This article has been accepted for publication in IEEE Transactions on Circuits and Systems for Video Technology. 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/TCSVT.2025.3585739

Flexible Partial Screen-shooting Watermarking with Provable Robustness

Mingyue Chen, Xin Liao, Han Fang, Jinlin Guo, Yanxiang Chen, Xiaoshuai Wu

***Abstract*—Screen-shooting watermarking is an effective means of protecting screen content from unauthorized capture and illegal dissemination. However, existing methods are primarily designed for full-image capture, making them ineffective for partial screen-shooting prevalent in real-world scenarios. To address this limitation, we propose FPSMark, a flexible watermarking method tailored for partial screen-shooting that embeds consistent watermarks in multiple uniformly distributed cover blocks. Specifically, considering that robustness requirements vary according to the layout of each image, we model the mathematical relationship between the watermark block count and robustness, proving the flexibility of FPSMark in ensuring partial screen-shooting robustness. Moreover, partial screen-shooting disrupts watermark synchronization, posing challenges for precise watermark localization. To overcome this, we design an intrinsic signal localization network optimized with a hybrid loss. The localization network exploits the inherent distinctions between the watermark and non-watermark features, while the hybrid loss constrains the network at three dimensions: pixel-level, region-level, and sample-level. Experimental results demonstrate the superiority of FPSMark, showing robust performance across partial capture percentages. Its extraction accuracy exceeds 98% even with only half of the image captured, and it achieves 82% accuracy at a 40% capture ratio, whereas existing methods achieve only around 50% under the same conditions.**

***Index Terms*—Robust watermarking, partial screen-shooting, flexibility.**

I. Introduction

DIGITAL watermarking is an important method for copyright protection, leak tracing and robust signal processing [1]–[3]. By embedding invisible identifiers within protected content, it effectively prevents theft. Malicious users may utilize digital channels to steal images. Traditional image watermarking algorithms can achieve traceability in this scenario because they are more concerned about digital channel distortions (e.g., JPEG compression, Gaussian noise) [4], [5]. However, sophisticated thieves may use portable devices like smartphones to capture displayed multimedia content. These distortions are more complex, involving digital-to-analog and analog-to-digital conversions, thus challenging the effectiveness of traditional robust watermarking techniques.

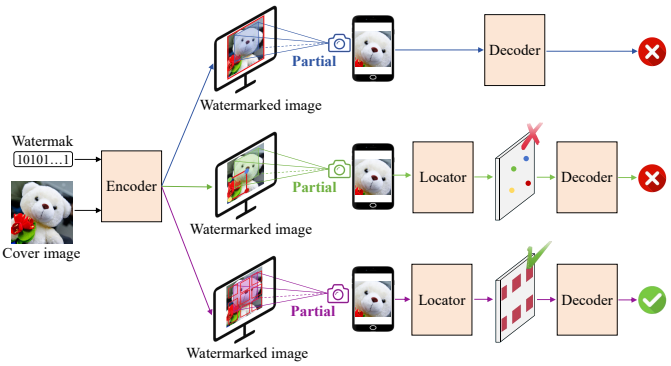


Fig. 1. Process comparison with three modes: entire image encoding (blue branch), sub-image encoding (green branch), and multi-blocks encoding (purple branch). The red box in watermarked image highlights the watermark region, and the camera's field of view indicates the shooting area. Embedding the watermark into the entire cover image cannot resist partial screen-shooting due to the incomplete watermark. Embedding the watermark in sub-image relies on four reference vertices for localization but fails if any vertex is missing in the partially captured image. FPSMark divides the cover image into multiple uniformly distributed blocks and embeds the watermark in each block, making it inherently resistant to partial capture of varying percentages.

Consequently, many researchers have studied screen-shooting watermarking [6]–[9]. Screen-shooting distortions can be categorized into three types: (1) environmental factors, e.g., brightness and contrast; (2) camera factors, e.g., defocus blur and compression; and (3) photographer factors, e.g., motion blur and geometric distortion. Most existing studies employ an encoder-noise layer-decoder architecture [8], [10], [11] to resist these distortions. The encoder and decoder are responsible for embedding and extracting information, respectively, while the noise layer aims to simulate the screen-shooting distortions. Wengrowski *et al.* [8] introduced a large-scale dataset of camera-display paired images and designed a transfer network to learn these distortions. On the other hand, Fang *et al.* [11] provided a new insight, suggesting that focusing on the most influential distortions during training was effective and more conducive to model convergence. However, as shown by the blue branch in Fig. 1, these methods generally rely on capturing entire images, with their robustness significantly decreasing and approaching random guessing when only partial images are available (see Table III).

Another solution mode involves embedding the watermark in a sub-image rather than the entire image and introducing a

This work is supported by the National Natural Science Foundation of China (Grant No. U22A2030), the National Key R&D Program of China (Grant Nos. 2022YFB3103500, 2024YFF0618800), and Hunan Provincial Funds for Distinguished Young Scholars (Grant No. 2024JJ2025).

Mingyue Chen, Xin Liao and Xiaoshuai Wu are with the College of Cyber Science and Technology, Hunan University, Changsha, China, 410082 (e-mail: chenmingyue@hnu.edu.cn; xinliao@hnu.edu.cn; shinewu@hnu.edu.cn).

Han Fang is with the National University of Singapore, Singapore 119077 (e-mail: fanghan@nus.edu.sg).

Jinlin Guo is with the Faculty of System Engineering, National University of Defense Technology, Changsha, China, 410000 (e-mail: gjlin99@nudt.edu.cn).

Yanxiang Chen is with the School of Computer Science and Information Engineering, Hefei University of Technology, Hefei, China, 230601 (chenyx@hfut.edu.cn).

*Corresponding author: Xin Liao

locator to identify the watermarked region before decoding [12]–[14], as shown by the green branch in Fig. 1. For instance, He *et al.* [13] proposed an adaptive amplitude template-based method to assist in instance-level watermark synchronization, while Jia *et al.* [12] used HRNet to locate watermarked sub-image, reducing the time of manually correcting geometric distortions. However, all these methods require four reference vertices to locate the watermark region. In practice, partial capture conditions are uncontrolled and diverse. If keypoints are not captured, it would result in localization errors, which subsequently lead to decoding failures. Therefore, these methods are unable to resist partial screen-shooting attacks.

In practical scenarios, partial image capture frequently occurs due to variables such as shooting angles and distances. Meanwhile, attackers may capture only the portion of the image they are interested in to evade tracking. Additionally, copyright protection requirements differ by image layout and usage context, resulting in different standards for determining infringement in partial captures. For instance, reproducing even 50% of an artistic image, such as a painting or illustration, may still constitute copyright infringement, as the unique value of these works often lies in intricate details. In contrast, for works depicting public scenes, such as landscape or architectural photography, the content is relatively common, and the styles are often similar, requiring a higher reproduction ratio (e.g., 80% or more) to trigger copyright enforcement. Therefore, there is an urgent need to develop partial screen-shooting watermarking methods that adapt to different robustness requirements.

In this paper, we propose a flexible watermarking method for partial screen-shooting, named FPSMark, as shown in Fig. 1. By using block-wise processing to embed consistent watermarks in multiple cover blocks, it is inherently resistant to partial capture. We consider the practical scenarios where copyright protection requirements vary across different images. By analyzing the relationship between the number of watermark blocks and robustness to varying partial capture percentages, we provide a rigorous mathematical proof demonstrating the provability of robustness. Dynamically adjusting the count of watermark blocks according to specific requirements provides a flexible protection strategy. It also effectively mitigates visual degradation from excessive embedding. FPSMark has two training stages. In the first stage, we apply a block-wise approach to divide the cover image into multiple blocks and use them to train an encoder-decoder network, ensuring robustness against distortions caused by screen-shooting. In the second stage, we utilize the intrinsic differences between watermark and non-watermark regions, which are imperceptible to human eyes but detectable by the network, to train an intrinsic signal localization network. To obtain watermark region boundaries with high confidence, we propose a hybrid loss that enhances partial screen-shooting robustness by optimizing the localization network at multiple levels. Our contributions are as follows:

1) We propose FPSMark, a flexible screen-shooting watermarking method designed to address the challenge of partial screen-shooting. It is provably robust to partial capture with varying percentages.
2) We develop a mathematical model that quantifies the relationship between the number of watermark blocks and partial screen-shooting robustness, innovatively proving the flexibility of FPSMark in addressing diverse robustness requirements.
3) We introduce an intrinsic signal localization network that leverages distinctions between watermark and non-watermark features. Additionally, we design a hybrid loss to enhance partial screen-shooting localization accuracy at pixel, region, and sample levels.
4) Extensive experiments demonstrate that our method exhibits strong robustness against partial capture with varying shooting distances and angles, while effectively adapts to various robustness requirements.

The remainder of this paper is organized as follows. Section II discusses related work. Section III conducts a theoretical analysis of the relationship between watermark block count and robustness, formulates their mathematical modeling, and demonstrates that FPSMark is provably robust. Section IV describes the proposed network architecture. In Section V, we present several experimental analyses and training strategies. Finally, Section VI provides the conclusion of this paper.

II. Related Work

In recent years, robust watermarking techniques have been extensively studied to resist various types of distortion. In this section, we first introduce the research on robust watermarking against common digital distortions and then discuss the robust watermarking based on screen-shooting distortion.

A. Digital distortion resistant robust watermarking

Digital distortion resistant robust watermarking effectively mitigates common noise, such as JPEG compression and blurring. It can be broadly categorized into traditional and deep learning-based digital watermarking. Traditional digital watermarking includes spatial and frequency domain techniques. Spatial domain watermarking manipulates image pixels by altering their values or distributions [15], with common algorithms including histogram-based embedding [16] and template-based embedding [17], [18]. On the other hand, frequency domain watermarking transforms the cover image from the spatial to the frequency spectrum. It modifies transform coefficients to encode watermark information using transformations such as the Discrete Cosine Transform (DCT) [19], [20], Discrete Fourier Transform (DFT) [21], and Discrete Wavelet Transform (DWT) [22]. In addition to classical frequency-based techniques, a variety of transform domain and hybrid-domain watermarking methods have been proposed to enhance robustness. For example, Onn *et al.* [23] introduced a two-layer authentication framework based on Integer Wavelet Transform (IWT) and Singular Value Decomposition (SVD), integrating robust and fragile watermarking for tamper detection and copyright protection. Fikri *et al.* [24] exploited the stability of IWT-SVD and incorporated human visual characteristics (HVC) to guide embedding, achieving a balance between robustness and imperceptibility. Gen *et al.* [25] embedded dual watermarks in both DWT and spatial domains to support

simultaneous copyright authentication and tamper detection. Ridwan *et al.* [26] proposed a Schur decomposition-based dual watermarking scheme with a threshold control mechanism to balance robustness and visual quality. These traditional methods perform well against common digital distortions such as compression and noise. However, they rely on manually designed features for embedding and extraction, providing robustness only against specific distortions and lacking generalizability.

With the advancement of deep learning, new perspectives for watermarking technology have emerged. A series of exemplary deep learning-based digital watermarking techniques has been developed. For instance, Zhu *et al.* [27] introduced Hidden, the first end-to-end training framework, comprising an encoder, noise layer, and decoder. The encoder embeds the watermark into the cover image, the noise layer simulates digital distortions in a differentiable manner, and the decoder extracts the watermark from the distorted image. To enhance robustness against JPEG compression, Jia *et al.* [5] randomly selected real JPEG and simulated differentiable JPEG for each mini-batch during training. Ma *et al.* [28] combined reversible and non-reversible mechanisms with a non-reversible attention module to address asymmetric extraction under noise attacks. Liu *et al.* [29] proposed TSDL, a two-stage learning architecture, initially training the encoder and decoder without noise, then fine-tuning the decoder to adapt to black-box distortions. Wang *et al.* [30] introduced an adapter to select appropriate strength factors for each image, thereby improving watermark invisibility. These methods are primarily designed for digital transmission scenarios and struggle to handle distortions introduced by screen-shooting, as the distortion types differ significantly, with screen-shooting distortions being more complex and challenging to address.

B. Screen-shooting distortion resistant robust watermarking

Screen-shooting distortion is complex, and some researchers have employed traditional methods to design embedding algorithms that improve watermark invisibility and robustness. Fang *et al.* [6] proposed a novel screen shooting resilient watermarking approach that utilizes an intensity-based scale-invariant feature transform (SIFT) algorithm to identify suitable regions for embedding well-designed watermark templates. To enhance real-world applicability, Wengrowski *et al.* [8] constructed a 1.9 TB camera-display dataset for training distortion network. However, this method relies on extensive datasets from multiple devices, which limits its ability to generalize across different devices. Consequently, many researchers have adopted simulation methods. Tancik *et al.* [7] introduced StegaStamp, which decomposes the physical capturing process into a series of image operations. Following this, Jia *et al.* [10] incorporated a 3D rendering module to more accurately simulate camera imaging distortion. Fu *et al.* [31] introduced a recovery network to gradually remove noise during decoding phase to improve accuracy. Fang *et al.* [11] simplified the noise layer by retaining only critical noises such as perspective distortion, illumination distortion, moiré distortion, and Gaussian noise. They proposed PIMoG, which achieves competitive performance. To address grayscale deviation under different viewpoints, Li *et al.* [32] modeled the relative positions of the viewing point and screen plane to simulate such distortion. However, most of existing methods require manual localization of the watermark region before decoding, increasing the complexity of watermark extraction.

To automatically locate the watermark region, Jia *et al.* [12] proposed embedding watermark in sub-image and incorporated a localization network within an end-to-end framework, ensuring both visual quality and robustness while simplifying the extraction process. Similarly, Zhu *et al.* [14] utilized Lite-HRNet as a localization network to improve training efficiency. He *et al.* [13] proposed a template encoding module that embeds watermarks into image instances, resisting instance-level screen-shooting attacks. However, the above methods rely on capturing entire watermarked images, making them ineffective in partial screen-shooting scenarios and limiting their applicability.

III. Mathematical Modeling of Watermark Block Count and Robustness

In practical applications, the robustness requirements of partial screen-shooting are different. Excessive watermark blocks can lead to over-embedding and degrade visual quality, while too few blocks reduce robustness. Therefore, selecting an appropriate watermark block count is essential to guarantee specific robustness. In this study, we define the "capture percentage triggering copyright tracking" as the infringement threshold. Exceeding this threshold triggers copyright tracking to identify potential infringement. We develop a mathematical model to quantify the relationship between watermark block count and robustness, where robustness is represented by the infringement threshold, demonstrating that FPSMark is provably robust.

Entire watermark blocks are crucial for copyright tracking. In practice, partial screen-shooting robustness is not only related to the integrity of the watermark block, but also affected by various distortions, such as noise and JPEG compression. However, these distortions are difficult to precisely quantify and analyze mathematically. In addition, the incompleteness of watermark block has a greater impact on robustness than other distortions, as the entire watermark block provides the necessary information for watermark recovery, while an incomplete watermark block reduces accuracy. Therefore, even in the presence of other distortions, the integrity of the watermark block remains the most fundamental condition for ensuring robustness. Therefore, in this study, we simplify the assumption by ignoring distortion factors, as long as the existence of the entire watermark block is guaranteed, it can be regarded as the FPSMark has robustness.

Therefore, we need to make sure that at least one entire watermark block is included within the infringement threshold. To achieve this, we employ reverse thinking to calculate the maximum capture percentage, which does not include any entire watermark block. We define the size of the cover image as $n \times n$ and divide it into $k \times k$ blocks, with a uniform gap of g between adjacent blocks and between boundary blocks

and the image edges. The size of a block is $m \times m$. The gap-to-block size ratio is t. The following equation expresses the relationship:

$$n = km + (k+1)tm \tag{1}$$

As shown in Fig. 2, one edge of the cover image contains k watermark blocks and $k+1$ gaps, where $g = tm$. From this, the block size m can be derived as:

$$m = \frac{n}{k(t+1)+t} \tag{2}$$

We assume that the side lengths of the rectangular capture area are x and y. When both $x < m$ and $y < m$, the area is smaller than watermark block size m^2, ensuring that no entire watermark block is captured. To find the maximum area excluding any entire watermark block, we gradually increase the capture area's side lengths. First, x increases to n while $y < m$, ensuring no entire watermark block is captured. When increasing y such that it spans two watermark blocks without aligning with the block boundaries (i.e., $y < 2m+g$), no entire watermark block is captured. Further increases in y will inevitably cover a entire watermark block. Therefore, the maximum capture percentage P_{uncom} without any entire watermark block occurs when one side of the captured area coincides with the watermarked image edge, while the other side spans exactly two watermark blocks. It is defined as:

$$P_{uncom} = \frac{n(2m+g)}{n^2} = \frac{(2+t)m}{n} = \frac{2+t}{k(t+1)+t} \tag{3}$$

When the actual capture percentage $P \geq P_{uncom}$, it ensures that at least one entire watermark block will be included, which proves the robustness of partial screen-shooting. Therefore, the mathematical model defining the relationship between the watermark block count and the infringement threshold P_{in} is as follows:

$$P_{in} \geq \frac{2+t}{k(t+1)+t} \tag{4}$$

Here, t is a hyperparameter. Since m and g are constrained to integers in different block-wise strategies, t cannot be strictly fixed. Inspired by [33], we set t to be about 30%, and the specific value adjusts with the block-wise strategy.

Subsequently, we illustrate the block-wise strategy with an example. For a 400×400 cover image, suppose the user requires a robustness of 60%, meaning copyright tracking will be activated when 60% or more of the image is illicitly captured. According to Eq. 4, the k satisfies $k \geq 2.72$, where $k \in Z^+$. We analyze the monotonicity of P_{in} to k:

$$\frac{\partial P_{in}}{\partial k} = -\frac{(2+t)(t+1)}{(k(t+1)+t)^2} \tag{5}$$

Since both t and k are greater than 0 ($k, t > 0$), it follows that $\frac{\partial P_{in}}{\partial k} < 0$. This indicates that as k increases, the infringement threshold P_{in} decreases, leading to higher robustness. However, increasing k (i.e., embedding watermarks in more cover blocks) degrades visual quality. Therefore, in this scenario, $k = 3$ represents the optimal choice, balancing robustness and visual quality. Based on this analysis, we proceed to compute the values of m and g:

$$\begin{cases} 400 = 3m + 4tm \\ t = \frac{g}{m} \approx 0.3 \\ m \in Z^+, m \equiv 0 \pmod 2, g \in Z^+ \end{cases} \tag{6}$$

Since neural network models typically require the input image dimensions to be even (e.g., due to the sensitivity of convolutional operations to input sizes), we constrain the parameter m to satisfy $m \equiv 0 \pmod 2$, ensuring compatibility. Further, we derive the values of m and g as 96 and 28, respectively. From Eq. 4, $P_{in} \geq 55\%$ in this scenario, meaning the watermarking method is robust against partial screen-shooting when the capture percentage exceeds 55%.

Through the quantitative analysis of the watermark block count and infringement threshold, we can determine the watermark block permutation (including the number of blocks, block size, and gap size) based on the partial screen-shooting robustness requirements. This demonstrates the flexibility of our method. Furthermore, based on the number of watermark blocks, we can calculate the infringement threshold, proving that FPSMark has provable robustness.

IV. Proposed Framework

The framework of the proposed FPSMark (illustrated in Fig. 2) follows the classic encoder-noise layer-decoder pipeline, which consists of five main parts: encoder E_{nc}, noise layer N, decoder D_{ec}, discriminator D_{is}, and localization network LOC. The training strategy consists of two stages, following a logical progression where robust watermark embedding is established first to provide reliable features for subsequent localization. In Stage-1, the watermark encoder E_{nc} and decoder D_{ec} are trained using block-wise procedure to ensure robustness against image processing distortions caused by partial screen-shooting. In Stage-2, the intrinsic signal localization network LOC is trained to achieve refined localization. Finally, during inference stage, the models trained in two stages are integrated, enabling FPSMark to handle various partial screen-shooting scenarios flexibly.

A. Encoder

Our method aims to embed watermark messages into multiple blocks of the cover image I_o while achieving better robustness and visual quality. To effectively handle partial screen-shooting, we select a set of uniformly distributed, non-overlapping blocks x_o from the cover image to embed a consistent watermark. Meanwhile, a binary template T (with watermark regions set to 1 and others set to 0) is preserved as a reference for localization. This encoding is inherently robust to partial capture, as even if only a part of the image is captured, it may still have several entire encoded blocks sufficient for watermark decoding.

The encoding process is performed by the encoder E_{nc}, which embeds the binary watermark W of length L into multiple cover blocks. The cover image is an $n \times n$ RGB image, divided into non-overlapping subregions of size $m \times m$. Our encoder is based on the network architecture from [5],

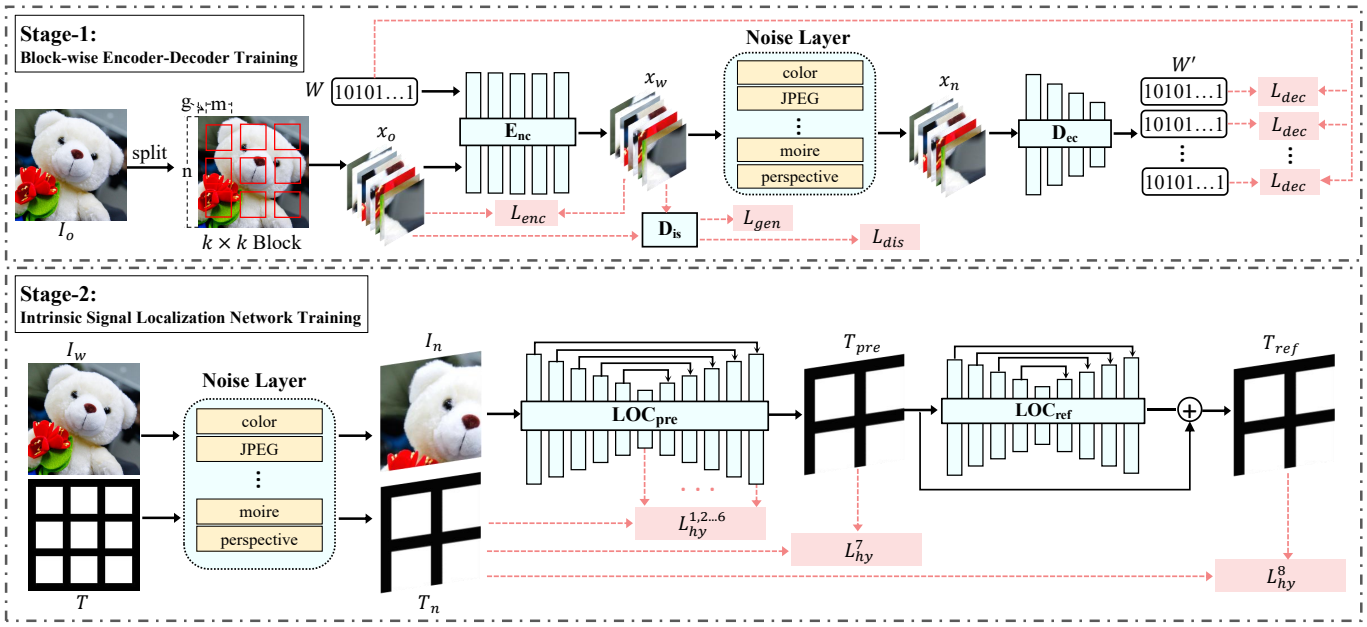


Fig. 2. The framework of the proposed FPSMark. It consists of two training stages. In Stage-1, the cover image I_o is uniformly split into multiple blocks of size $m \times m$ with a gap of g between the blocks, denoted as x_o. The encoder E_{nc} embeds a consistent watermark W into these blocks. The noise layer simulates partial screen-shooting noise attacks, and the decoder D_{ec} extracts the watermark $W^{'}$ from the noisy blocks x_n. At the same time, a discriminator D_{is} is introduced to improve the visual quality. In Stage-2, the intrinsic signal localization network is trained. A pre-localization module LOC_{pre} identifies coarse watermark region template T_{pre} from noisy image I_n, followed by a refinement module LOC_{ref} for precise watermark localization.

[33], with the Squeeze-and-Excitation (SE) module [34] serving as the basic block. Firstly, the watermark is repeated and reshaped to $m \times m$, followed by a 3×3 convolution, batch normalization, and ReLU activation (denoted as ConvBNReLU) to generate the watermark tensor. Meanwhile, the image block is processed by ConvBNReLU and the SE module to obtain the image tensor. Secondly, the watermark and image tensor are concatenated, followed by a series of SE modules, upsampling, and downsampling to generate the watermarked block x_w. Throughout this process, intermediate features from the watermark and image block are repeatedly concatenated to achieve a trade-off between robustness and invisibility. Finally, the corresponding image blocks in the cover image are replaced with watermarked image blocks to obtain watermarked image I_w.

To ensure that the watermarked image block is similar to the cover image block, we train the encoder with MSE loss and LPIPS [35] loss.

$$L_{mse} = \left\| \gamma(x_w) - \gamma(x_o) \right\|^2 \tag{7}$$

$$L_{enc} = \lambda_1 L_{mse} + \lambda_2 L_{lpips}(x_w, x_o) \tag{8}$$

where $\gamma(.)$ is a differentiable non-parametric mapping function from RGB to the YUV space, and L_{lpips} denotes the LPIPS loss widely used for evaluating image quality. The values of λ_1 and λ_2 are both set to 2.

B. Noise Layer

Since the screen-shooting process involves complex distortions, we decompose these and design a differentiable noise layer between the encoder and decoder to simulate the overall procedure. Handheld camera capture may cause image blur, which we simulate using Gaussian blur with a kernel size randomly selected from $\{3, 5, 7\}$. When the user captures an image at an angle that is not parallel to the screen, the resulting image may exhibit geometric distortions. To simulate this, we introduce perspective distortion by randomly perturbing the four corners of the image within 10% of the original image size. The homography is then computed to map the original corner positions to their perturbed locations. In addition, environmental factors such as lighting can also introduce distortions. We simulate these using the algorithm from [11]. Given the diversity of camera and screen brands, variations in color domains are inevitable. Therefore, we consider brightness, contrast, and saturation distortions, with their degrees constrained within the range $[0.5, 1.5]$. Camera system noise is simulated using Gaussian noise with a mean of 0 and variance ranging from $[3, 10]$. Spatial frequency aliasing between the camera sensor pixel array and the screen pixel array can result in moiré patterns, which we simulate using the moiré distortion from [11]. When images are stored, they often undergo JPEG compression, which can introduce artifacts. To simulate this process, we employ a mixed noise approach that combines real JPEG compression, simulated JPEG compression, and an identity transformation, as proposed by [5]. Losses during image transmission are modeled with dropout, where a random proportion of pixels, ranging from $[0, 0.3]$, from the watermarked image is replaced by the corresponding pixels from the cover image. Different cameras store images in various sizes, so before decoding, all images are resized to a uniform dimension. To account for this, we introduce resize distortion by randomly scaling the watermarked image by a

factor in the range $[0.5, 2]$.

Unlike full capture, we also consider cases where the user captures only part of the image, simulated using crop distortion. A region with dimensions $c_1H \times c_2W$ is randomly cropped from the watermarked image. During the training of the localization network in Stage-2, c_1 and c_2 are set to $[0.7, 1]$ to simulate a crop that captures approximately 50% of the image. Since the distorted image input to the decoder is cropped from the captured large image after localization, images with excessive distortion are considered invalid. Therefore, in Stage-1, c_1 and c_2 are set to smaller values, within the range $[0.9, 1]$. This is the only difference in the noise layers between the two stages.

By integrating these distortions into the training pipeline, we ensure that our approach closely aligns with real-world screen capture scenarios, enhancing its robustness and adaptability to practical environments.

C. Decoder

The decoder receives the noised image I_n generated by the noise layer, aiming to recover the watermark information $W^{'}$. Inspired by [33], we design our decoder to align with the encoder, using SE as the core module. The noised image is processed through ConvBNReLU, three SE modules, and downsampling operations to extract watermark feature. Finally, a linear layer ensures that the output dimension matches the length of the original watermark information. The decoder is trained using MSE as the loss function.

$$L_{dec} = \left\| W^{'} - W \right\|^2 \tag{9}$$

D. Discriminator

To improve the visual quality of watermarked images, we introduce adversarial learning during training. Given the outstanding performance of PatchGAN [36] to generate more realistic and detailed images, we adopt it as the discriminator D_{is} to maximize the probability of accurately distinguishing between real and generated images. It is optimized by minimizing the following loss:

$$L_{dis} = \log(1 - D_{is}(I_w)) + \log(D_{is}(I_o)) \tag{10}$$

Meanwhile, the encoder serves as an adversary to the discriminator, generating images similar to the cover image. This is achieved by minimizing the following:

$$L_{gen} = \log(D_{is}(I_w)) \tag{11}$$

E. Localization Network

The localization network aims to precisely identify the position of watermark in captured images and output a binary template image. Existing methods, such as [12], [14], use the HRNet [37] keypoints detection network as a locator, achieving notable performance in localizing watermark sub-image. However, these methods rely on four reference vertices and heatmaps, requiring the input image to contain the entire target region. In partial screen-shooting, where only partial images are captured, and one or more reference vertices may be missing, the performance of HRNet degrades significantly.

To tackle this issue and improve localization accuracy, we explore the intrinsic signal of images for localization. Specifically, by exploiting feature differences between watermarked and non-watermarked regions, which are imperceptible to human eyes but detectable by localization network, we design a localization framework based on U^2-Net† [38], termed the intrinsic signal localization network LOC. Unlike existing methods, LOC operates independently of four reference vertices, providing a more robust and flexible solution.

As shown in Stage-2, the LOC consists of two cascaded sub-networks: pre-localization module LOC_{pre} and refinement module LOC_{ref}. The LOC_{pre} is based on U^2-Net†, a lightweight variant of U^2-Net, designed to capture both global and local features through multi-level feature extraction and contextual information fusion. While U^2-Net† performs well in general image segmentation tasks [38], directly applying it to watermark localization poses two key challenges. First, geometric noise, such as perspective distortion introduced by screen-shooting, imposes higher requirements on network robustness. Second, the high density of watermark blocks requires more precise boundary localization, exceeding the demands of general semantic segmentation tasks.

To address these, we introduce the refinement module LOC_{ref} following LOC_{pre}. LOC_{ref} refines the predicted results by learning the residual between the pre-localized template T_{pre} and the ground truth (GT), which is generated by applying the same noise processing to the binary template T as done on the watermarked image. This residual learning enables the network to capture features from irregular boundaries caused by geometric distortions. LOC_{ref} employs a lightweight U-Net architecture [39], incorporating multi-level feature extraction to enhance both local and global contextual understanding. The U-Net encoder consists of four ConvBNReLU modules and pooling layers to extract high-level features. The bridge layer consists of a 3×3 convolutional layer, while the decoder restores spatial resolution through bilinear interpolation and four ConvBNReLU modules, achieving sharper boundary localization. This design ensures precise localization of watermark block boundaries.

The implementation details are as follows: the $n \times n$ watermarked image I_w, obtained from the Stage-1, undergoes noise layer attacks to produce a noisy image I_n. Concurrently, the clean template T undergoes similar processing to yield the noisy template T_n. The noisy image I_n is then fed into the localization network LOC. First, the initial sub-network generates a $n \times n$ binary pre-localization template T_{pre}. Subsequently, the second sub-network produces a $n \times n$ refinement residual, which is added to T_{pre}, resulting in the final refined localization template T_{ref}.

To achieve precise localization, we propose a hybrid loss function composed of Binary Cross Entropy (BCE) [40], IoU [41] and Focal loss [42], which provides supervision at pixel, region, and sample levels, respectively. BCE minimizes pixel-wise differences between the predicted and GT templates, while IoU measures their overlap to optimize the overall region. Focal Loss addresses sample imbalance by assigning

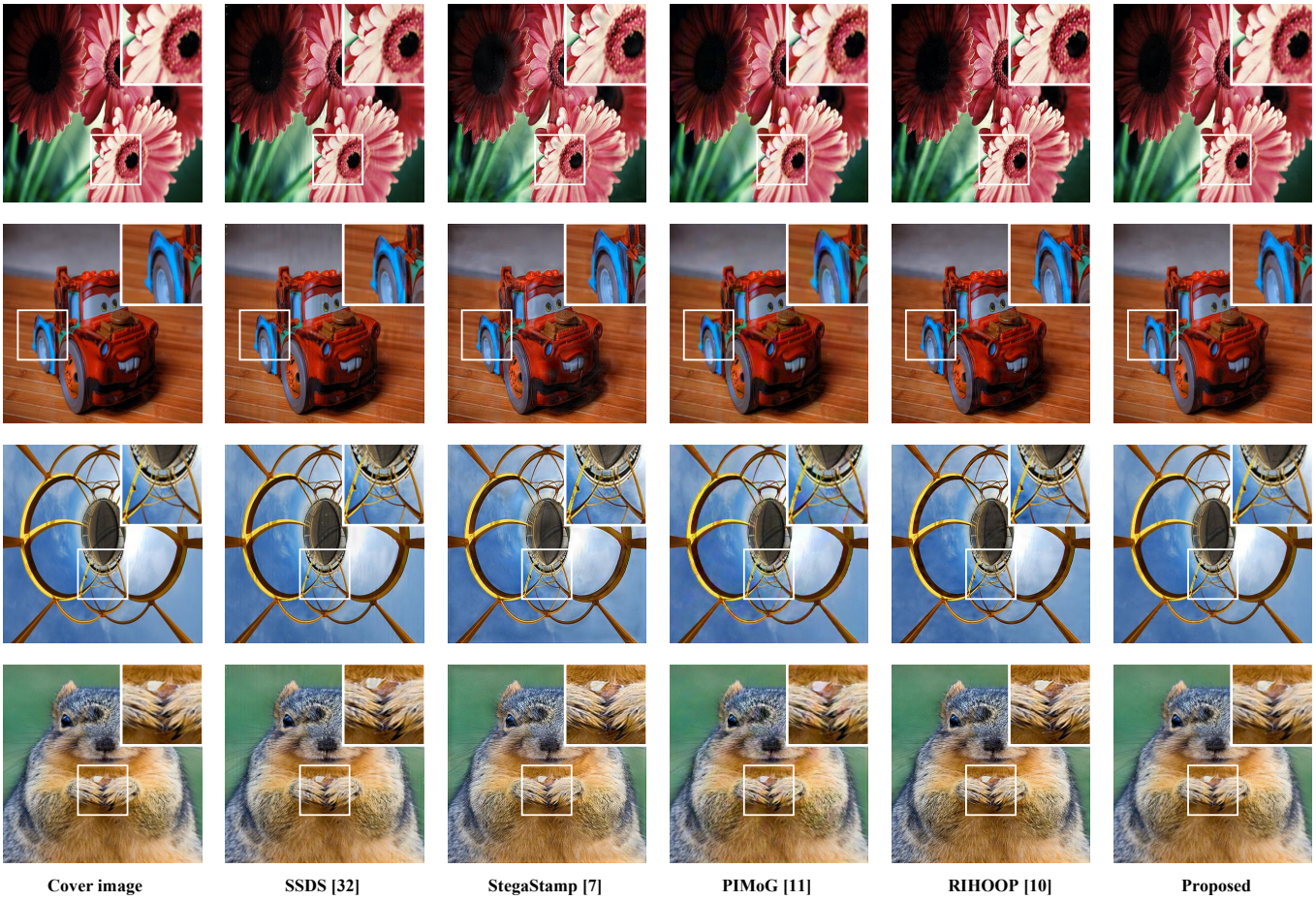


Fig. 3. The visual quality and details of watermarked images embedded with different methods. A detailed zoom-in of the white rectangular region is provided in the upper-right corner. Our method demonstrates superior performance, with no visible artifacts or color changes.

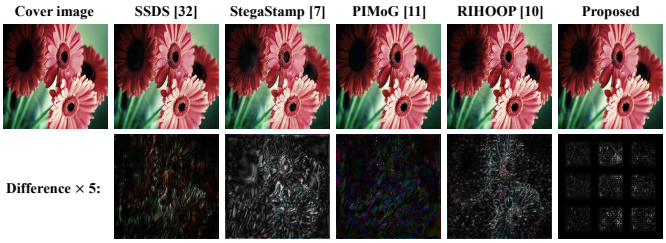


Fig. 4. Visual examples of watermarked images and watermark residuals. The first row shows the watermarked images, and the second row displays the difference, multiplied by 5.

higher weights to hard samples, particularly the boundaries between watermark and non-watermark regions, encouraging the model to focus on these areas. The hybrid loss function is defined as follows:

$$L_{hy} = L_{bce}(T_{ref}, T_n) + L_{iou}(T_{ref}, T_n) + L_{foc}(T_{ref}, T_n) \tag{12}$$

where L_{bce}, L_{iou}, L_{foc} denote BCE loss, IoU loss and Focal loss, respectively.

To mitigate overfitting, we refer to HED [43] and apply direct supervision from the GT template to the final layer in each decoder stage of U^2-Net†. In summary, the loss function of the localization network is defined as follows:

$$L_{loc} = \sum_{k=1}^{K} \alpha_k L_{hy}^{(k)} \tag{13}$$

where $L_{hy}^{(k)}$ is the L_{hy} of the k-th side output, K denotes the total number of the outputs and α_k is the weight of each loss. In our localization network, $K = 8$, comprising seven outputs from L_{pre} and one output from L_{ref}. The loss weight for the final outputs of LOC_{pre} and LOC_{ref} is 1, while that for the intermediate outputs of LOC_{pre} is 0.25.

Given the complexity of screen-shooting noise, the localized watermark blocks may not be regular rectangles, necessitating correction. We apply erosion, a classic morphological operation, to remove boundary noise and then employ the "findContours" algorithm from OpenCV to extract the contours of the watermark blocks. To prevent smaller blocks from degrading decoding performance, we set a minimum area threshold β relative to the largest watermark block, retaining only blocks meeting this condition. Finally, perspective transformation is performed using the four corner points of each contour.

F. Training Strategy

The model consists of several components, each with distinct training objectives, which makes joint training difficult to

TABLE I
THE PSNR AND SSIM VALUES OF EACH METHOD

Methods	SSDS [32]	StegaStamp [7]	PIMoG [11]	RIHOOP [10]	Proposed
PSNR (db)	33.63	27.60	34.67	31.80	**35.54**
SSIM	0.963	0.862	0.972	0.925	**0.972**

TABLE II
THE EXTRACTION ACCURACY FOR PARTIAL IMAGES AT VARIOUS SHOOTING DISTANCES

Distances (cm)	18	17	16	15	14	13
Percentages	100%	90%	80%	70%	60%	50%
SSDS [32]	99.20%	88.45%	61.14%	60.35%	54.95%	48.30%
StegaStamp [7]	99.35%	97.44%	97.99%	86.84%	63.62%	52.74%
PIMoG [11]	95.86%	98.11%	76.00%	50.67%	48.00%	49.22%
RIHOOP [10]	98.64%	86.26%	58.44%	45.83%	53.85%	54.24%
Proposed	**99.91%**	**99.70%**	**98.21%**	**99.05%**	**96.58%**	**98.31%**

converge. To address this challenge, we draw inspiration from [29], where fixing certain parameters allows each component to achieve a local optimal solution. Thus, a two-stage training strategy is employed: in Stage-1, the encoder and decoder are trained to achieve a trade-off between visual quality and robustness; in Stage-2, the encoder is fixed, and the localization network is optimized to improve watermark localization accuracy. This staged training strategy helps the model train more effectively and tune better.

1) Stage-1: Block-wise Encoder-Decoder Training: In this stage, we focus on the joint optimization of the encoder, decoder, and discriminator to ensure that the watermarked image achieves high visual quality and sufficient robustness. The total loss function for this stage is defined as follows:

$$L_{Stage-1} = \lambda_{enc} L_{enc} + \lambda_{dec} L_{dec} + \lambda_{gen} L_{gen} \tag{14}$$

where λ_{enc}, λ_{dec} and λ_{gen} are set to 1, 10 and 0.5, respectively. The optimization objective for the discriminator is to minimize L_{dis}.

2) Stage-2: Intrinsic Signal Localization Network Training: In Stage-2, the pre-trained encoder from Stage-1 is used to generate watermarked images. As the captured images contain multiple watermark blocks, the localization network is employed to accurately identify the watermark regions, facilitating precise and efficient decoding. At this stage, only the localization network is trained, with the loss function defined as follows:

$$L_{Stage-2} = L_{loc} \tag{15}$$

G. Inference Stage

1) Message Fusion: Due to the image embedding multiple identical watermarks, the captured image may contain multiple or incomplete watermark blocks, and the complex noise introduced during screen-shooting can cause variations between these blocks. Therefore, message fusion is essential for extracting the final watermark information. We adopt the message fusion strategy based on information similarity proposed in [33] to determine the final watermark. Differently, we prioritize using watermark blocks that are not located at the boundaries for fusion, as boundary blocks may be incomplete. If no blocks away from the boundaries are available, fusion is performed on all blocks.

TABLE III
THE EXTRACTION ACCURACY FOR PARTIAL IMAGES AT VARIOUS SHOOTING POSITIONS

Positions (cm)	Left 10	Left 5	0	Right 5	Right 10
Percentages	40%	70%	100%	70%	40%
SSDS [32]	46.70%	51.90%	99.20%	51.48%	47.63%
StegaStamp [7]	52.43%	53.48%	99.35%	54.01%	51.90%
PIMoG [11]	49.67%	43.67%	95.86%	44.34%	51.04%
RIHOOP [10]	50.97%	50.85%	98.64%	51.11%	49.72%
Proposed	**82.55%**	**99.54%**	**99.91%**	**99.23%**	**83.40%**

2) Flexible Application: In practical applications, to accommodate varying requirements for partial screen-shooting robustness, users first employ the mathematical model to determine the block strategy, which divides the cover image I_o into multiple image blocks x_o. Specifically, the number of watermark blocks is calculated by Eq. 4, and the size of each image block is determined by Eq. 6. These are prior knowledge that remains fixed during inference and can be easily maintained through metadata. A pre-trained encoder is then employed to embed the watermark W, producing a watermarked image I_w. This image may be illicitly captured during screen display and subsequently transmitted over the internet. Upon detection, users can authenticate copyright ownership based on the captured image. The same watermark block configuration used during encoding is applied during decoding, where the suspicious image is manually cropped into a rectangle to remove irrelevant content and then processed through our network for localization, correction, decoding, and message fusion, ultimately extracting the final watermark information to verify copyright ownership. This user-customizable block partitioning strategy highlights the flexibility of our method.

V. EXPERIMENTAL RESULTS

This section first presents the implementation details of the proposed method. We then conduct a series of experiments to evaluate the performance of FPSMark in terms of visual quality and robustness, under both partial and full screen-shooting. Next, we verify the flexible robustness of the proposed method. To assess its generalization in digital transmission scenarios, we evaluate its performance under various digital distortions. Finally, an ablation study is performed to validate our design.

A. Implementation Details

To train our network, we randomly select 10000 images from the COCO dataset [44] as our training dataset. For comparison with previous methods, we use the Mirflickr database [45], which was utilized in StegaStamp [7], as our test set. All images are resized to 400×400. The size of each block is 96×96, and the watermark messages are randomly generated bit strings, each of length 100. This configuration

TABLE IV
THE EXTRACTION ACCURACY FOR PARTIAL IMAGES AT VARIOUS SHOOTING ANGLES

Angles (°)	Left 30	Left 20	Left 10	Right 10	Right 20	Right 30
Percentages	40%	60%	80%	80%	60%	40%
SSDS [32]	46.97%	48.83%	61.82%	60.75%	48.35%	46.80%
StegaStamp [7]	50.97%	53.82%	54.55%	53.78%	52.80%	51.22%
PIMoG [11]	45.58%	44.65%	46.25%	45.74%	43.98%	45.19%
RIHOOP [10]	51.70%	51.04%	50.52%	49.42%	49.87%	50.38%
Proposed	**74.40%**	**99.25%**	**99.88%**	**99.37%**	**98.56%**	**72.58%**

TABLE V
THE EXTRACTION ACCURACY AT DIFFERENT SHOOTING DISTANCES

Distance (cm)	20	30	40	50	60
SSDS [32]	99.25%	99.39%	99.65%	99.35%	**99.05%**
StegaStamp [7]	99.48%	99.70%	99.48%	99.22%	98.19%
PIMoG [11]	98.59%	98.56%	97.40%	97.06%	96.67%
RIHOOP [10]	99.75%	99.80%	99.73%	99.62%	98.29%
Proposed	**99.97%**	**100.00%**	**99.83%**	**99.66%**	98.23%

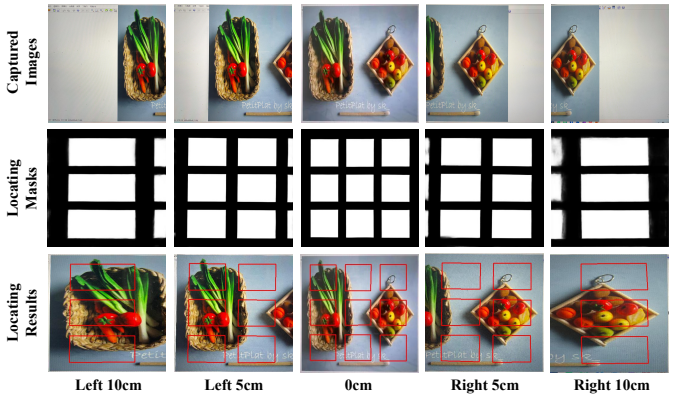


Fig. 5. Captured images and localization performance for partial images at various shooting positions.

ensures consistency with prior works for fair comparison. The entire framework is implemented using the PyTorch library and executed on an NVIDIA RTX 4090 GPU. We use the AdamW optimizer [46] to minimize the loss function, with a learning rate of 1e-4. For localization, the minimum area threshold β is 0.95. In the training Stage-1, we set the batch size to 32, and in the Stage-2, it is reduced to 8 due to the larger size of the images to be processed. Each stage is trained for 300 epochs.

For experiments involving manual shooting, we randomly select 100 images from the test set. We use three monitors to display the watermarked image: Lenovo XiaoXin Pro 14, ENVISION G249G, and AOC 27E12HM. In order to capture the watermarked image, we select three mobile devices equipped with cameras: an iPhone 12, a Realme GT7 Pro, and a SAMSUNG S20FE. To ensure fairness in the experiment, each mobile device is mounted on a tripod and remotely operated via Bluetooth to avoid errors due to manual handling. In this study, we compare our proposed method with four representative screen-shooting watermarking methods: SSDS [32], StegaStamp [7], PIMoG [11], and RIHOOP [10]. All

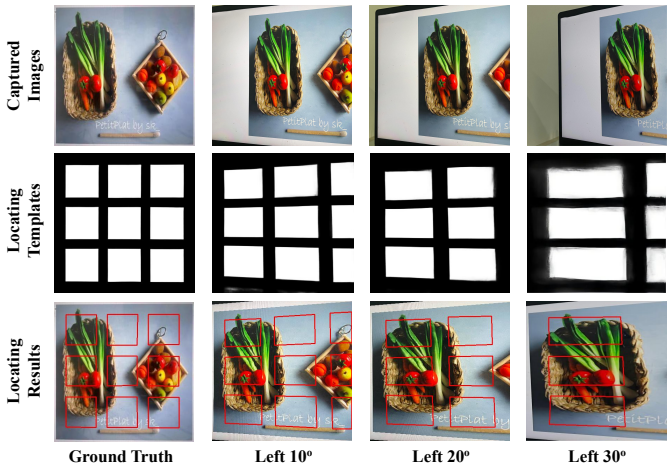


Fig. 6. Captured images and localization performance for partial images at various shooting angles.

methods are evaluated for robustness against practical screen-shooting attacks. The default shooting device is iPhone 12, and the default display is Lenovo XiaoXin Pro 14. In the comparative experiments, watermarked images for all methods are captured under the same conditions.

To evaluate the visual quality of the watermarked images compared to the cover images, we use standard metrics, Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM), as performance indicators. For robustness, we measure it by bit accuracy, which is the ratio of identical bits to total bits between input watermark W and output watermark $W^{'}$. In addition, to evaluate the performance of the localization network, we use the Intersection over Union (IoU) as the metric.

B. Visual Quality

In this section, we compare the visual quality of watermarked image with different methods. As shown in Table I, the proposed method achieves the highest PSNR of 35.54 dB and the SSIM of 0.972, demonstrating its competitive performance compared to other methods.

Additionally, we give qualitative comparison results in Fig. 3, with a detailed zoomed-in view shown in the upper-right corner. The locally enlarged image reveals that SSDS [32] and StegaStamp [7] produce noticeable artifacts, whereas our method preserves superior visual quality. Furthermore, as illustrated in Fig. 4, our watermark effectively covers the entire embedding region in the cover image, providing a reference for subsequent watermark localization.

C. Partial Screen-shooting Robustness

To evaluate the robustness of the proposed method against partial screen-shooting, we consider three typical scenarios: First, as the shooting distance decreases, capturing partial images becomes more likely. Second, when the shooting position moves parallel to the screen towards the boundary, only the edge portion of the image is captured. Third, the deviation in shooting angle may lead to capturing a distorted partial image.

TABLE VI
THE EXTRACTION ACCURACY AT DIFFERENT SHOOTING ANGLES

Angles (°)	Left 40	Left 30	Left 20	Left 10	Right 10	Right 20	Right 30	Right 40
SSDS [32]	**99.13%**	99.15%	99.14%	99.42%	99.35%	99.00%	**99.05%**	**98.50%**
StegaStamp [7]	97.88%	98.77%	99.31%	99.45%	99.43%	99.21%	98.70%	97.01%
PIMoG [11]	94.21%	96.35%	97.22%	97.16%	96.81%	96.65%	95.62%	93.89%
RIHOOP [10]	99.10%	**99.27%**	99.34%	99.31%	99.24%	99.28%	98.54%	98.24%
Proposed	98.02%	99.15%	**99.41%**	**99.50%**	**99.39%**	**99.45%**	98.43%	97.86%

1) Partial Shooting Test at Various Distances: Due to the complexity of the partial capture process, it is challenging to control the capture percentage precisely, but it can still be approximated. In this scenario, we set the capture region to 1:1 to obtain square images. The entire image is captured at a shooting distance of 18 cm, which corresponds to a specific Field of View (FOV) defined by the camera's optical parameters. Since the captured image size is proportional to the shooting distance, we change the distance to capture partial images at different percentages. For instance, at a shooting distance of 16 cm, the capture area is approximately 80% of the full image. The decoding results at different shooting distances are shown in Table II.

As shown in Table II, our method achieves over 96% accuracy with partial shooting percentages exceeding 50%, demonstrating superior robustness to partial capture. This highlights the advantages of our block embedding strategy, as well as the resilience of the localization network and decoder to screen-shooting noise. In contrast, most comparative methods can accurately extract the watermark when the partial shooting percentage exceeds 80%, primarily due to the redundancy inherent in the embedded watermark. However, when the capture percentage is reduced to 80%, only StegaStamp [7] and PIMoG [11] remain effective in extracting watermark information. This is because both methods incorporate perspective transformations in their noise layers, granting them some degree of resistance to scaling and cropping. Notably, PIMoG shows weaker robustness to local capture than StegaStamp, as the perspective transformation in PIMoG is smaller (8 pixels) than the 40 pixels in StegaStamp. Furthermore, when the capture distance is reduced to 13 cm, corresponding to a partial shooting percentage of approximately 50%, all comparative methods fail to extract the watermark, while our method achieves an extraction accuracy of 98.31%, demonstrating significant robustness to local capture. This further validates the robustness of our approach, demonstrating its strong potential for practical implementation in real-world scenarios.

2) Partial Shooting Test at Various Positions: In this scenario, we fix the shooting distance at 18 cm and move the phone parallel to the screen to capture partial images at different scales. Since the captured images are rectangular, the difference in aspect ratio introduces a greater challenge to the localization network. The decoding results and localization performance at different shooting positions are shown in Table III and Fig. 5.

Table III illustrates that our method significantly outperforms the comparison methods, with all results exceeding 82%, while the comparison methods are around 50%, nearing random guess. This indicates that existing methods are ineffective under partial shooting. Specifically, when the movement distance is 10 cm, the local capture percentage is only 40%, with the captured watermarked image region exhibiting an aspect ratio of nearly 5:2. Despite this extreme aspect ratio imbalance, our localization network accurately identifies the watermark region (see Fig. 5), demonstrating the robustness of our method against scaling. In summary, our method exhibits superior accuracy and robustness in handling partial shooting and extreme aspect ratio imbalances, markedly outperforming existing comparison methods.

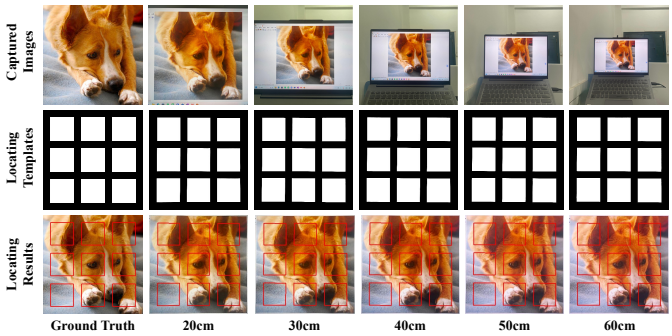


Fig. 7. Captured images and localization performance at different shooting distances. The second row shows the locating template images from localization network, and the third row displays the visualized localization results.

3) Partial Shooting Test at Various Angles: We fix the shooting distance at 18 cm, positioning the camera at the center, and adjust the phone's angle to capture partial images with perspective distortion. In this scenario, the large image scaling and geometric transformations introduce a greater challenge to the robustness of the localization network. The decoding results and localization performance at different shooting angles are shown in Table IV and Fig. 6.

As shown in Table IV, our method demonstrates robust performance across various angular offsets. Compared to other methods, our method exhibits a significant accuracy advantage, consistently exceeding 72%. It is evident that as the angular offset increases, accuracy declines. This decrease is primarily due to the greater disparity in the aspect ratio of captured watermarked images at larger angles, which amplifies scaling differences during network processing and introduces perspective transformations, thereby presenting significant challenges to both the localization network and the decoder. As illustrated in Fig. 6, at a 30° offset, localization performance weakens compared to smaller angles. Nevertheless, our method still achieves an accuracy of 72.58%, indicating that the intrinsic signal localization network and hybrid loss function effectively

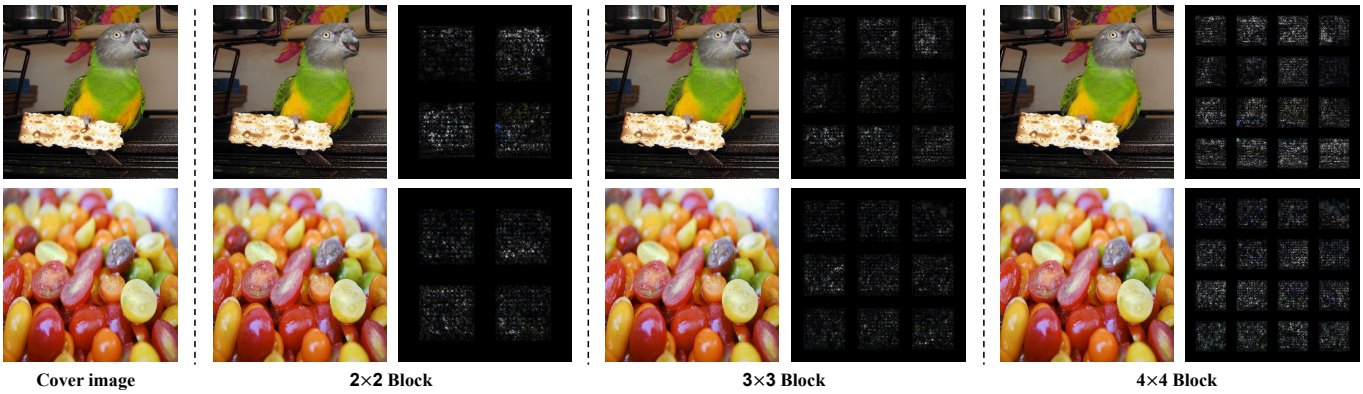


Fig. 8. Visual comparison of different application cases. The first column is the cover image, followed by the watermarked image and its corresponding watermark residual image for each case, where the residual is magnified by a factor of 5.

TABLE VII
THE EXTRACTION ACCURACY WITH DIFFERENT DEVICES

Devices	iPhone 12	Realme GT7 Pro	SAMSUNG S20FE
Lenovo XiaoXin Pro 14	99.83%	98.20%	99.96%
ENVISION G249G	97.21%	94.34%	100%
AOC 27E12HM	100%	100%	100%

TABLE VIII
THE PSNR AND SSIM VALUES OF DIFFERENT APPLICATION CASES

Cases	2×2 Block	3×3 Block	4×4 Block
PSNR (db)	36.94	35.54	34.02
SSIM	0.970	0.972	0.968

contribute to high precision decoding.

D. Full Screen-shooting Robustness

In this section, we conduct the robustness experiments under the full-shot perspective, which include variations in distance, angle, and devices.

1) Distance Test: To evaluate the robustness of full screen-shooting distortions at various distances, we fix the orientation of the monitor and adjust the shooting distances to 20 cm, 30 cm, 40 cm, 50 cm, and 60 cm. At these distances, the entire watermarked image is captured. The captured images are simply cropped to a rectangle to remove irrelevant information at the border. Due to the varying resolutions of the captured images, we resize these images to $n \times n$ before inputting them to the localization network. This process is illustrated in Fig. 7, and the decoding results are presented in Table V.

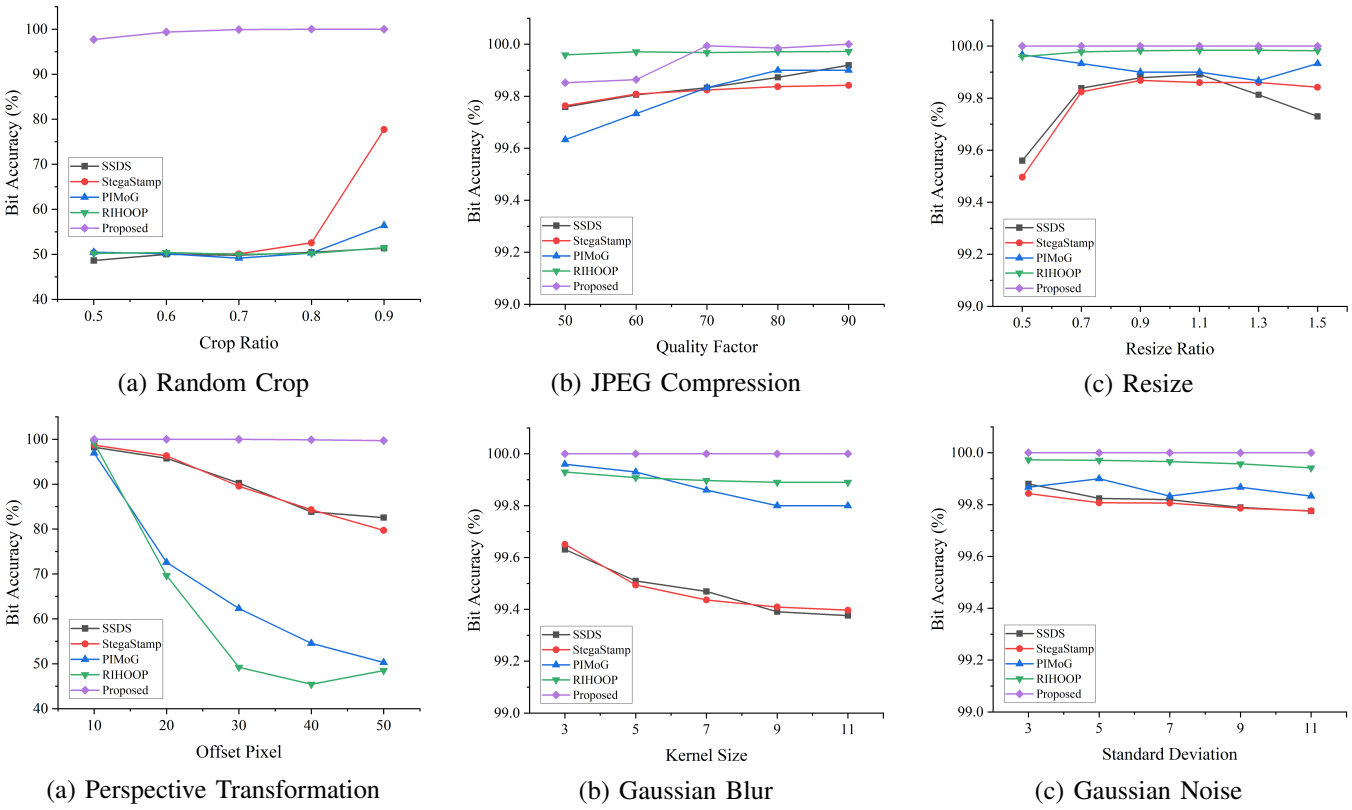
Quantitative results show that the bit extraction accuracy of the proposed method consistently exceeds 98%, demonstrating its robustness under various shooting distances. At a shorter distance of 20 cm, accuracy is lower than at 30 cm due to visible moiré patterns in the captured images (as shown in the third row of Fig. 7), yet the accuracy still reaches 99.97%. All methods perform well, and their differences are not significant. At the longer distance of 60 cm, RIHOOP [10] slightly outperforms our method by only 0.06%, a negligible difference. But in all other distances, our method achieves the highest bit extraction accuracy.

Fig. 7 shows the captured images and the binary location template images generated by the localization network. It is evident that the localization performance at different distances closely matches the ground truth. To facilitate visualization, we use the OpenCV polylines library to outline watermark block borders, clearly presenting the localization results. As shown, at a shooting distance of 20 cm, moiré patterns are visible. At different distances, the image color and brightness obviously change due to multiple complex factors such as lighting. Despite these distortions, our method demonstrates remarkable robustness.

2) Angle Test: Capturing images from various angles is common in practice, often leading to perspective distortion. To evaluate the robustness of our method in such complex scenarios, we display the watermarked images on the screen and use a mobile phone to capture them at different angles while the shooting distance is fixed at 40 cm. The shooting angle ranges from left 40° to right 40°, with the interval of 10°. The extraction results are presented in Table VI.

As shown in Table VI, our method achieves an extraction accuracy greater than 97% at all shooting angles. These results demonstrate that FPSMark is suitable for real-world screen-shooting scenarios, as shooting at more than 40° is too narrow to effectively capture the screen content. As the shooting angle increases, the degree of perspective distortion becomes more pronounced, resulting in a reduction in accuracy. Compared to other methods, our method performs better within the range of 0° to 20°. However, at larger angles, our method exhibits slightly lower performance compared to SSDS [32] and RIHOOP [10], likely due to SSDS's grayscale deviation simulation and RIHOOP's 3D rendering operations, which contribute to their improved decoding accuracy. Notably, unlike the compared methods, our extraction process avoids manually locating the accurate vertices of the encoded image and has a certain level of automatic localization capability.

3) Device Test: Various display screens have distinct resolutions, optoelectronic conversion components, and display technologies, while different capturing cameras employ diverse sensor types, resolutions, and image processing algorithms. These differences can introduce various distortions,



(a) Random Crop (b) JPEG Compression (c) Resize

(a) Perspective Transformation (b) Gaussian Blur (c) Gaussian Noise

Fig. 9. Bit Accuracy of watermark decoding under various digital distortions, including crop, JPEG compression, resize, perspective transformation, Gaussian blur, and Gaussian noise. Our method consistently achieves higher accuracy, particularly under random crop.

TABLE IX
THE EXTRACTION ACCURACY FOR PARTIAL IMAGES OF THREE CASES

Distance (cm)	24	20	16	12
Percentages	100%	88%	67%	44%
2 × 2 Block	99.69%	97.90%	60.39%	87.32%
3 × 3 Block	99.89%	99.36%	98.15%	91.49%
4 × 4 Block	99.76%	98.58%	94.10%	98.85%

TABLE X
THE ABLATION RESULTS OF THE LOCALIZATION NETWORK

LOC_{pre}	LOC_{ref}	2 × 2 Block		3 × 3 Block		4 × 4 Block	
		IoU	bit_acc	IoU	bit_acc	IoU	bit_acc
√	×	0.9144	99.56%	0.9272	99.90%	0.8926	99.58%
×	√	0.8533	74.55%	0.9081	97.76%	0.7927	86.76%
√	√	**0.9263**	**99.69%**	**0.9373**	**99.98%**	**0.9097**	**99.76%**

artifacts, and perceptual discrepancies during image capture, potentially affecting the robustness of the method. To evaluate the generalization capability of our method across different display and capturing devices, we use three screens (Lenovo XiaoXin Pro 14, ENVISION G249G, and AOC 27E12HM) to present watermarked images, and three mobile phones (iPhone 12, Realme GT7 Pro, and SAMSUNG S20FE) to capture the images. The capture distance is fixed at 40 cm, and consistent lighting conditions are maintained to minimize environmental influences on the results.

As shown in Table VII, our method achieves high extraction accuracy (over 94%) for all device pairs. This result not only demonstrates the excellent adaptability and generalizability of our method across different devices but also highlights its superiority in handling distortions and artifacts caused by device differences. Particularly, even with significant differences in device configurations, the method maintains robust performance, significantly improving watermark extraction accuracy, and exhibits broad potential for application.

E. Flexibility Evaluation

To validate the flexibility of FPSMark in meeting specific robustness requirements, we detailed three application cases. As discussed in Section III, for a 400×400 image to be protected, Case 1 utilizes a uniform 2×2 block embedding strategy, the block size is 128, the gap is 48, ensuring that when the capture percentage exceeds 76%, an entire watermark block is inevitably captured for copyright tracking. Case 2 employs a uniform 3×3 watermark block embedding strategy, with the mathematical model determining the infringement threshold to be 55%. Case 3 adopts a uniform 4×4 block strategy, enabling effective copyright authentication when the local capture percentage surpasses 41%. In this case, the block size is 70, and the gap is 24. Notably, our model is trained exclusively on 3×3 block. Inspired by the resolution scaling algorithm in [47], we generate residual artifacts through bilinear interpolation and overlay them onto the cover image block with different resolutions. During decoding, the watermarked image block is resized to 96×96 and fed into

TABLE XI
THE ABLATION RESULTS OF THE HYBRID LOSS

BCE Loss	IoU Loss	Focal Loss	2×2 Block		3×3 Block		4×4 Block	
			IoU	bit_acc	IoU	bit_acc	IoU	bit_acc
√	×	×	0.8964	98.96%	0.8989	99.84%	0.8463	98.25%
×	√	×	0.8949	98.87%	0.8971	99.83%	0.8429	97.97%
×	×	√	0.8970	99.03%	0.8996	99.87%	0.8477	98.36%
√	√	×	0.8976	99.17%	0.8999	99.87%	0.8585	98.82%
√	×	√	0.9137	99.43%	0.9286	99.95%	0.8898	99.31%
×	√	√	0.8990	99.36%	0.9265	99.94%	0.8785	99.18%
√	√	√	**0.9263**	**99.69%**	**0.9373**	**99.98%**	**0.9097**	**99.76%**

the decoder to extract the watermark. This approach not only validates the flexibility of our method but also demonstrates its generalization to images with different resolutions.

1) Visual Quality of Different Application Cases: In Table VIII, we present the PSNR and SSIM values for three different watermarking cases. Overall, as the number of watermark blocks increases, the watermark signal density rises, resulting in a corresponding decrease in PSNR values. Additionally, Fig. 8 provides a visual example of the watermarked image along with its corresponding residual, amplified by a factor of five. It is clearly observable that Case 3, with 16 watermark blocks, produces residual signals that are more pronounced than those of the other two cases, resulting in a slight reduction in visual quality. However, the visual quality of all three cases remains satisfactory. Therefore, when addressing specific partial screen-shooting robustness requirements, it is feasible to select an appropriate watermark embedding scheme by considering both robustness and visual quality. This approach ensures robustness while preventing excessive embedding that could compromise visual quality.

2) Partial Screen-shooting Robustness of Different Application Cases: To evaluate the robustness of three application cases against partial screen-shooting, we adjust the camera aspect ratio to 16:9. At a distance of 24 cm from the screen, the image is fully captured, with its width exactly filling the screen. By reducing the distance to obtain partial images with different capture percentages, Table IX illustrates the extraction accuracy of the three cases. In full-capture, all three cases achieve accuracy rates exceeding 99%, with no significant differences observed. When the distance is reduced to 20 cm, the partial capture percentage is approximately 88%. At this distance, all cases successfully captured the entire watermark block with high bit accuracy, demonstrating strong robustness to partial capture.

Further reducing the distance to 16 cm results in a capture percentage of about 67%. According to our proposed mathematical model of watermark block count and robustness, Case 1 may fail under this condition, as the captured image may not contain an entire watermark block. In Case 1, no entire watermark block is captured, and scaling distortion occurs during decoding, resulting in an accuracy of only 60.39%, indicating poor robustness. In contrast, entire watermark blocks are captured in Case 2 and Case 3, with slight accuracy degradation due to moiré patterns, but they maintain high robustness. When the distance is further reduced to 12 cm, the partial capture percentage drops to 44%, below the infringement threshold of 55% for Case 2, indicating that Case 2 may fail under this condition. To verify this, we shift the camera approximately 2 cm to the left, causing no entire watermark block to be captured in Case 2. Table IX illustrates that the accuracy of Case 2 decreased to 91.49%, representing only a minor decline compared to other distances. This minimal reduction can be attributed to two main factors: firstly, the watermark block lost very few pixels, and the decoder has a certain tolerance to cropping; secondly, the decoder is trained on 96×96 images, which minimizes scaling distortion of the watermark block. Additionally, in Case 1, our method captures nearly the entire watermark block, despite significant aspect ratio differences, with an accuracy of 87.32%, demonstrating some robustness. In contrast, Case 3 maintained an accuracy exceeding 98%, exhibiting superior overall robustness compared to Case 1 and Case 2.

In conclusion, when the partial capture percentage falls below the infringement threshold, robustness is not necessarily lost, and the actual capture position must be considered. Our proposed mathematical model, based on the worst-case capture scenario, ensures robustness under any capture position. In practical applications, the embedding scheme can be selected based on robustness and visual quality.

F. Digital Distortion Robustness

An excellent watermark framework resistant to screen-shooting should maintain robustness against digital distortions. To assess this, we comprehensively evaluate it under common digital distortions. These include random cropping, JPEG compression, resizing, perspective transformation, Gaussian blur, and Gaussian noise, each applied with varying intensity levels.

As shown in Fig. 9, our method consistently achieves high decoding accuracy under all distortion types, with results remaining above 99% in most cases. In particular, under the challenging random crop scenario where up to 50% of the image is cropped, our method still maintains an accuracy above 95%. In contrast, the benchmark methods drop to around 50%, which is close to random guessing, indicating a loss of robustness. Furthermore, in the case of perspective transformation with a 50-pixel offset, our method achieves the highest accuracy among all compared methods. This clear advantage demonstrates that our method possesses unique

robustness under digital distortion scenarios, which lays a solid foundation for its effectiveness in the screen-shooting scenario.

G. Ablation Study

In this section, we evaluate the fundamental design of the proposed method through ablation studies, particularly focusing on the intrinsic signal localization network LOC and the hybrid loss function.

1) The Effect of the LOC: We validate the effect of the proposed LOC. Table X quantitatively presents the localization performance and decoding accuracy. The results indicate that the combination of LOC_{pre} and LOC_{ref} performed best. The refinement module LOC_{ref} effectively optimizes the residuals of the localization template, refining boundaries within densely packed watermark blocks, thereby enhancing robustness, particularly in the case of 4×4 block.

2) The Effect of the hybrid loss: To validate the effectiveness of the proposed hybrid loss, we conduct a series of experiments using different loss functions. We examine various combinations of BCE, IoU, and Focal loss, as shown in Table XI. The results indicate that Focal Loss plays a significant role when a single loss is employed. Furthermore, combining of two losses outperforms individual loss functions, while integrating of all three losses achieves the highest performance. Overall, the performance differences are most pronounced in the case of 4×4 block. This suggests that imposing constraints at the pixel, region, and sample levels effectively enables the localization network to distinguish between watermark and non-watermark features within densely packed watermark blocks, thereby enhancing robustness.

VI. Conclusion and Future Work

In this paper, considering the prevalence of partial screen-shooting in real-world scenarios and the various robustness requirements, we propose FPSMark, which could enhance resistance to partial screen-shooting by embedding watermarks in multiple non-overlapping blocks within the image. By modeling the relationship between the number of watermark blocks and partial screen-shooting robustness, we provide the flexibility of our method. To achieve precise localization, we propose an intrinsic signal localization network that distinguishes between watermark and non-watermark features. Additionally, we design a hybrid loss function that constrains the localization network at the pixel, region, and sample levels, increasing the precision of watermark boundary localization and robustness of decoding. We also present various application cases tailored to different robustness requirements, demonstrating the flexibility of our method. Extensive experiments indicate that FPSMark provides excellent robustness and flexibility in partial screen-shooting, with promising applications in digital copyright protection and multimedia forensics.

Although FPSMark achieves strong robustness and flexibility, it has several limitations. First, the method is object-agnostic, focusing on uniform spatial block distribution without considering semantic saliency. As a result, partial captures of small, prominent objects may not contain complete watermark blocks, leading to degraded decoding performance. Second, the model is designed for fixed-size images. Although resizing enables broader applicability, applying the model to high-resolution images may result in significant loss of watermark details due to aggressive downscaling. Future work will aim to address these limitations to enhance adaptability and robustness in more diverse and practical scenarios.

References

[1] Z. Wang, Z. Zhang, W. Qi, F. Yang, and J. Xu, "Freqgan: Infrared and visible image fusion via unified frequency adversarial learning," *IEEE Transactions on Circuits and Systems for Video Technology*, 2024.

[2] K. Gu, H. Liu, Y. Liu, J. Qiao, G. Zhai, and W. Zhang, "Perceptual information fidelity for quality estimation of industrial images," *IEEE Transactions on Circuits and Systems for Video Technology*, 2024.

[3] F. Yao, H. Zhang, Y. Gong, Q. Zhang, and P. Xiao, "A study of enhanced visual perception of marine biology images based on diffusion-gan," *Complex & Intelligent Systems*, vol. 11, no. 5, pp. 1–20, 2025.

[4] W. Sun, J. Zhou, Y. Li, M. Cheung, and J. She, "Robust high-capacity watermarking over online social network shared images," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 3, pp. 1208–1221, 2020.

[5] Z. Jia, H. Fang, and W. Zhang, "Mbrs: Enhancing robustness of dnn-based watermarking by mini-batch of real and simulated jpeg compression," in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 41–49.

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

[7] M. Tancik, B. Mildenhall, and R. Ng, "Stegastamp: Invisible hyperlinks in physical photographs," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2117–2126.

[8] E. Wengrowski and K. Dana, "Light field messaging with deep photographic steganography," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1515–1524.

[9] H. Fang, D. Chen, F. Wang, Z. Ma, H. Liu, W. Zhou, W. Zhang, and N. Yu, "Tera: Screen-to-camera image code with transparency, efficiency, robustness and adaptability," *IEEE Transactions on Multimedia*, vol. 24, pp. 955–967, 2021.

[10] J. Jia, Z. Gao, K. Chen, M. Hu, X. Min, G. Zhai, and X. Yang, "Rihoop: Robust invisible hyperlinks in offline and online photographs," *IEEE Transactions on Cybernetics*, vol. 52, no. 7, pp. 7094–7106, 2020.

[11] H. Fang, Z. Jia, Z. Ma, E.-C. Chang, and W. Zhang, "Pimog: An effective screen-shooting noise-layer simulation for deep-learning-based watermarking network," in *Proceedings of the 30th ACM International Conference on Multimedia*, 2022, pp. 2267–2275.

[12] J. Jia, Z. Gao, D. Zhu, X. Min, G. Zhai, and X. Yang, "Learning invisible markers for hidden codes in offline-to-online photography," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 2273–2282.

[13] M. He, B. Feng, Y. Guo, J. Weng, and W. Lu, "Camera-shooting resilient watermarking on image instance level," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 34, no. 11, pp. 10 874–10 887, 2024.

[14] L. Zhu, Y. Fang, Y. Zhao, Y. Peng, J. Wang, and J. Ni, "Lite localization network and due-based watermarking for color image copyright protection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 34, no. 10, pp. 9311–9325, 2024.

[15] S. Kumar, A. K. Yadav, A. Gupta, and P. Kumar, "Rgb image steganography on multiple frame video using lsb technique," in *2015 International Conference on Computer and Computational Sciences*. IEEE, 2015, pp. 226–231.

[16] T. Zong, Y. Xiang, I. Natgunanathan, S. Guo, W. Zhou, and G. Beliakov, "Robust histogram shape-based method for image watermarking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 5, pp. 717–729, 2014.

[17] H. Fang, D. Chen, Q. Huang, J. Zhang, Z. Ma, W. Zhang, and N. Yu, "Deep template-based watermarking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 4, pp. 1436–1451, 2020.

[18] A. Pramila, A. Keskinarkaus, and T. Seppänen, "Toward an interactive poster using digital watermarking and a mobile phone camera," *Signal, Image and Video Processing*, vol. 6, no. 2, pp. 211–222, 2012.

[19] K. Fares, A. Khaldi, K. Redouane, and E. Salah, "Dct & dwt based watermarking scheme for medical information security," *Biomedical Signal Processing and Control*, vol. 66, p. 102403, 2021.

[20] W. C. Chu, "Dct-based image watermarking using subsampling," *IEEE Transactions on Multimedia*, vol. 5, no. 1, pp. 34–38, 2003.
[21] H. Tian, Y. Zhao, R. Ni, L. Qin, and X. Li, "Ldft-based watermarking resilient to local desynchronization attacks," *IEEE Transactions on Cybernetics*, vol. 43, no. 6, pp. 2190–2201, 2013.
[22] H. Joseph and B. K. Rajan, "Image security enhancement using dct & dwt watermarking technique," in *2020 International Conference on Communication and Signal Processing*. IEEE, 2020, pp. 0940–0945.
[23] N. C. Onn, S. N. Avivah, M. M. Alam, and A. Ahmad, "Secure and effective image integrity and copyright protection using two layer authentications with integer wavelet transform," *Int. J. Adv. Comput. Inform*, vol. 1, pp. 13–27, 2025.
[24] C. Fikri, F. A. Nugraha, B. Apriyansyah, and M. Fakhreldin, "Dual watermarking based on human visual characteristics with iwt-svd," *Int. J. Adv. Comput. Inform*, vol. 1, pp. 1–12, 2025.
[25] L. C. Gen, S. N. Avivah, and A. J. M. Muzahid, "Image watermarking for ensuring image integrity and robust copyright protection based on discrete wavelet transform," *Int. J. Adv. Comput. Inform*, vol. 1, pp. 28–38, 2025.
[26] Ridwan and N. K. Kabir, "Robust color image watermarking using dual embedding via schur decomposition," *Int. J. Adv. Comput. Inform*, vol. 1, pp. 39–47, 2025.
[27] J. Zhu, R. Kaplan, J. Johnson, and L. Fei-Fei, "Hidden: Hiding data with deep networks," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 657–672.
[28] R. Ma, M. Guo, Y. Hou, F. Yang, Y. Li, H. Jia, and X. Xie, "Towards blind watermarking: Combining invertible and non-invertible mechanisms," in *Proceedings of the 30th ACM International Conference on Multimedia*, 2022, pp. 1532–1542.
[29] Y. Liu, M. Guo, J. Zhang, Y. Zhu, and X. Xie, "A novel two-stage separable deep learning framework for practical blind watermarking," in *Proceedings of the 27th ACM International Conference on Multimedia*, 2019, pp. 1509–1517.
[30] B. Wang, Y. Wu, and G. Wang, "Adaptor: Improving the robustness and imperceptibility of watermarking by the adaptive strength factor," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 11, pp. 6260–6272, 2023.
[31] L. Fu, X. Liao, J. Guo, L. Dong, and Z. Qin, "Waverecovery: Screen-shooting watermarking based on wavelet and recovery," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1–1, 2024.
[32] Y. Li, X. Liao, and X. Wu, "Screen-shooting resistant watermarking with grayscale deviation simulation," *IEEE Transactions on Multimedia*, vol. 26, pp. 10 908–10 923, 2024.
[33] H. Guo, Q. Zhang, J. Luo, F. Guo, W. Zhang, X. Su, and M. Li, "Practical deep dispersed watermarking with synchronization and fusion," in *Proceedings of the 31st ACM International Conference on Multimedia*, 2023, pp. 7922–7932.
[34] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7132–7141.
[35] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 586–595.
[36] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2223–2232.
[37] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang *et al.*, "Deep high-resolution representation learning for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 10, pp. 3349–3364, 2020.
[38] X. Qin, Z. Zhang, C. Huang, M. Dehghan, O. R. Zaiane, and M. Jagersand, "U2-net: Going deeper with nested u-structure for salient object detection," *Pattern Recognition*, vol. 106, p. 107404, 2020.
[39] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*. Springer, 2015, pp. 234–241.
[40] P.-T. De Boer, D. P. Kroese, S. Mannor, and R. Y. Rubinstein, "A tutorial on the cross-entropy method," *Annals of Operations Research*, vol. 134, pp. 19–67, 2005.
[41] G. Máttyus, W. Luo, and R. Urtasun, "Deeproadmapper: Extracting road topology from aerial images," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3438–3446.
[42] T.-Y. Ross and G. Dollár, "Focal loss for dense object detection," in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2980–2988.
[43] S. Xie and Z. Tu, "Holistically-nested edge detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1395–1403.
[44] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer, 2014, pp. 740–755.
[45] M. J. Huiskes and M. S. Lew, "The mir flickr retrieval evaluation," in *Proceedings of the 1st ACM International Conference on Multimedia Information Retrieval*, 2008, pp. 39–43.
[46] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019.
[47] T. Bui, S. Agarwal, and J. Collomosse, "Trustmark: Universal watermarking for arbitrary resolution images," *arXiv preprint arXiv:2311.18297*, 2023.

Mingyue Chen received the M.S. degree from the College of Computer Science and Electronic Engineering, Hunan University, Changsha, China, in 2021. She is currently pursuing the Ph.D. degree with the College of Cyber Science and Technology, Hunan University. Her research interests include the data hiding, multimedia security, machine learning.

Xin Liao (Senior Member, IEEE) received the B.E. and Ph.D. degrees in information security from the Beijing University of Posts and Telecommunications in 2007 and 2012, respectively. He is currently a Professor and a Doctoral Supervisor with Hunan University, China. He worked as a Post-Doctoral Fellow with the Institute of Software, Chinese Academy of Sciences, and also 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, USA. His current research interests include multimedia forensics, steganography, and watermarking. He is serving as an Associate Editor for the IEEE Signal Processing Magazine.

Han Fang received the B.S. degree from Nanjing University of Aeronautics and Astronautics (NUAA) in 2016 and the Ph.D. degree from the University of Science and Technology of China (USTC) in 2021. Currently, he is a Research Fellow with the National University of Singapore. His research interests include image watermarking, information hiding, and adversarial machine learning.

Jinlin Guo received the B.S. degree from Central South University, Changsha, Hunan China, in 2006 and the M.S. degree from National University of Defense Technology, Changsha, Hunan China, and Ph.D. from Dublin City University, Dublin, Ireland. He is currently a Professor with National University of Defense Technology, China. His research interests are machine learning and multimedia information processing.

Yanxiang Chen received the B.Sc. and M.Sc. degrees in electronic information engineering from the Hefei University of Technology, Hefei, China, in 1993 and 1996, respectively, and the Ph.D. degree in signal and information processing from the University of Science and Technology of China, Hefei, China, in 2004. From 2006 to 2008, she was a Visiting Scholar with the University of Illinois at Urbana-Champaign, Champaign, IL, USA, and with the National University of Singapore, Singapore, from 2012 to 2013. She is currently a Professor with the School of Computer Science and Information Engineering, Hefei University of Technology. Her research interests include multimodal signal processing, pattern recognition, and machine learning.

Xiaoshuai Wu received the B.S. degree from the Nanyang Institute of Technology in 2019 and the M.S. degree from Hangzhou Dianzi University in 2022. He is currently pursuing the Ph.D. degree with Hunan University. His research interests include data hiding and AI security.