conf. Multimedia (ACM MM)*, 2023, pp. 7559–7568.

[7] L. Chen, J. Sun, and W. Xu, "FAWA: Fast adversarial watermark attack on optical character recognition (OCR) systems," in *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2020, Ghent, Belgium, September 14–18, 2020, Proceedings, Part III*. Springer, 2021, pp. 547–563.

[8] M. Tancik, B. Mildenhall, and R. Ng, "StegaStamp: Invisible hyperlinks in physical photographs," in *Proc. IEEE/CVF Conf. Computer Vis. Pattern Recognit. (CVPR)*, 2020, pp. 2117–2126.

[9] H. Fang, W. Zhang, H. Zhou, H. Cui, and N. Yu, "Screen-shooting resilient watermarking," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 6, pp. 1403–1418, 2018.

[10] "xSecuritas: Enterprise screen watermark solution," accessed: Apr. 1, 2026. [Online]. Available: https://www.xsecuritas.com/

[11] "Fasoo smart screen: Screen watermark and capture prevention," accessed: Apr. 1, 2026. [Online]. Available: https://en.fasoo.com/products/fasoo-smart-screen/

[12] Huawei Cloud, "Document watermarking – data security center," accessed: Apr. 1, 2026. [Online]. Available: https://support.huawei.com/enterprise/en/doc/EDOC1100403716/c946d81e/document-watermarking

[13] E. Quiring, D. Arp, and K. Rieck, "Forgotten siblings: Unifying attacks on machine learning and digital watermarking," in *Proc. IEEE Eur. Symp. Security Privacy (EuroS&P)*. IEEE, 2018, pp. 488–502.

[14] X. Jia, X. Wei, X. Cao, and X. Han, "Adv-watermark: A novel watermark perturbation for adversarial examples," in *Proc. 28th ACM Int. Conf. Multimedia (ACM MM)*, 2020, pp. 1579–1587.

[15] S. Feng, F. Feng, X. Xu, Z. Wang, Y. Hu, and L. Xie, "Digital watermark perturbation for adversarial examples to fool deep neural networks," in *Proc. Int. Jt. Conf. Neural Netw. (IJCNN)*. IEEE, 2021, pp. 1–8.

[16] H. Fang, W. Zhang, Z. Ma, H. Zhou, S. Sun, H. Cui, and N. Yu, "A camera shooting resilient watermarking scheme for underpainting documents," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 11, pp. 4075–4089, 2019.

[17] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, and J. Liang, "EAST: An efficient and accurate scene text detector," in *Proc. IEEE Conf. Computer Vis. Pattern Recognit. (CVPR)*, 2017, pp. 5551–5560.

[18] M. Liao, B. Shi, and X. Bai, "TextBoxes++: A single-shot oriented scene text detector," *IEEE Trans. Image Process.*, vol. 27, no. 8, pp. 3676–3690, 2018.

[19] W. Wang, E. Xie, X. Li, W. Hou, T. Lu, G. Yu, and S. Shao, "Shape robust text detection with progressive scale expansion network," in *Proc. IEEE/CVF Conf. Computer Vis. Pattern Recognit. (CVPR)*, 2019, pp. 9336–9345.

[20] W. Wang, E. Xie, X. Li, X. Liu, D. Liang, Z. Yang, T. Lu, and C. Shen, "Pan++: Towards efficient and accurate end-to-end spotting of arbitrarily-shaped text," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 9, pp. 5349–5367, 2022.

[21] Y. Baek, B. Lee, D. Han, S. Yun, and H. Lee, "Character region awareness for text detection," in *Proc. IEEE/CVF Conf. Computer Vis. Pattern Recognit. (CVPR)*, 2019, pp. 9365–9374.

[22] M. Liao, Z. Wan, C. Yao, K. Chen, and X. Bai, "Real-time scene text detection with differentiable binarization," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 07, 2020, pp. 11 474–11 481.

[23] M. Liao, Z. Zou, Z. Wan, C. Yao, and X. Bai, "Real-time scene text detection with differentiable binarization and adaptive scale fusion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 1, pp. 919–931, 2022.

[24] WeChat OCR. Accessed: Apr. 1, 2026. [Online]. Available: https://cloud.tencent.com/developer/article/1798403

[25] Y. Xu, P. Dai, Z. Li, H. Wang, and X. Cao, "The best protection is attack: Fooling scene text recognition with minimal pixels," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 1580–1595, 2023.

[26] S. Wu, T. Dai, G. Meng, B. Chen, J. Lu, and S.-T. Xia, "Transferable adversarial attacks for deep scene text detection," in *Proc. 25th Int. Conf. Pattern Recognit. (ICPR)*. IEEE, 2021, pp. 8945–8951.

[27] J. Zhu, R. Kaplan, J. Johnson, and L. Fei-Fei, "HiDDeN: Hiding data with deep networks," in *Proc. Eur. Conf. Computer Vis. (ECCV)*, 2018, pp. 657–672.

[28] H. Fang, D. Chen, Q. Huang, J. Zhang, Z. Ma, W. Zhang, and N. Yu, "Deep template-based watermarking," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 4, pp. 1436–1451, 2020.

[29] Z. Ma, W. Zhang, H. Fang, X. Dong, L. Geng, and N. Yu, "Local geometric distortions resilient watermarking scheme based on symmetry," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 12, pp. 4826–4839, 2021.

[30] C. Chen, B. Zhou, and W. H. Mow, "RA code: A robust and aesthetic code for resolution-constrained applications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 11, pp. 3300–3312, 2017.

[31] C. Chen, W. Huang, L. Zhang, and W. H. Mow, "Robust and unobtrusive display-to-camera communications via blue channel embedding," *IEEE Trans. Image Process.*, vol. 28, no. 1, pp. 156–169, 2018.

[32] J. Wang, H. Wang, J. Zhang, H. Wu, X. Luo, and B. Ma, "Invisible adversarial watermarking: A novel security mechanism for enhancing copyright protection," *ACM Trans. Multimedia Comput., Commun. Appl.*, vol. 21, no. 2, pp. 1–22, 2024.

[33] Y. Zhang, D. Ye, S. Shen, C. Xie, Z. Liu, J. Deng, and L. Tang, "Double privacy guard: Robust traceable adversarial watermarking against face recognition," *CoRR*, 2024.

[34] Y. Zhang, D. Ye, C. Xie, L. Tang, X. Liao, Z. Liu, C. Chen, and J. Deng, "Dual defense: Adversarial, traceable, and invisible robust watermarking against face swapping," *IEEE Trans. Inf. Forensics Security*, 2024.

[35] S. Xu, T. Qiao, M. Xu, W. Wang, and N. Zheng, "Robust adversarial watermark defending against gan synthesization attack," *IEEE Signal Process. Lett.*, vol. 31, pp. 351–355, 2024.

[36] T. Qiao, B. Zhao, R. Shi, M. Han, M. Hassaballah, F. Retraint, and X. Luo, "Scalable universal adversarial watermark defending against facial forgery," *IEEE Trans. Inf. Forensics Security*, 2024.

[37] Q. Huang, I. Katsman, H. He, Z. Gu, S. Belongie, and S.-N. Lim, "Enhancing adversarial example transferability with an intermediate level attack," in *Proc. IEEE/CVF Int. Conf. Computer Vis. (ICCV)*, 2019, pp. 4733–4742.

[38] J. Zhang, W. Wu, J.-t. Huang, Y. Huang, W. Wang, Y. Su, and M. R. Lyu, "Improving adversarial transferability via neuron attribution-based attacks," in *Proc. IEEE/CVF Conf. Computer Vis. Pattern Recognit. (CVPR)*, 2022, pp. 14 993–15 002.

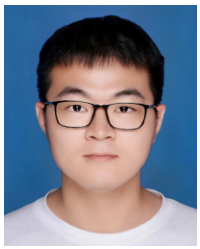
[39] Y. Zhao, H. Zhu, R. Liang, Q. Shen, S. Zhang, and K. Chen, "Seeing isn't believing: Towards more robust adversarial attack against real world object detectors," in *Proc. ACM SIGSAC Conf. Computer Commun. Security (CCS)*, 2019, pp. 1989–2004.

[40] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2018.
[41] J.-C. Yoo and T. H. Han, "Fast normalized cross-correlation," *Circuits, Syst. Signal Process.*, vol. 28, pp. 819–843, 2009.
[42] H. Yazdani, M. Doostari, and V. Yazdani, "A new method to persian text watermarking using curvaceous letters," *J. Basic Appl. Sci. Res.*, vol. 3, no. 4, 2013.
[43] S. H. Low, N. F. Maxemchuk, and A. M. Lapone, "Document identification for copyright protection using centroid detection," *IEEE Trans. Commun.*, vol. 46, no. 3, pp. 372–383, 1998.
[44] P. V. K. Borges, J. Mayer, and E. Izquierdo, "Robust and transparent color modulation for text data hiding," *IEEE Trans. Multimedia*, vol. 10, no. 8, pp. 1479–1489, 2008.
[45] C. Song and V. Shmatikov, "Fooling OCR systems with adversarial text images," *arXiv preprint arXiv:1802.05385*, 2018.
[46] X. Xu, J. Chen, J. Xiao, L. Gao, F. Shen, and H. T. Shen, "What machines see is not what they get: Fooling scene text recognition models with adversarial text images," in *Proc. IEEE/CVF Conf. Computer Vis. Pattern Recognit. (CVPR)*, 2020, pp. 12 304–12 314.
[47] W. N. Francis and H. Kucera, "Brown corpus manual," *Letters to the Editor*, vol. 5, no. 2, p. 7, 1979.
[48] W. Yu, Y. Liu, W. Hua, D. Jiang, B. Ren, and X. Bai, "Turning a clip model into a scene text detector," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 6978–6988.

Fangjun Yan received the B.Eng. degree from Wenzhou University, Wenzhou, China, in 2022. He is currently pursuing the M.E. degree with the Department of Computer Science, Faculty of Electrical Engineering and Computer Science, Ningbo University, Ningbo, China. His research interests include multimedia security and forensics.

Yutong Huang received the B.Eng. degree from Ningbo University, Ningbo, China, in 2024. She is currently pursuing the M.E. degree with the Department of Computer Science, Faculty of Electrical Engineering and Computer Science, Ningbo University, Ningbo, China. Her research interests include multimedia security and forensics.

Li Dong (Member, IEEE) received the B.Eng. degree from Chongqing University, Chongqing, China, in 2012, and the M.S. and Ph.D. degrees from the University of Macau, Macau, in 2014 and 2018, respectively. He is currently an Associate Professor with the Department of Computer Science, Faculty of Electrical Engineering and Computer Science, Ningbo University, Ningbo, China. His research interests include statistical image modeling and processing, and multimedia security and forensics.



Jiacheng Deng received the M.S. degree in computer science from Ningbo University, Ningbo, China, in 2023. He is currently pursuing the Ph.D. degree with the School of Cyber Science and Engineering, Wuhan University, Wuhan, China. His research interests include theories in synthetic speech detection, adversarial attacking, and explanation of neural networks.

Qiuping Jiang (Senior Member, IEEE) is a Full Professor with the School of Information Science and Engineering, Ningbo University, Ningbo, China. His research interests include image quality assessment, visual perception modeling, and underwater visual information processing. He is currently serving as an Associate Editor for several SCI-indexed journals such as Displays, Journal of Visual Communication and Image Representation, IET Image Processing, and Journal of Electronic Imaging.

Zhibo Wang (Senior Member, IEEE) received the B.E. degree in automation from Zhejiang University, Hangzhou, China, in 2007 and the Ph.D. degree in electrical engineering and computer science from the University of Tennessee, Knoxville, TN, USA, in 2014. He is currently a professor with the School of Cyber Science and Technology, College of Computer Science and Technology, Zhejiang University. His research interests include AI security, the Internet of Things, network security, and privacy protection.

Xin Liao (Senior Member, IEEE) received the B.E. and Ph.D. degrees in information security from Beijing University of Posts and Telecommunications in 2007 and 2012, respectively. He was a Post-Doctoral Fellow with the Institute of Software, Chinese Academy of Sciences, and a Research Associate with The University of Hong Kong. From 2016 to 2017, he was a Visiting Scholar with the University of Maryland, College Park, MD, USA. He is currently a Professor and a Ph.D. Supervisor with Hunan University, China. His current research interests include multimedia forensics, steganography, and watermarking. He is a Secretary and a member of the Technical Committee (TC) on Multimedia Security and Forensics of Asia Pacific Signal and Information Processing Association, a member of TC on Computer Forensics of Chinese Institute of Electronics, and a member of TC on Digital Forensics and Security of the China Society of Image and Graphics. He is serving as an Associate Editor for IEEE Signal Processing Magazine.