WaveRecovery: Screen-Shooting Watermarking Based on Wavelet and Recovery

Linbo Fu, Xin Liao, *Senior Member, IEEE*, Jinlin Guo, Li Dong, and Zheng Qin, *Member, IEEE*

***Abstract*— The demand for resilient watermarking technology in the context of the screen-shooting scenario is steadily on the rise. The principal objective of this technique is to embed messages into the cover image, with the ability to effectively recover the message from the screen-captured image at the extraction end. However, current watermarking methods result in low visual quality watermarked images and are insufficiently robust in screen-shooting scenarios. This is mainly because they only utilize spatial domain information during embedding, and they do not consider the impact of noise that introduced during screen capturing. This paper introduces an innovative network framework, including the wavelet domain concatenation and recovery mechanism, to overcome the dual challenges encountered in robust watermarking, namely visual fidelity and robustness. For fidelity, we present a cascade network operating in the wavelet domain. This network excel at detecting watermark information in the wavelet domain. This capability makes it more sensitive to high and low-frequency details. Discrete wavelet transform can make CNN focus on different frequency characteristics, and the use of discrete inverse wavelet transform in upsampling can make the information high fidelity. As a result, it can more accurately identify and preserve critical visual details in this frequency domain, leading to an overall enhancement in visual quality. For robustness, a recovery network is specifically designed to mitigate the influence of noise introduced during screen-shooting on watermark information extraction. Experimental validation of our proposed method substantiates its effectiveness in significantly enhancing the visual quality and the accuracy of the watermarked images.**

***Index Terms*— Robust watermarking, screen-shooting scenarios, wavelet domain, recovery network.**

I. Introduction

THE domain of digital watermarking emerges as a powerful instrument in copyright protection. The spread of contemporary digital content, encompassing images, videos, etc. They have exacerbated the pressing issue of unauthorized reproduction and widespread dissemination. In response, digital watermarking distinguishes itself as a concealed yet potent protective mechanism, offering copyright proprietors a means to embed distinctive identifiers within their digital assets. Such identifiers serve the vital purpose of easily detecting unauthorized reproductions. Robustness assumes a paramount and inescapable role within the watermarking system design and implementation framework. Traditional digital watermarking has prioritized robustness against digital editing distortions [1], [2], [3], given that the prevailing avenues of information leakage are primarily situated within the electronic channels, including but not limited to JPEG compression [4], Gaussian noise [5]. Of course, these problems exist in video [6] media as well. The advent of digital devices has substantially altered this landscape. Contemporary mobile devices, for instance, facilitate the effortless acquisition of high-quality reproductions of digital content through on-the-fly image capture. Robust watermarking is also used to protect some other objects of property rights. On the one hand, in the field of resisting the use of facial forgery models, watermarks can be used as noise to interfere with the output of those GAN models, effectively realizing the defense of GAN forgery models. Qiao et al. [7] proposed a watermark expansion scheme, which mainly includes inheritance, defense and constraint steps. Their method can resist new forgery models. On the other hand, there are also some studies on how to protect the intellectual property of the model, that is, model watermarking. Malicious attackers may steal the trained GAN model to infringe the IP of the real model owner. In order to solve this problem, Qiao et al. [8] proposed a model watermarking framework for GAN model. By combining the watermark label and the verification image, the trigger set is established. Only by relying on the correct watermark label, the model owner can successfully trigger the synthetic watermark for IP protection.

However, the screen capture process often inflicts severe and worse consequences upon the embedded watermark signal. As shown in Fig. 1, the pirated content comes from the camera shooting from the screen, which is no longer the previous digital transmission channel. Specifically, in Fig. 1 (a), the watermark information is fused with the cover image to complete the watermark embedding. Subsequently, the watermarked image is displayed on the screen, as shown in Fig. 1 (b). Unlike traditional watermarks, copyright infringement occurs during the phone's screen capture, as shown in Fig. 1 (c), rather than transmitting via electronic channel. Notably, the enhanced clarity of images captured by cameras allows more watermarks to be retained. When end-users

Received 1 September 2024; revised 19 November 2024; accepted 27 November 2024. Date of publication 2 December 2024; date of current version 7 April 2025. This work was supported in part by the National Natural Science Foundation of China under Grant U22A2030 and Grant 62171244, in part by the National Key Research and Development Program of China under Grant 2024YFF0618800 and Grant 2022YFB3103500, in part by Hunan Provincial Funds for Distinguished Young Scholars under Grant 2024JJ2025, and in part by Ningbo Natural Science Foundation-Young Doctoral Innovation Research Project under Grant 2022J080. This article was recommended by Associate Editor S. Zhu. *(Corresponding author: Xin Liao.)*

Linbo Fu, Xin Liao, and Zheng Qin are with the College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China (e-mail: linbofu21@hnu.edu.cn; xinliao@hnu.edu.cn; zqin@hnu.edu.cn).

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

Li Dong is with the Faculty of Electrical Engineering and Computer Science, Ningbo University, Ningbo 315211, China (e-mail: dongli@nbu.edu.cn).

Digital Object Identifier 10.1109/TCSVT.2024.3510355

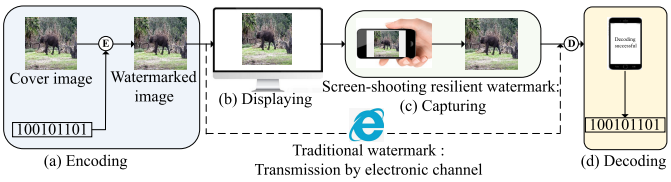

Fig. 1. Screen-shooting resilient watermark. It includes encoding, displaying, capturing, and decoding stages, demonstrating each step in preserving and recovering watermarks against distortions encountered during screen-shooting.

engage in screen capture activities to procure digital content, the original image or file frequently undergoes varying degrees of distortion, resulting in irreversible modifications to the concealed watermark information. Such alterations can profoundly impede the extraction process of the watermark, potentially rendering it entirely imperceptible or unrecoverable. Consequently, there is an urgent need for a robust watermarking framework tailored specifically to screen capture scenarios.

In the context of conventional watermarking schemes tailored to screen capture resilience, their methodology typically involves identifying robust image feature points or regions within the image [9], [10], [11], complemented by applying generic watermark embedding techniques to enhance robustness. In recent years, there has been a notable shift towards integrating deep learning methodologies into watermarking architectures, with several seminal contributions in this domain [12]. At the core of deep learning-based watermarking lies an end-to-end [13] auto-encoder-like framework comprising an encoder, a decoder, and an interposed noise layer. Hidden [14] is the first to use this architecture, and this method also has strong robustness. Later on, there were many innovative works on the noise layer [15], [16], [17] in screen-shooting resilient watermarking. Until 2024, Li et al. [18] proposed a novel screen-shooting resilient watermarking[16], which we called SSRGDS. They proposed a gray-scale inversion method to simulate the distortion caused by the camera 's shooting screen, especially when the shooting angle offset is relatively large. They use this way of generating noise as a noise layer to train the watermark model. The noise layer constitutes a modest component within the broader framework of robust watermarking. The noise layer, encoder, and decoder form a complete framework. The key to ensuring good visual quality for watermarked images resides in the design of a proficient encoder. An effective encoder must adeptly facilitate the seamless watermarking embedding into the cover image while preserving its integrity. This necessitates the ability of the encoder to strike a delicate balance between the objectives of data embedding and minimizing the distortion caused by watermarking. Furthermore, an essential aspect of encoder design entails preserving perceptual consistency acknowledging the human visual system's sensitivity to image attributes. In practice, the watermark information is strategically embedded within regions least susceptible to human visual detection, thereby mitigating conspicuous visual artifacts. This strategic approach ensures that the watermarked image remains highly recognizable during extraction without incurring perceptible degradation in visual quality resulting from the embedding process.

In previous studies, the encoder component was often directly adapted from semantic segmentation models, exemplified by architectures like U-net [19] and its counterparts. For example, HiDDeN [14], Stegastamp [13], RIHOOP [15], PIMoG [16] use the same network structure as the encoder. Unfortunately, the way to use a universal encoder remains suboptimal as it fails to seamlessly integrate the watermark embedding process, posing challenges in effectively concealing the watermark within regions of the cover image that evade ready human visual detection. Furthermore, the encoders for these methods primarily rely on spatial information, which can inadvertently engender conspicuous artifacts in the watermarked image post-embedding. These artifacts, in turn, engender adverse effects on the overall effectiveness of the algorithm, resulting in reduced robustness.

Henceforth, the foremost obstacle in realizing a watermarking algorithm resilient to screen capture is the formulation of a skilled encoder. In pursuit of this objective, we combine CNN with the wavelet domain to embed the watermark. Wavelet transform can provide high fidelity low-frequency and high-frequency feature maps for convolution operations. This approach combines the adaptive embedding capabilities inherent to deep learning methodologies with the wealth of cover image information collected from frequency domain transformations. The combination of these principles serves as the cornerstone for augmenting the embedded watermark's visual fidelity and robustness. Since the discrete wavelet transform is a small region, it can easily fit the sharp changes of the signal. Combining wavelet transform with CNN to design an encoder can more accurately control the embedding and reduce the loss in the feature extraction process compared to convolution or pooling. The training parameters will not be reduced.

Previous work on screen-shooting watermarking only focused on designing noise layers, such as RIHOOP [15]. My contributions focus on designing innovative encoders and decoders. With the harmonious interplay of the thoughtfully designed encoder and decoder, the watermarking framework posited herein attains an equilibrium, concurrently improving visual fidelity and robustness. To substantiate the validity of our conceptual basis, we devise a wavelet domain cascade encoder network to serve as the encoder within our watermarking framework. This network exhibits a distinctive architecture incorporating multiple multi-level cascaded blocks, each sharing an identical structural blueprint. Each multi-level cascade block encompasses three integral constituents, particularly emphasizing the multi-level cascade structure. Within this structural framework, we integrate wavelet transformation and wavelet inverse transformation. During the training phase, the spatial and frequency domain constituents of the watermark and the cover image are fused. Extensive experiments have proved our conceptual framework's soundness and underscored the effectiveness of the encoder we conceived. In comparative assessments against alternative methodologies, our wavelet domain cascade encoder network exhibits pronounced advantages, particularly in robustness and visual fidelity. Discrete wavelet transform can separate the detail information and contour information of the image. The high-frequency part contains details such as edges and textures, while the low-frequency part contains the overall contour

and structure of the image. This separation allows CNN to focus more on features of different frequencies. For example, in our designed wavelet domain cascade encoder network, the convolutional layer close to the DWT-2D layer can focus on processing the low-frequency region of the image, which not only improves the efficiency of feature extraction, but also reduces the loss of features during downsampling. In addition, the wavelet transform has the characteristics of decomposing the image into multi-resolution, which can decompose the image at different scales and obtain multi-scale features. Inputting these multi-scale features into CNN can better integrate information of different scales and improve the perception ability and recognition accuracy of the model. Our model uses the Discrete Wavelet Transform-2D (IDWT-2D) layer, which can make the image features high fidelity during upsampling. We craft a recovery decoder network to fulfill the role of the decoder within our framework. This decoder assumes the pivotal task of mitigating, to a certain extent, the noise introduced during the traversal of the watermark through a communication channel spanning from the display to the camera. By eliminating these noise artifacts, the overall robustness of the watermark is correspondingly enhanced.

Our Contributions: The contributions of the proposed scheme are shown as follows:

1) We propose a new insight into designing watermarking for screen-shooting resilience by training a deep learning network in the wavelet domain to embed the watermark. Wavelet transform can decompose feature maps high fidelity. It can be proved by PSNR and SSIM in the experiment that it can improve the quality of watermarking image.
2) We architect a wavelet domain cascade encoder network explicitly intended for watermark embedding. This network undergoes rigorous training through the fusion of spatial domain and frequency domain constituents of the watermark. This training scheme affords a marked enhancement in visual quality.
3) We propose a recovery decoder network as the decoder component. This decoder performs the pivotal function of noise elimination, stemming from the inherent noise introduced during the transit of the watermark through a communication channel spanning from the display to the camera. This noise attenuation significantly augments the watermark's robustness.

The rest of this article is arranged as follows. Section II discusses the techniques related to robust watermarking. At the same time, some training work in the frequency domain is briefly described. Section III proposes how to construct a robust watermarking model. It includes two innovative networks. One is an encoder network based on a wavelet domain cascade encoder network. The other is a decoder network based on a recovery decoder network. The experimental results and analysis are shown in Section IV. Finally, the conclusion is drawn in Section V.

II. Related Work

A. Traditional Robust Watermarking

The domain of robust digital watermarking has been the subject of extensive and enduring research topic. The screen capture process, characterized by multiple uneven sampling and quantization operations [20], introduces a huge challenge. Furthermore, the distortion from the disparate configurations of display devices [21] and camera equipment presents a huge obstacle. Qualitative analysis of this multifaceted process remains a formidable undertaking. Recognizing the potential of geometric invariance among feature points within cover images [5], [22], specific traditional methods have endeavored to partition regions utilizing three such feature points [23], [24], subsequently embedding watermarks within these specific regions. Feature points derived from the intermediate frequency region within the natural logarithmic magnitude frequency domain have attracted attention [25]. Fang et al. [26], for instance, have advocated the use of an enhanced scale-invariant feature transformation to pinpoint regions suitable for watermark embedding. This method is a representative watermarking framework for anti-screen shooting in traditional methods. Huang et al. [2] proposed a spread spectrum scheme with adaptive embedding strength and a differential quantization scheme with adaptive quantization threshold.

B. Deep Learning-Based Robust Watermarking

Several prior studies have explored strategies for enhancing the resilience of digital watermarks within cover images against distortion, such as incorporating watermarks into single-frame synchronization mechanisms [27] and examining correlations among neighboring frames [28]. To comprehensively address the challenges posed by complex distortion scenarios, specific algorithms have exploited the creation of a huge dataset [29] designed to withstand screen capture-induced perturbations. To ensure the applicability of these watermarking techniques across various imaging pipelines, Tancik et al. [13] have harnessed noise modeling as a means to simulate common distortions encountered in screen capture environments. The idea of distortion simulation in [13] has influenced the following researches [30], [31], [32]. In a similar vein, Jia et al. [15] have delved into optimizing noise layers by using 3D modeling. Conversely, Fang et al. [16] have advocated for the selective disregard of less influential noise components and introduced a method to simulate the moiré patterns generated by the screen. Ge et al. [33] proposed a document image anti-screenshot watermarking scheme based on deep neural network. They designed end-to-end neural networks, designed encoders and decoders, and introduced a series of distortions as noise layers. At the same time, they designed a background sensitive loss and a lpips loss to improve image quality, and also proposed an intensity factor adjustment strategy to train the watermark model. Cao et al. [34] proposed a general screenshot robust image watermarking scheme. They introduced the channel attention mechanism into the DCT domain and used the noise layer to guide the watermark model. Guo et al. [35] proposed a screenshot elastic image watermark of a dual-branch network. The weight of the dual-branch encoder conforms to the Gaussian distribution. In order to improve the performance, they use the double frame alternating fusion mechanism and the JND loss function of the residual image. The decoder of the scheme integrates a quality enhancement module (QEM) for denoising.

C. Deep Learning in the Frequency Domain

Conducting network training in the frequency domain offers a strategic advantage [36], [37], [38], [39], [40], as it empowers the encoder to allocate increased attention to the frequency domain constituents of the watermarking signal, as well as the fused features originating from the cover image. An influential precursor in this realm is the work of Li et al. [41], who introduced a frequency domain neural network for image super-resolution, merging the frequency domain with neural network paradigms and inspiring subsequent research endeavors. Notably, Xu et al. [42] delved into exploring implicit biases intrinsic to the training dynamics of neural networks. Furthermore, Lu et al. [43] ventured into applying frequency-domain learning to robust watermarking, with the primary objective of elevating the visual quality of images. Concurrently, Zheng et al. [44] introduced a degradation model designed to characterize images afflicted by moiré patterns. Considering the spatial redundancy often inherent in convolutional neural network (CNN) filters, Liu et al. [45] devised a dynamic pruning scheme operating in the frequency domain to exploit spatial correlations. Xu et al. [46] introduced the conditional normalizing flow to model the distribution of the redundant high-frequency component with the condition of the container image. Combined with the reversible network to adjust network parameters at different distortion levels, the robustness of the steganographic framework is enhanced. Their approach substantially surpassed prior methods operating in the spatial domain.

III. Methodology

A. Overview

This section describes the robust watermarking model, an innovative combination of the wavelet domain cascade encoder network and the recovery decoder network. The proposed watermarking model consists of three pivotal modules: Wavelet domain cascade encoder network (WCN), Recovery decoder network (RDN) and Screen shooting noise simulator. The input of the framework proposed in our paper is a six-channel tensor of 400×400 pixels. The generation process of this tensor is as follows: an RGB color cover image I_c of any size and a 100bits binary string w, which is the watermark information to be embedded. The cover image is converted into a 400×400 size tensor through a resizing operation. The watermark information w can be reshaped into a $50 \times 50 \times 3$ tensor after being converted by a fully connected layer and then upsampled to generate a $400 \times 400 \times 3$ tensor. When converting 100 bits of information, first input the 100 bits of information into a fully connected layer, and the output of this fully connected layer is 7500. This is because the result of $50 \times 50 \times 3$ is exactly 7500. At this point, we get a one-dimensional tensor with a total number of 7500. Then, we rearrange these 7500 data, in the way that each feature map is 50×50, for a total of 3 feature maps. In this way, we get a $50 \times 50 \times 3$ tensor. This tensor is concatenated with I_c to generate a six-channel tensor. The encoder network finally generates a one-channel residual image, which is added to I_c to obtain I_e.

As shown in Fig. 2, the specific workflow of the watermarking framework is as follows. The encoder receives the cover image and the watermark information to be embedded, outputs a residual image, and then fuses the residual image with the original cover image, that is, adding the corresponding pixels to generate the watermarked image I_e. In the screen-shooting scenario, the watermarked image will introduce noises before extracting the watermark information. These noises are caused by the camera shooting the watermarked image displayed on the screen. The noise layer is used to simulate introducing noise into watermarked images taken by a camera. The noise layer includes perspective transformation, motion blur, color manipulation, Gaussian noise, and JPEG compression. Based on the watermarked image I_e, a series of watermarked images with different types of distortion, such as I_{ej}, where $j \in \{A, B, C, D, E\}$ are generated through the noise layer, which is used to simulate typical distortion characteristics in screen-shooting scenarios. Among them, j represents the watermarked image with different types of noise added. For example, I_{eA} represents a watermarked image with perspective distortion. The noise simulator operates by linearly superimposing noise onto the watermarked image, simulating the distortion characteristics typical of screen capture scenarios. The image intended for decoding by the decoder module is designated as I_{eE}. It is worth noting that our decoder incorporates noise reduction capabilities, and the ultimate image intended for extracting by the genuine decoding module is denoted as I'_e. In the overarching framework, the encoder and decoder constitute the crux of the methodology.

B. Deep Learning Network Based on Discrete Wavelet Transform

Discrete wavelet transform (DWT) can extract different frequency components of the image. Combined with deep learning networks, features extracted by DWT can be used for some tasks such as classification, segmentation, or generation. Discrete wavelet transform can decompose the image data into four components, including one low-frequency component and three high-frequency components. The low-frequency component contains most of the content of the image. The high-frequency components include horizontal components, vertical components and oblique components. These high-frequency components contain subtle features of the image data. When an image is decomposed into low-frequency components, this low-frequency component can continue to be decomposed using discrete wavelet transform. For example, when the resolution of an image is $N \times N$, four $(N / 2) \times (N / 2)$ components are generated after discrete wavelet transform. As long as N can continue to be divisible by 2, the same method can be used to continue to decompose. Therefore, when combining DWT with deep learning networks, DWT can often be used to perform feature map downsampling operations [47]. Due to the high fidelity nature of the inverse process of DWT, the network uses DWI and IDWT for up and down sampling, which usually retains more real details. And due to it can sample multiple times and is high fidelity, the network composed of it can better control the image details.

We introduce the way of using DWT to design deep learning network, that is, designing DWT and IDWT layers. The key

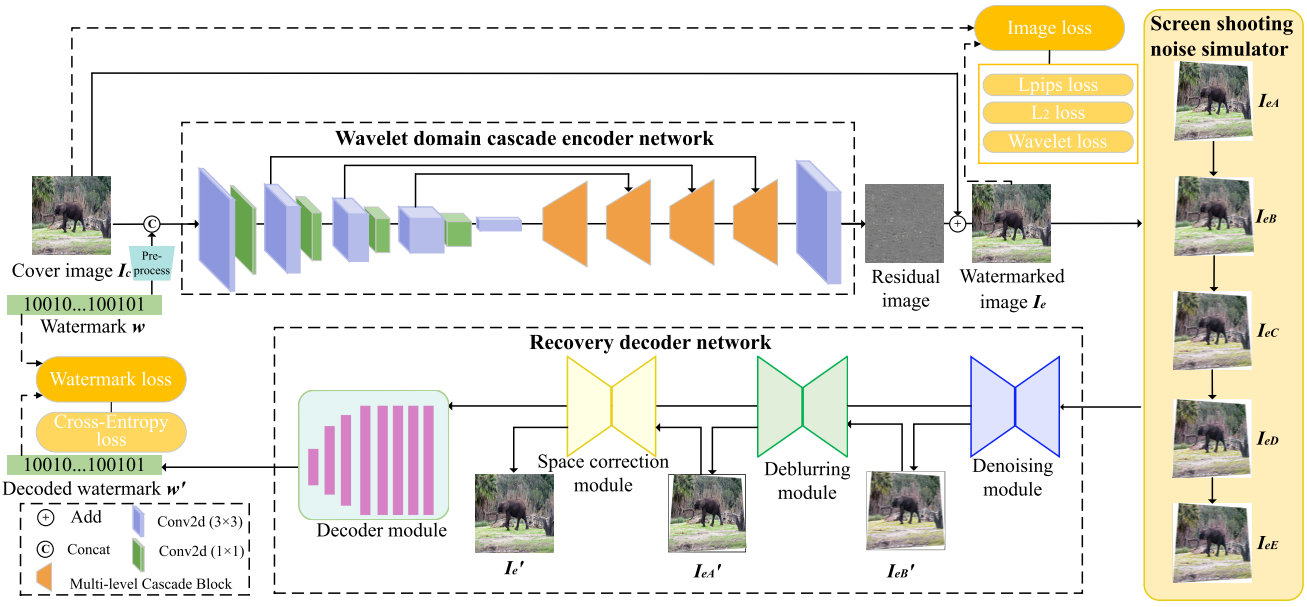

Fig. 2. Schematic overview of the WaveRecovery method for robust watermarking in screen-shooting contexts: This figure presents a detailed framework of the proposed method, showcasing the intricate process flow from cover image preparation through watermark embedding to the final watermark decoding.

problem is the forward propagation and reverse propagation of data. For one-dimensional signal s, DWT decomposes it into its low-frequency component $s1$ and high-frequency component $d1$.

$$\begin{cases} s_{1k} &= \sum_j l_{j-2k} s_j, \\ d_{1k} &= \sum_j h_{j-2k} s_j \end{cases} \tag{1}$$

where l and h are the low-pass and high-pass filters of an orthogonal wavelet. According to the equation (1), the DWT layer consists of filtering and downsampling. Using the IDWT layer, s can be reconstructed from $s1$, $d1$ as follows.

$$s_j = \sum_k (l_{j-2k} s_{1k} + h_{j-2k} d_{1k}) \tag{2}$$

In order to simplify the expression, the (1) and (2) are rewritten as (3), where L and H represent the two filters described above.

$$\begin{cases} s_1 = Ls, d_1 = Hs, \\ s = L^T s_1 + H^T d_1 \end{cases} \tag{3}$$

The discrete wavelet transform of the above one-dimensional signal can be extended to the two-dimensional signal. A typical representative of two-dimensional signals is image data. Therefore, the two-dimensional wavelet transform is shown in (4).

$$\begin{cases} X_H = LXL^T, \\ X_{lh} = HXL^T, \\ X_{hl} = LXH^T, \\ X_{hh} = HXH^T \end{cases} \tag{4}$$

The two-dimensional inverse discrete wavelet transform can be expressed as (5).

$$X = L^T X_{ll} L + H^T X_{lh} L + L^T X_{hl} H + H^T X_{hh} H \tag{5}$$

In order to enable the discrete wavelet transform and inverse transform to be trained in the deep neural network, the high-pass filter and the low-pass filter are set as trainable parameters, as shown in (6).

$$\frac{\partial s_1}{\partial s} = L^T, \frac{\partial d_1}{\partial s} = H^T \tag{6}$$

It can be seen from (6) that discrete wavelet transform and inverse transform are a series of trainable parameter matrices in deep learning networks, which have a lot in common with the training method of convolution kernel. Therefore, like convolutional neural networks, the design of discrete wavelet layers in neural networks is very flexible. Combined with the existing deep neural network model research, different types of networks with different characteristics can be built. Based on this principle, we built the wavelet domain cascade encoder network. The implementation methods of other frequency domains, such as discrete cosine domain, are completely different from those of wavelet domain, and they cannot closely combine domain transformation structure with convolutional neural network. Therefore, it is impossible to directly use other domains to replace wavelet domain network. In our work, we choose the discrete wavelet domain.

C. Wavelet Domain Cascade Encoder Network

The encoder network is engineered to seamlessly integrate the message into the cover image, with utmost attention to preserving the perceptual consistency between the original cover image and the resultant watermarked images.

Within the context of this research, a pioneering network architecture is conceived to process a six-channel input of dimensions 400 × 400 pixels. This input comprises a three-channel cover image intricately fused with a pre-processed three-channel message. To harmonize with the inherent characteristics of the cover image, the encoder network yields a one-channel residual image as its ultimate

output. A key of our approach is the segmentation of the feature map of the cover image through the adept application of the discrete wavelet transform. This strategy facilitates the harmonious integration of the spatial domain component of the watermark with its frequency domain counterpart. The overarching goal is to enhance the model's responsiveness to the high and low-frequency nuances manifest in the cover image. An integral facet of our strategy revolves around training the watermark embedding process in the frequency domain, thereby addressing a limitation associated with confining training to the spatial domain. This strategic shift enables the network to control the distribution of different frequency components within the image precisely. The encoder architecture is a cascade network operating within the wavelet domain. This structural choice enables the fusion and training of the watermark's spatial and frequency domain constituents through the agency of the wavelet transform.

Specifically, as illustrated in Fig. 2, the tensor enters the down-sampling stage after inputting into the model, where $l \in \{1, 2, 3, 4\}$ represents the number of down-sampling stages. Each down-sampling stage will generate feature maps with different dimensions. Let $F_{CNN}(x, y, l)$ denote the feature map generated by the CNN module for a specific pixel (x, y) at downsampling level l. After multiple downsampling iterations, these feature maps will be integrated into different cascade modules. The fusion and transformation operations performed by the cascade module can be defined as $F_{cas}(x, y, l) = \text{CasMod}(F_{CNN}(x, y, l), F_{cas}(x, y, l-1))$. The variable l signifies the cascade module's level in this expression. For each cascade structure, the fusion of spatial and frequency domain elements can be succinctly articulated as follows:

$$F_{cas}(x, y, i+1) = [F_{CNN}(x, y, i) \copyright F_{DWT}(x, y, i)] \copyright F_{IDWT}(x, y, i) \quad (7)$$

where F_{CNN} and F_{DWT} correspond to the spatial and frequency domain components of the cascade feature map at coordinates (x, y) and level l, $i \in [1, 4]$, respectively. The © means concatenate operation. Concatenate operation refers to the splicing of tensors along the spatial dimension, specifically spatial-wise concatenation, which is employed after two-dimensional discrete wavelet transform. This operation combines feature maps directly along the spatial dimension, allowing the integration of low-frequency and high-frequency components without altering their original spatial layout. It plays a crucial role in constructing effective multi-scale feature representations, enhancing the model's ability to capture cross-scale relationships and improving overall performance.

The intricate interplay between these components is orchestrated. The module is founded upon incorporating two-dimensional discrete wavelet sampling blocks, which meticulously process the feature maps emanating from corresponding downsampling stages. As exemplified in Fig. 3, these cascade modules adhere to a consistent structural blueprint while affording the flexibility to adapt feature map resolutions in response to contextual requirements. It is noteworthy that deploying four cascade blocks achieves the optimal equilibrium between model effectiveness and computational training overhead. This eventually leads to the final generation of the residual image. This residual image, in turn, assumes a pivotal role in governing the intensity of the watermark and the coordinates at which it is embedded, thus finally realizing the watermarked image.

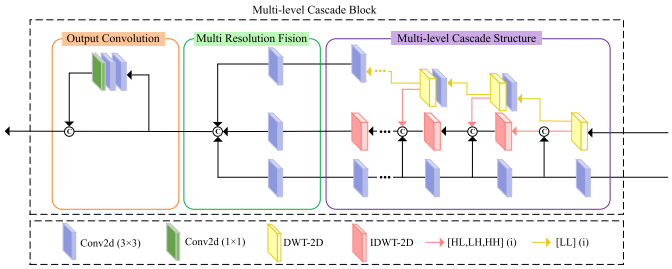

Fig. 3. Detailed architecture of the multi-level cascade block within the wavelet domain cascaded encoder network. This architecture delves into the specific design of the multi-level cascade block, an integral part of the encoder network. It outlines the block's various stages, including convolutional operations and wavelet transformations.

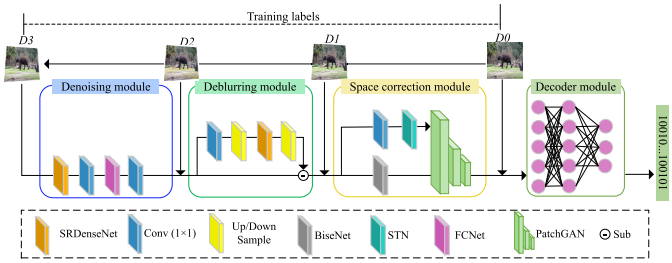

Fig. 4. The structure of the proposed recovery decoder network. The network contains four modules, namely the denoising module, deblurring module, spatial correction module, and decoding module. The first three eliminate noise introduced by the screen-shooting process. The last module is used for decoding.

D. Recovery Decoder Network

The noise introduced by the screen-shooting process will reduce the accuracy of watermark decoding. Therefore, we intend to specifically eliminate the above noise before decoding. To enhance robustness, we introduce a dedicated Recovery decoder Network (RDN) engineered to eliminate noise artifacts before extracting concealed watermark information. The input of RDN is I_{eE} in Fig. 2. Through its internal denoising modules, I_{eE} can be restored to I'_e, aiming to approximate I'_e to I_e to the greatest extent. Using I'_e as the input of the watermark information extraction module reduces the impact of noise introduced by screen-shooting on the watermark.

As shown in Fig. 4, the core of the recovery network includes four modules: the first module is the denoising module $Net_{Denoise}$. $Net_{Denoise}$ is built based on the SRDenseNet [48] network and is used to remove Gaussian noise in the image while reducing color changes. SRDenseNet is a model for super-resolution tasks. We choose it to remove noise. On the one hand, the super-resolution model has certain denoising ability. On the other hand, the super-resolution model can maintain the details of the image, which is an advantage in the watermarking task. Therefore, we train the denoising ability of SRDenseNet through supervised training. The impact of JPEG compression on watermarked images is shown in Fig. 2. The input of the denoising module is I_{eE}, and the output is I'_{eB}. SRDenseNet contains six Dense blocks and utilizes skip connections to capture high-resolution features,

thereby preserving watermarking details. The second module is the deblurring module Net_{Deblur}. Since image blurring is also a form of image noise, we also choose SRDenseNet as the basis to build a deblurring module. Different from the previous, we use the residual structure to establish this module. This is because in the process of screen shooting, blur mainly comes from the translational motion of the camera. We train the residual image as the redundant part generated during imaging, and subtract the residual image from the input to complete the purpose of deblurring. The deblurring module enhances the clarity of areas affected by motion blur. This module is characterized by introducing an up/down sampling structure and operates in residual mode. Net_{Deblur} can generate a residual image, train the residual image to approximate the ghost part caused by motion blur, and then subtract I'_{eB} from the residual image to generate I'_{eA}.

The third module is the spatial correction module. The spatial correction module combines the spatial transformation network STN [49] and the pre-trained image positioning network BiSeNet [50]. BiseNet is sensitive to the location information of the image and is often used for image positioning. Therefore, we choose it as a component of the third module. At the same time, STN also has the effect of correcting images. The obtained features of the two branches enter the PatchGAN [51] network as multiple inputs. At the same time, a 1×1 convolution kernel is used to fine-tune the channel configuration, aiming to solve the problem of geometric deformation of the image caused by perspective distortion in screen-shooting scenarios. The fourth is the decoding module, which outputs the finally extracted watermark information. Fig. 4 shows it comprises multiple layers of fully connected neurons. The number of neurons in the last layer is set to 100, the same as the initial input. The number of watermark information bits is the same. Take a tensor of shape $400 \times 400 \times 3$ as an example. In the decoding module of the RDN model, we will explain how to convert this tensor into 100 bits of data. The output of the previous stage of the decoding module is designed to be a $400 \times 400 \times 3$ tensor. After entering the decoding module, it is processed by multiple convolutional layers. These layers apply convolution operations layer by layer. Multiple convolution kernels are used in the convolution operation to extract features from the input data and generate new feature maps. These feature maps capture the local feature information of the input image. In this process, the size of the convolution kernel used can be 3×3, and the number can be increased or decreased within a certain range. It only needs to ensure that there are a certain number of neurons. In order to introduce nonlinear characteristics, an activation function is usually applied to the output of each convolution layer. In our experiment, the activation function we use is ReLU. After these convolutional layers, we start the linearization step, that is, flattening the final feature map into a one-dimensional vector. This step is crucial for subsequent processing. Next, the flattened vector is input into the fully connected layer, in which each neuron is connected to each element of the input, and the neurons generate new feature representations through weighted calculations. The fully connected layer can contain multiple neurons, and the final output layer is designed to have 100 neurons, with the purpose of generating 100 linear probability values. At the end of the output layer, an activation function is usually applied. We choose the sigmoid function to convert the output result into a probability distribution, ensuring that each output value is between 0 and 1. Finally, each probability is rounded to generate an array with only two values of 0 and 1, which is consistent with the input bit information.

The first three modules of the recovery network should be trained separately. When the training of the three modules is completed, the recovery network is constructed according to the order in Fig. 4. At the same time, the parameters of the three modules should be fixed in the subsequent training. The invention self-made three training decoder datasets for training the above three modules. The specific production method is as follows: Let D_0 be the original watermark image without any external noise generated by the encoder of Stegastamp [13]. Then, by introducing perspective distortion, D_1 is derived from D_0. Then, D_2 is generated by moving D_1 through motion and defocus blur. Finally, D_3 is generated by passing D_2 through a series of noises, including Gaussian noise, color manipulation, and JPEG compression. The three datasets are: $\{D_3, D_2\}$, $\{D_2, D_1\}$, $\{D_1, D_0\}$. The first element in the set is the network's input, and the second element is the network's label, the output.

Before constructing the recovery decoder network, the first three modules have undergone separate training. The denoising module $Net_{Denoise}$ uses the paired dataset $\{D_3, D_2\}$ for training. The training process is $D_2 = Net_{Denoise}(D_3)$, where D_3 is the input and D_2 is the corresponding label. Similarly, the deblurring module uses the dataset $\{D_2, D_1\}$ for training, and the training process is $D_1 = Net_{Deblur}(D_2)$. The training of the spatial correction module uses the dataset $\{D_1, D_0\}$, and the training process is $D_0 = PatchGAN(STN(D_1) \oplus BiSeNet(D_1))$. Among them, the $\oplus$ represents fusion through bitwise addition. These three modules use the same loss function for training, and all use image loss mentioned in subsection E. When these three modules are trained separately, their parameters are fixed. The recovery decoder network is constructed in the order in Fig. 4. Through the processing of I_{eE} by the above network, the noise is extracted, and the watermark information is extracted, which can be used as a decoder to participate in training in the watermark framework.

E. Screen Shooting Noise Simulator

To authentically simulate the screen capture process and enhance the robustness of watermarked images for real-world transmission scenarios, we introduce a noise simulator from the pioneering work of Stegastamp [13]. This simulator, inserted between the encoder and decoder within our robust watermarking model, encompasses a comprehensive spectrum of distortions, including perspective warp, motion and defocus blur, color manipulation, camera system noise, and JPEG compression. It is important to note that our novelty in reducing the impact of screen-shooting lies in the recovery network. It is designed to eliminate noise generated during screen-shooting. Therefore, we eliminate the distortion caused by the screen capture process before decoding.

Algorithm 1 Training WaveRecovery Network N

Input : Cover image C. Initial learning rate $\alpha = 0.0001$. Batch size $b = 8$. Training DataSet M. $Step = 180000$.

Output: Trained network N, θ_N.

while $i < 3$ **do**
  $\{D_i\} \leftarrow$ simulator(D_{i-1}) based on Section III Subsection D;
  //F_i is the feature map generated by RDN ;
  $F_i \leftarrow \mathrm{R}(D_i)$;
  //$\mathcal{L}$ represents the relevant loss function ;
  $g_{\theta R} \leftarrow \nabla_{\theta R}\mathcal{L}_R(F_i, D_{i-1})$;
  // θ represents the hyperparameter in the network ;
  $\theta_R \leftarrow \theta_R + \alpha \cdot \mathrm{SGD}(\theta_R, g_{\theta R})$;
end
foreach $j \leftarrow 1$ **to** $Step$ **do**
  $\{C^{(j)}\}_{j=1}^{b} \sim M$, a batch from the image set;
  $(W, N) = \mathrm{ReShape}(C^j)$;
  //F_j is the feature map generated by WCN ;
  $F_j \leftarrow \mathrm{N}(C^j_{W,N})$;
  // g represents the gradient in the model ;
  $g_{\theta N} \leftarrow \nabla_{\theta N}\mathcal{L}_N(F_j, C^j)$;
  $\theta_N \leftarrow \theta_N + \alpha \cdot \mathrm{Adam}(\theta_N, g_N)$;
end

The rationale underlying the generation of these specific distortions is as follows. Imaging situations where the plane containing the screen is not parallel to the camera's focal plane invariably yields perspective-warping effects. When shooting, people's hands holding the camera will inevitably shake. Sometimes, imaging may also occur while people are walking. Motion and defocus blur are used to simulate the above scenarios. There are various brands of screens and cameras. Therefore, cameras and screens may have different color domains. Given the diversity of camera and screen brands, variations in color domains become inevitable. Discrepancies in color domains result in perceptible color manipulations during image acquisition. Gaussian noise modeling is employed to simulate the inherent system noise of diverse cameras, accounting for the stochastic nature of noise in real-world imaging devices. Compression is frequently employed during image capture and electronic transmission to conserve storage space. Among various compression methods, JPEG compression is a symbol of image compression techniques. By integrating these distortions into the training pipeline, we ensure that our watermarking model is attuned to the intricacies of real-world screen capture scenarios, thereby enhancing its robustness and adaptability to practical environments.

F. Loss Function

To ensure minimal perceptual distortion in the watermarked image, we introduce a frequency information control loss denoted as L_{freq}, specifically focusing on the wavelet domain low-frequency loss. We call it Wavelet loss. This loss is instrumental in regulating the distribution of information within the watermarked image, with the primary objective of minimizing disparities between the low-frequency regions of the cover image and the watermarked image. In conjunction with this, our method incorporates a diverse array of loss functions, encompassing the Mean Squared Error loss denoted as L_2, and LPIPS [53] (Learned Perceptual Image Patch Similarity) loss denoted as L_p, all of which play a pivotal role in facilitating effective image reconstruction. L_2 and L_p are to improve the image quality of the watermarking image. Additionally, we employ the Cross-Entropy loss L_m to supervise the fidelity of the embedded message content. The overarching training loss is mathematically expressed as a weighted composite of these constituent loss components, meticulously formulated as follows:

$$L = \lambda_{\text{freq}} L_{\text{freq}} + \lambda_2 L_2 + \lambda_p L_p + \lambda_m L_m \tag{8}$$

Here, L encapsulates the collective loss, and λ_{freq}, λ_2, λ_p, λ_m denote the weight coefficients assigned to their corresponding loss terms.

To facilitate the training process, the image loss weights λ_{freq}, λ_2, λ_p are initialized to zero during the initial stages of decoder training, gradually being linearly incremented as the decoder attains higher accuracy.

Assuming the watermarking model network as N and the recovery decoder network as R, we outline the detailed training algorithm for the proposed model in Algorithm 1.

This training procedure can be further elaborated by expressing the individual loss terms:

1) Wavelet Loss:

$$L_{\text{freq}} = \sum_{i=1}^{N_{\text{freq}}} \| F^N_{\text{low},i} - F^R_{\text{low},i} \|_2 \tag{9}$$

where N_{freq} represents the number of low-frequency wavelet components, $F^N_{low,i}$ and $F^R_{low,i}$ denote the low-frequency components of the watermark and cover image in the i-th frequency band, respectively.

2) Mean Squared Error Loss (L_2):

$$L_2 = \frac{\sum_{i-1}^{n} (I^N - I^R)^2}{n} \tag{10}$$

where I^N and I^R denote the original cover image and the watermarked image, respectively.

3) LPIPS Loss (L_p):

$$L_p = \mathrm{LPIPS}(I^N, I^R) \tag{11}$$

where $LPIPS$ computes the perceptual similarity between I^N and I^R using a pre-trained neural network.

4) Cross-Entropy Loss(L_m): Cross entropy is often used to evaluate the difference between two probability distributions, especially in deep learning problems. Cross entropy can be used to measure the distance between the true distribution and the predicted distribution. The watermark information can be regarded as a one-dimensional vector composed of 0 and 1. Therefore, the gap between the embedded watermark information and the watermark information extracted by the decoding network can be calculated using cross entropy. The

cross entropy loss can be obtained. The cross entropy loss can be expressed as:

$$L_m = -(wlog(w') + (1-w)log(1-w')) \quad (12)$$

where w represents the embedded watermark information, and w' represents the extracted watermark information.

In Algorithm 1, the network parameters are updated using gradient descent based on the computed L concerning the network weights. This algorithm outlines the training process for the Wavelet Domain Cascade and Recovery network N. It takes a cover image C, initializes learning rate α, batch size b, training dataset M, and training steps $Step$. It begins by iteratively refining the parameters θ_R of the recovery decoder network. The loss function used by the recovery decoder network is the same as the Image loss in Fig. 2. During each iteration, distorted images are generated, and their features are obtained through recovery. The parameters θ_R are updated using stochastic gradient descent (SGD). After three iterations, training shifts to the wavelet domain cascade network N. Cover image batches are sampled here, and feature maps are extracted using N. The network's parameters θ_N are updated using the Adam optimizer. This process continues until the error rate of network N no longer decreases, signifying convergence. This training regimen efficiently integrates these loss terms to enable the watermarking model to exhibit resilience against various distortions while ensuring minimal perceptual distortion between the cover and watermarked images.

IV. Experiments

This section presents our experimental setup, providing details regarding the experimental configuration. We furnish a simulation example in our experimental settings illustrating the watermarked image's susceptibility to various attacks. Concurrently, we demonstrate the correlation between information capacity and visual quality. Subsequently, we embark on a rigorous validation of the effectiveness of our proposed method, conducting ablation experiments that systematically assess the individual contributions of the wavelet cascade encoder network and the recovery decoder network. To determine the comparative performance of our method, we undertook a thorough evaluation. We compare it with some advanced robust watermarking techniques for watermarking frameworks tailored to screen capture scenarios. Additionally, we study the impact of varying shooting angles on decoding accuracy, thus illuminating the degree of orientation sensitivity inherent in our model. Furthermore, we undertake a series of experiments, deliberately introducing diverse intensities of singular noise sources, thereby elucidating the robustness of our approach across a spectrum of noise categories. This comprehensive assessment empowers us to discern the specific noise profiles to which our model exhibits superior resilience. Finally, our framework is the baseline for comparing methods across different combinations of monitors and shooting devices. These methods, thoughtfully curated to encompass both traditional and deep learning-based approaches within the robust watermarking domain, are chosen to be representative of the broader landscape of watermarking techniques.

A. Data Preparation and Experiments Setup

Our dataset for this investigation comprises three components to ensure versatility across diverse image contexts. First, we utilize the COCO2014 dataset, which contains 164,000 natural images. From this, we randomly select 35,000 images for training and partition 4,300 images each for validation and testing, following a balanced 8:1:1 split to standardize learning influence. Second, to adapt the watermarking model to social scenes, we employ the MIRFLICKR-25K [54] dataset, previously used in the Stegastamp framework, allocating 20,000 images for training and 2,500 for validation and testing. Lastly, for the recovery decoder network, we create a dedicated recovery dataset by selecting 10,000 images from COCO2014. These images undergo watermark embedding as set D0, and through the process described in Section III, Subsection D, we generate sets D1, D2, and D3 with increasing distortion levels.

For our experiments, we use 400 × 400 pixel images along with a random bit string for watermark embedding. While the bit string's content varies, its length remains fixed for each model to support different watermarking scenarios. After training the recovery decoder network, we proceed with end-to-end training of the encoding and decoding networks. Models with varying embedding capacities are trained separately, each optimized for the best balance between image quality and decoding accuracy. During training, we use an image batch size of 8 to maintain efficiency within equipment constraints. For testing, watermarked images displayed on AOC series displays are captured using an iPhone 8 Plus to evaluate the model's performance under real-world conditions.

B. The Influence of Watermark Capacity on Visual Quality

For the image's visual quality, we choose the peak signal-to-noise ratio (PSNR) [55] to measure the degree of distortion of the watermark image relative to the cover image. At the same time, we use the structural similarity index (SSIM) [56] to measure the similarity between the image watermark image and the cover image. SSIM considers not only the similarity of brightness but also the similarity of structure and texture. Combining PSNR and SSIM, on the one hand, we evaluate the degree of distortion of the image in the process of watermark embedding. We can encode at least 56 error-corrected bits using BCH [52] codes. BCH error correction codes are pivotal in enhancing the likelihood of accurately recovering the original message. Even in cases where watermark information is severely degraded, the BCH code allows us to discern erroneous message recovery. This attribute is precious when the original message is unknown, as it streamlines the judgment process and confers practical utility to our method. On the other hand, we evaluate the changes in brightness, structure, and texture of the image caused by watermark embedding. For the robustness of the watermark, we choose BitAcc as a measure. It is calculated by dividing the number of correct bits recovered by the total number of bits. The higher the bit accuracy, the stronger the reliability of the watermark. That is, the higher the robustness of the watermark. In each embedding, we encode a 100bits string that consists of a 96bits string with BCH (96, 56) and four zero bits. Acc represents

TABLE I
MODEL PERFORMANCE TRAINED WITH DIFFERENT INFORMATION CAPACITIES

Metric	20bits	50bits	100bits	150bits
PSNR	36.68	35.51	34.22	32.17
SSIM	0.9751	0.9705	0.9637	0.9462
BitAcc	99.80%	99.32%	98.27%	97.79%

the accuracy calculated from the number of successful decodes with the help of BCH.

To assess the impact of watermark capacity on the visual quality of watermarked images, we conduct various experiments involving watermark models with varying capacity specifications. This encompassed scenarios with watermark embedding capacities of 20bits, 50bits, 100bits, and 150bits. Since $BitAcc$ is calculated, we do not use the BCH error correction code here. The outcomes of these experiments are meticulously presented in Table I, which delineates image quality and robustness metrics across these diverse capacity settings. Our observations reveal an inherent trade-off among image quality, watermark robustness, and watermarking capacity. Specifically, as the embedded watermark capacity increases, image quality and robustness exhibit a gradual decrement. In essence, larger watermark capacities tend to introduce a concomitant decline in image quality and robustness. In our subsequent experiments, we opt for a watermark capacity of 100 bits. Other methods use no more capacity than ours.

In order to explore the performance of the model in terms of visual quality under different capacities, Table II compares the performance of the three methods under different capacity conditions, using PSNR and SSIM as evaluation indicators. In the experiment, the data embedding capacity was set to 50 bits, 100 bits, 150 bits and 200 bits respectively. Since RIHOOP does not provide experimental results under 150 bits, no comparison is made here. It can be seen from Table II that in all cases, the PSNR value of our proposed method is the highest. In most cases, the SSIM value of our method is also the highest. When the embedding capacity is 200 bits, the SSIM of our method is not much lower than that of RIHOOP. This proves that our method has an advantage in visual quality.

C. Real-World Distortion Scenario

We present a comprehensive set of sample images systematically generated through the application of our noise simulator, as exemplified in Fig. 5. This sequence of distorted images is methodically produced via linear combinations, encapsulating the entirety of simulated noise effects. The simulator takes the watermarked image as input and yields an output watermark image that incorporates all simulated noise distortions. To ensure the accuracy and impartiality of our comparative assessments, we diligently striven to replicate the experimental conditions adopted in Stegastamp.

The first picture of Fig. 6 shows the specific shooting environment employed in our experiments. Within this controlled environment, we prioritize maintaining consistent lighting conditions and ensuring the stability of shooting

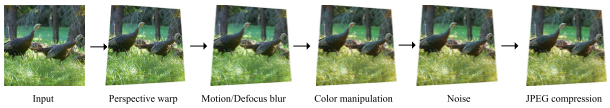

Fig. 5. The pipeline of the image noise simulator. The operational pipeline of the image noise simulator for displaying screen-shooting distortions: These noises are superimposed on the watermark image in a linear sequence, such as perspective warp and color manipulation, mimicking the conditions encountered in screen-shooting scenarios.

Fig. 6. Captured images (first row) with different perspective angles and their corresponding recovered images (second row) by perspective correction. This figure showcases a series of watermarked images captured from different horizontal angles ranging from 15 to 35 degrees. In addition, real shooting environment display.

angles and distances between the shooting device and the screen. We employ a camera phone securely affixed to a tripod to achieve this precision. Additionally, we employ scaling and protractor devices to regulate shooting angles and distances meticulously. Due to the need to control the set angle and distance, the robustness against motion blur cannot be tested in a real environment. Therefore, we demonstrate the robustness of our model against motion blur in simulated noise environments in subsection F. Fig. 6 provides a representative sample of images captured at varying angles using an iPhone 8 Plus. In Fig. 6, the first row meticulously demonstrates the effect of capturing images with left and right angular offsets of 15 degrees, 25 degrees, and 35 degrees, mirroring realistic pirate scenarios where extreme angular offsets are typically avoided due to their adverse impact on image quality during capture. To this end, we constrain the maximum allowable offset to 35 degrees. In the second row in Fig. 6, we showcase the images after perspective correction, which serve as input for message recovery within our decoder. It is pertinent to note that even under stable lighting conditions, corrected images may still exhibit variations in brightness due to differing shooting angles. Consequently, we introduce brightness and contrast distortions within the noise layer to simulate this variability, a crucial consideration for our experiments. Upon comparative analysis of Fig. 5 and Fig. 6, it becomes evident that simulated distortion in the former tends to exceed the real distortion experienced in the latter slightly. This deliberate deviation ensures that our model attains a robustness threshold sufficient for real-world applications through training.

D. Decoding Performance at Different Shooting Angles

By the previously delineated shooting and correction methodologies, we present the decoding outcomes obtained at diverse shooting angles, as comprehensively detailed in Table III. We test it at both horizontal and vertical angles. Notably, we observe the highest decoding accuracy when the shooting angle is maintained at 0 degrees. As the angular offset increases, there is a gradual decrement in decoding

TABLE II
COMPARISON OF RESULTS AT DIFFERENT CAPACITIES

Methods	PSNR/SSIM			
	50bits	100bits	150bits	200bits
Stegastamp [13]	29.88/0.9300	28.50/0.9050	26.47/0.8760	21.79/0.7930
RIHOOP [15]	29.81/0.9474	28.60/0.9362	-	28.01/**0.9320**
WaveRecovery	**35.51/0.9705**	**34.22/0.9637**	**32.17/0.9462**	**30.55**/0.9110

TABLE III
THE DECODING RESULTS UNDER DIFFERENT SHOOTING ANGLES. THE HIGHEST DECODING ACCURACY IS OBSERVED WHEN THE SHOOTING ANGLE IS KEPT AT 0 DEGREES

Horizontal	Acc	Vertical	Acc
Left 35°	92.57%	Down 35°	91.83%
Left 25°	94.29%	Down 25°	93.37%
Left 15°	96.65%	Down 15°	96.62%
0°	98.19%	0°	98.19%
Right 15°	97.62%	Up 15°	97.93%
Right 25°	96.89%	Up 25°	96.95%
Right 35°	96.56%	Up 35°	94.73%

accuracy. This empirical observation underscores our model's propensity for achieving notably high accuracy in scenarios characterized by straight-angle shooting, a vital attribute that bolsters the reliability and consistency of decoding outcomes. It is pertinent to highlight that the influence of vertical offsets on decoding accuracy tends to be generally more pronounced than horizontal offsets. This phenomenon may be attributed to display design conventions, which often prioritize optimizing left-right viewing angles, given users' frequent use of horizontal visual observation. Consequently, our model has evolved to exhibit enhanced robustness in response to left-right shifts. Nonetheless, when the shooting angle is dramatically inclined to 35 degrees downward, we do observe a notable increase in error rates, albeit remaining below 9%. This observation underscores the model's relative sensitivity in extreme scenarios such as steep downward angles. However, it is crucial to emphasize that even in such challenging conditions, our model still demonstrates a commendable level of resilience, underscoring its ability to perform capably across a spectrum of shooting angles. In summary, our experimental findings underscore the robustness of our model in accommodating different shooting angles. This robustness holds paramount significance in ensuring the stability and reliability of decoding outcomes, particularly in light of the diverse shooting angles encountered in everyday applications. It is worth mentioning that if you need to maintain high robustness at larger angles, you can model the noise at large angles in the noise layer.

E. Decoding Performance of The Model Under Different Intensities of Noise

We conduct an extensive series of experiments to comprehensively and meticulously explore the model's robustness across distinct distortion types. As shown in Fig. 7, these noises include brightness manipulation, contrast manipulation, color manipulation, Gaussian noise, median filtering, motion blur, and perspective warp. We construct accuracy curves to assess the model's performance under varying noise intensity levels. Our evaluation metrics include bit accuracy and decoding accuracy, which furnish critical insights into the model's adaptability to individual distortion types. The analysis presented in Fig. 7 yields several key findings. Firstly, our model exhibits commendable robustness to Gaussian noise and color distortion. Even in the presence of these disturbances, the model consistently maintains high accuracy. However, in the context of brightness variation, the model's robustness is slightly reduced, indicating a comparatively heightened sensitivity to the specific distortion types. Furthermore, the model displays its lowest level of robustness when confronted with perspective distortion, primarily because the geometric transformations introduced by perspective distortion profoundly impact image content, thereby significantly hindering watermark information decoding.

Notably, it is observed that our model, despite not explicitly modeling median blur, still exhibits a noteworthy degree of resilience when faced with this distortion, with its robustness comparable to that against perspective distortion. This suggests that the model possesses a certain degree of generalization capability when confronted with previously unencountered distortions. However, it is also evident that the introduction of unmodeled noise types has a more pronounced impact on the model, underscoring the critical importance of precise distortion modeling for successful watermark information decoding.

We explored the impact of wavelet decomposition level on the results. The indicators we used are PSNR, SSIM and BitAcc. As shown in TABLE VII, we recorded the experimental results of wavelet decomposition levels 1, 2 and 3. When level $=1$, the image quality of the watermarked image is the lowest. When level $=3$, the image quality is the highest. This proves that wavelet decomposition is very helpful for network training, especially in terms of image quality.

Our experiments reveal that the model demonstrates resistance to JPEG compression to a certain extent. As shown in Table VIII, this indicates that the embedded watermark information carries a degree of redundancy, which can enhance the model's resilience to JPEG compression. When QF <80, the decoding accuracy will be reduced. At the same time, we conduct transmission tests on social media such as WeChat and QQ. As shown in Table VIII, although social media transmission reduces the accuracy, it does not affect the usage attributes of the model to a considerable extent.

F. Ablation Study

To elucidate the individual performance of each network component, we conduct a comparative analysis between single networks and our proposed methodology, with the baseline aligning with StegaStamp [13]. When a network component

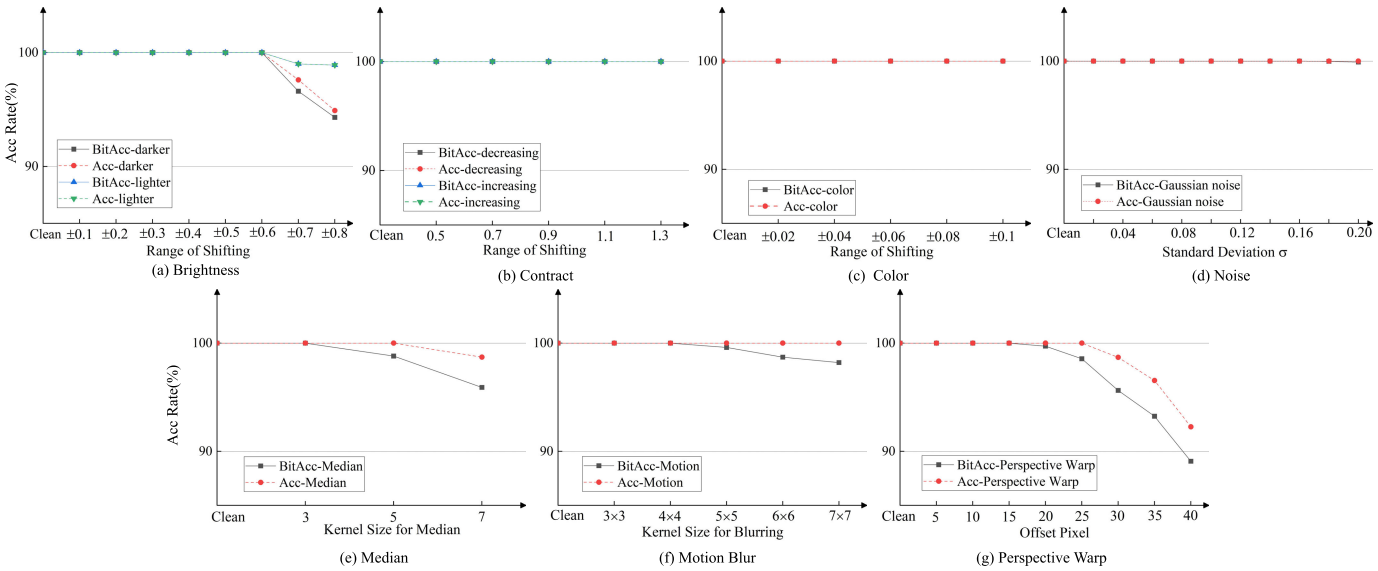

Fig. 7. BitAcc and Acc of the decoding results under different distortions. These line charts present a comprehensive analysis of the watermark decoding performance under different distortions, including brightness, contrast, color manipulation, noise, median, motion blur, and perspective warp. This shows that different noises have different effects on our model, and at the same time, our model is robust to unknown noise.

TABLE IV
EXTRACTION ACCURACY WITH DIFFERENT SHOOTING DISTANCES

Baseline	WCN	RDN	20cm	30cm	40cm	50cm	60cm
✓	-	-	95.18%	93.67%	93.01%	92.17%	91.26%
✓	-	✓	95.59%	94.32%	93.27%	92.61%	92.01%
✓	✓	-	98.08%	97.92%	97.43%	96.98%	96.58%
✓	✓	✓	**98.19%**	**98.07%**	**97.52%**	**97.10%**	**96.67%**

TABLE V
EXTRACTION ACCURACY WITH DIFFERENT SHOOTING ANGLES

Baseline	WCN	RDN	15°	20°	25°	30°	35°
✓	-	-	93.18%	93.06%	91.21%	90.53%	90.17%
✓	-	✓	95.28%	94.62%	92.56%	91.63%	90.21%
✓	✓	-	96.63%	96.17%	95.31%	93.56%	93.28%
✓	✓	✓	**97.14%**	**96.63%**	**95.59%**	**95.31%**	**94.56%**

is omitted from our approach, the baseline is employed with identical configurations. Initially, we evaluate a method without the inclusion of a wavelet domain cascade network for watermark embedding. Subsequently, we introduce the recovery decoder network to integrate into the watermark framework. The watermark embedding process was undertaken using a U-net architecture. As shown in the third row of Table V, the performance exhibited a substantial degradation compared to the combined approach. Conversely, we examine a method that omitted the recovery decoder network but integrated the proposed wavelet domain cascade network into the framework. Watermark extraction was accomplished by combining multiple convolutional layers and a fully connected layer. As indicated in the fourth row of Table V, the performance was notably inferior to the combined approach.

The results clearly illustrate that the optimal outcomes are achieved when both networks are jointly employed. The design of these two networks synergistically enhances image quality and robustness within the watermarking framework, resulting in superior performance compared to the utilization of a single network. Consequently, the simultaneous utilization of both networks enhances the overall performance of the watermarking framework. Moreover, our systematic analysis reveals that the model attains its highest accuracy at a shooting distance of 20 cm. To ensure experimental consistency and comparability in subsequent investigations, we decide to maintain a fixed shooting distance within a 20 cm radius. To enhance the comprehensiveness of the experimental analysis, we conduct a series of ablation studies to investigate the impact of varying angles on diverse models. In Table VI, the data for each angle is calculated from the average of the horizontal viewing angles. For example, 15° represents the average accuracy of 15° to the left and 15° to the right. The data presented in Table VI corroborates the findings, indicating that the outcomes for each model align consistently with the conclusion delineated in Table V. This proves the effectiveness of our network.

To investigate the effectiveness of the recovery decoder network, we conduct experiments involving the generation of recovered images at various stages. As depicted in Fig. 8, the images in the first, third, and fifth columns are generated from the watermark image using the noise simulator. The remaining columns display the corresponding watermark image restoration process with the addition of full noise, as indicated by the arrows denoting the sequence of watermark image restoration. Through visual examination of the watermark image during the recovery process, it is evident that our recovery decoder network is effective. For instance, by inspecting the images in the fourth and third rows within the sixth column, it is apparent that the denoising module within the recovery decoder network is able to partially restore color. Notably, as the final restored watermark image is subject to watermark extraction, the first row in the second column, fourth column, and sixth column illustrate the results of manual-assisted restoration. In these cases, we identify the four vertex coordinates of the image and applied inverse perspective transformation to correct the image. The results presented in Table V corroborate that our proposed recovery decoder network contributes to an enhancement in accuracy within the watermarking framework.

TABLE VI
THE IMAGE QUALITY AND ROBUSTNESS OF THE MODEL TRAINED WITH DIFFERENT LOSSES

Losses			Image Quality		
L_{freq}	L_2	L_p	PSNR	SSIM	BitAcc
-	-	✓	29.65	0.9096	99.87%
✓	-	-	26.69	0.8582	**100.00%**
✓	✓	-	27.13	0.8836	99.15%
✓	-	✓	33.59	0.9367	98.98%
-	✓	✓	32.28	0.9218	98.32%
✓	✓	✓	**34.22**	**0.9637**	98.19%

TABLE VII
THE RESULTS OF WAVELET DECOMPOSITION LEVEL INFLUENCE

Metric	Level		
	1	2	3
PSNR	25.21	28.56	**34.22**
SSIM	0.8132	0.8657	**0.9637**
BitAcc	97.56%	98.10%	**98.27%**

TABLE VIII
EXTRACTION ACCURACY WITH DIFFERENT QF AND SOCIAL MEDIA

QF / Types	95	80	65	50	WeChat	QQ
BitAcc	100.0%	98.5%	97.7%	95.7%	96.3%	97.5%
Acc	100.0%	100.0%	99.5%	96.6%	98.7%	99.2%

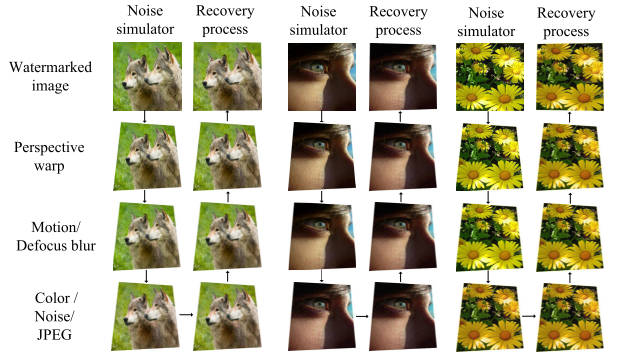

Fig. 8. Recovery process display. Detailed description of the stages involved in the recovery of watermarked images. It demonstrates how watermarked images, initially distorted by various noise simulations such as perspective warp, motion blur, and color manipulation, undergo a process of recovery.

In this paper, we conduct ablation experiments for three different loss functions to evaluate their impact on image quality and robustness. In each set of ablation experiments, we will use the cross entropy function L_m because there is only one information reconstruction loss, and if it is missing, the robustness of the model cannot be trained. Table VII shows the results of PSNR, SSIM and BitAcc under different loss function combinations. When using perceptual loss L_p alone, PSNR is 29.65, SSIM is 0.9096, and BitAcc is 99.87%. In the case of combining L_{freq} and L_p, PSNR and SSIM are significantly improved, reaching 33.59 and 0.9367 respectively. When combining all three loss functions, the best image quality was achieved, with PSNR reaching 34.22 and SSIM reaching 0.9637. In summary, using different combinations of loss functions has different effects on image quality and bit accuracy, and the combination of the three can achieve optimal results in image quality.

G. Comparison

To validate our model, we compare it with other methods, including Stegastamp [13], RHIOOP [15], SSRW [26], HIDDeN [14], and PIMoG [16], at different angles. In addition, we verify the robustness of the model under different device combinations. The experimental results show that our method has better performance.

1) Robustness Comparison Under Different Perspective Angles: We conduct a comparative analysis between our model and other existing methods. SSRW [26] represents a traditional technique tailored for screen-shooting resilient watermarking, while the remaining methods are rooted in deep learning and are also designed for screen-shooting resilient watermarking. To ensure equitable comparisons, we retrain the deep learning-based methods using the dataset outlined in Section IV Subsection A. The results of our comparative evaluation are presented in Table IX, with the most parameters following the default settings in these references. The angle settings in the table are the same as in Table VI. Stegastamp [13], RHIOOP [15], and our method use 100bits embedding capacity, and SSRW [26], HIDDeN [14], and PIMoG [16] use 30bits embedding capacity. Across various shooting angles, our watermarking model consistently outperforms other methods in terms of decoding performance. This demonstrates the superior ability of our model to accurately recover watermark information from images captured from screens. Furthermore, concerning visual quality, our method closely approaches the performance of RIHOOP [15] in objective scores and significantly surpasses Hidden and Stegastamp. This signifies that our approach excels not only in delivering high decoding performance but also in preserving image quality.

2) Comparison of Bit Accuracy for Extracted Watermark Message: Table X presents a comparison of BitAcc for extracted watermark messages among various methods, including Ge et al., Cao et al., Guo et al., SSRGDS, Wavelet-Based, and the proposed WaveRecovery approach. The results demonstrate that DoBMark achieves the highest bit accuracy at 100%, followed closely by Ge et al. with 99.64%. The proposed WaveRecovery method achieves a competitive bit accuracy of 98.19%, outperforming Cao et al. (94.69%) and Wavelet-Based methods (95.20%). Meanwhile, the performance of our method is close to that of SSRGDS.

3) Robustness Comparison Under Different Distances: The Table XI presents the comparative performance of various methods, in terms of accuracy across different distances ranging from 50 cm to 100 cm. WaveRecovery demonstrates consistent performance with accuracy ranging from 95.26% to 97.10%. It shows stable and reliable results across all tested distances. Among the methods compared, Li's Method exhibits the highest overall accuracy, peaking at 99.81% at a distance of 70 cm and maintaining superior performance across all distances. Overall, the results indicate that while our method, WaveRecovery, offers reliable accuracy, there are other methods, such as Li's Method and RIHOOP, that achieve higher accuracy in certain scenarios. This comparison

TABLE IX
THE COMPARISON RESULTS WITH OTHER METHODS. ACROSS VARIOUS SHOOTING ANGLES, OUR WATERMARKING MODEL CONSISTENTLY OUTPERFORMS OTHER METHODS IN TERMS OF DECODING PERFORMANCE

Methods	Message Length	BitAcc under Different Shooting Angle						PSNR	SSIM
		0°	15°	20°	25°	30°	35°		
SSRW [26]	30	93.30%	92.93%	90.07%	89.31%	89.26%	87.32%	33.59	0.9479
HiDDeN [14]	30	69.77%	65.53%	59.26%	61.15%	55.53%	55.21%	29.36	0.9127
PIMoG [16]	30	84.62%	83.35%	79.92%	77.63%	76.59%	76.35%	**37.53**	**0.991**
WaveRecovery	30	**99.36%**	**98.86%**	**98.17%**	**97.32%**	**96.62%**	**95.86%**	34.89	0.9693
Stegastamp [13]	100	97.76%	94.79%	94.68%	93.22%	91.17%	91.07%	28.78	0.9103
RIHOOP [15]	100	96.26%	95.71%	95.37%	94.79%	94.62%	93.98%	**36.63**	**0.9798**
WaveRecovery	100	**98.19%**	**97.14%**	**96.63%**	**95.59%**	**95.31%**	**94.56%**	34.22	0.9637

TABLE X
COMPARISON OF BIT ACCURACY FOR EXTRACTED WATERMARK MESSAGE

Methods	Ge *et al.* [33]	Cao *et al.* [34]	DoBMark [35]	SSRGDS [18]	Wavelet-Based [43]	WaveRecovery
BitAcc (%)	0.9964	0.9469	**1.0000**	0.9870	0.9520	0.9819

TABLE XI
COMPARISON OF BIT ACCURACY FOR EXTRACTED WATERMARK MESSAGE UNDER DIFFERENT SHOOTING DISTANCE

Methods	Distance (cm)					
	50	60	70	80	90	100
Stegastamp [13]	97.11%	97.87%	98.90%	97.21%	97.85%	93.45%
RIHOOP [15]	**99.45%**	99.34%	99.75%	99.25%	98.85%	97.00%
SSRW [26]	93.23%	96.56%	96.33%	93.98%	87.81%	95.23%
PIMOG [16]	95.33%	95.67%	96.83%	94.67%	94.67%	92.83%
SSRGDS [18]	99.20%	**99.59%**	**99.81%**	**99.60%**	**99.05%**	**98.70%**
WaveRecovery	97.10%	96.67%	96.53%	96.20%	95.75%	95.26%

TABLE XII
EXTRACTION ACCURACY WITH DIFFERENT DEVICES IN THE REAL WORLD

Device	Methods	iPhone8 plus	IMX598
AOC 22B2HN	SSRW [26]	92.52%	94.71%
	Stegastamp [13]	97.93%	97.63%
	WaveRecovery	**98.75%**	**99.85%**
AOC 24G2SP	SSRW [26]	91.12%	91.73%
	Stegastamp [13]	96.69%	98.26%
	WaveRecovery	**97.21%**	**98.72%**
INNOCN 24B1XHM	SSRW [26]	91.71%	92.88%
	Stegastamp [13]	97.50%	98.32%
	WaveRecovery	**97.66%**	**99.17%**

TABLE XIII
COMPLEXITY COMPARISON WITH OTHER METHODS

Methods	Encoding (s)	Decoding (s)	Parameters (M)
Stegastamp [13]	24.05	19.62	57.84
RIHOOP [15]	26.37	20.86	59.60
WaveRecovery	31.19	26.83	78.89

provides a comprehensive overview of the effectiveness of various approaches in different distance conditions.

4) Robustness Comparison Under Different Device Combinations: To evaluate the scheme's adaptability comprehensively, we conduct shooting experiments involving diverse screen and mobile phone configurations while maintaining a fixed shooting distance of 20 cm. The outcomes of these experiments are presented in Table XII. We employ three distinct screens and two different mobile phones in our experiment. The results in Table XII reveal that our scheme consistently achieves extraction accuracies exceeding 97% across all device pairs. In contrast, for SSRW [26], the extraction accuracy falls within the range of approximately 91% to 94%, consistently trailing the proposed scheme by at least 3%. This substantial performance gap underscores the robust adaptability of our scheme. We attribute the robust adaptability of our scheme to the design of the recovery decoder network. Irrespective of the device used, the noise introduced during the watermarking process exhibits similarities. Consequently, by effectively eliminating this noise introduced by the screen-shooting process prior to watermark extraction, our scheme ensures high robustness across a variety of devices.

5) Complexity Comparison With Other Methods: To explore the complexity of our proposed model, we compared it with other methods. We mainly focus on the test time and the number of parameters. The test time is divided into encoding time and decoding time. Among the compared methods, we only have the experimental code of some methods. Therefore, we choose the two most classic methods for complexity comparison. In order to reflect the innovation of each method, we only calculate the sum of the number of parameters of the main network model proposed by each method. We selected 50 pictures for testing. The experimental results are shown in Table XIII. The more the number of parameters, the longer the time spent.

V. CONCLUSION

This paper introduces a novel and robust watermarking framework that leverages wavelet domain learning to enhance the visual quality of watermark images and employs restoration techniques to boost model robustness. We propose a multi-level cascade network built upon a wavelet domain transform architecture. This architecture ensures that fewer watermarks are embedded in the low-frequency regions of the cover image, optimizing the visual quality of the resulting

watermarked image. Furthermore, our framework incorporates the training of a recovery decoder network designed to handle various types of distortion encountered during watermark recovery from distorted transmissions. This approach enhances the robustness of the watermarked image against transmission distortions. Instead, we focus on noise elimination incurred during the complex transmission process. However, our method shows limited robustness against perspective distortion and motion blur, particularly at extreme angles. There is a trade-off between image quality and watermark capacity: as capacity increases, both resilience and visual quality decrease. This compromise could limit practical usability when high visual fidelity is needed. Furthermore, real-world scenarios with dynamic lighting, extreme angles, or rapid movement may introduce unforeseen challenges not covered in controlled testing. In future research endeavors, we aim to further enhance robustness through more precise noise modeling. Using the frequency domain, we can predict the location of screen-shooting noise during embedding to adaptively embed highly robust regions.

REFERENCES

[1] W. Kim, S. H. Lee, and Y. Seo, "Image fingerprinting scheme for print-and-capture model," in *Proc. Adv. Multimedia Inf. Process.-PCM*, Nov. 2006, pp. 106–113.

[2] Y. Huang, B. Niu, H. Guan, and S. Zhang, "Enhancing image watermarking with adaptive embedding parameter and PSNR guarantee," *IEEE Trans. Multimedia*, vol. 21, no. 10, pp. 2447–2460, Oct. 2019.

[3] Z. Chen, L. Li, H. Peng, Y. Liu, and Y. Yang, "A novel digital watermarking based on general non-negative matrix factorization," *IEEE Trans. Multimedia*, vol. 20, no. 8, pp. 1973–1986, Aug. 2018.

[4] Y. Wu, G. Meng, and Q. Chen, "Embedding novel views in a single JPEG image," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 14519–14527.

[5] R. Hu and S. Xiang, "Cover-lossless robust image watermarking against geometric deformations," *IEEE Trans. Image Process.*, vol. 30, pp. 318–331, 2021.

[6] L. Lin, D. Wu, J. Wang, Y. Chen, X. Zhang, and H. Wu, "Automatic, robust and blind video watermarking resisting camera recording," *IEEE Trans. Circuits Syst. Video Technol.*, early access, Aug. 23, 2024, doi: 10.1109/TCSVT.2024.3448502.

[7] T. Qiao et al., "Scalable universal adversarial watermark defending against facial forgery," *IEEE Trans. Inf. Forensics Security*, vol. 19, pp. 8998–9011, 2024.

[8] T. Qiao et al., "A novel model watermarking for protecting generative adversarial network," *Comput. Secur.*, vol. 127, Apr. 2023, Art. no. 103102.

[9] A. Pramila, A. Keskinarkaus, and T. Seppänen, "Toward an interactive poster using digital watermarking and a mobile phone camera," *Signal, Image Video Process.*, vol. 6, no. 2, pp. 211–222, Feb. 2011.

[10] C.-H. Chen, Y.-L. Tang, and W.-S. Hsieh, "Print-and-scan resilient watermarking through polarizing DCT coefficients," *IEICE Trans. Inf. Syst.*, vol. 96, no. 10, pp. 2208–2214, 2013.

[11] H. Fang et al., "A camera shooting resilient watermarking scheme for underpainting documents," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 11, pp. 4075–4089, Nov. 2020.

[12] J. Hayes and G. Danezis, "Generating steganographic images via adversarial training," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 30, Dec. 2017, pp. 1–10.

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

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

[15] J. Jia et al., "RIHOOP: Robust invisible hyperlinks in offline and online photographs," *IEEE Trans. Cybern.*, vol. 52, no. 7, pp. 7094–7106, Jul. 2022.

[16] 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 *Proc. 30th ACM Int. Conf. Multimedia*, Oct. 2022, pp. 2267–2275.

[17] M. He, B. Feng, Y. Guo, J. Weng, and W. Lu, "Camera-shooting resilient watermarking on image instance level," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 34, no. 11, pp. 10874–10887, Nov. 2024.

[18] Y. Li, X. Liao, and X. Wu, "Screen-shooting resistant watermarking with grayscale deviation simulation," *IEEE Trans. Multimedia*, vol. 26, pp. 10908–10923, 2024.

[19] M. Plata and P. Syga, "Robust spatial-spread deep neural image watermarking," in *Proc. IEEE 19th Int. Conf. Trust, Secur. Privacy Comput. Commun. (TrustCom)*, Dec. 2020, pp. 62–70.

[20] K. Solanki, U. Madhow, B. S. Manjunath, S. Chandrasekaran, and I. El-Khalil, "'Print and scan' resilient data hiding in images," *IEEE Trans. Inf. Forensics Security*, vol. 1, no. 4, pp. 464–478, Dec. 2006.

[21] P. Chakravarthula, E. Tseng, T. Srivastava, H. Fuchs, and F. Heide, "Learned hardware-in-the-loop phase retrieval for holographic near-eye displays," *ACM Trans. Graph.*, vol. 39, no. 6, pp. 1–18, Nov. 2020.

[22] J. Fu et al., "Chartem: Reviving chart images with data embedding," *IEEE Trans. Vis. Comput. Graph.*, vol. 27, no. 2, pp. 337–346, Feb. 2021.

[23] P. Bas, J.-M. Chassery, and B. Macq, "Geometrically invariant watermarking using feature points," *IEEE Trans. Image Process.*, vol. 11, no. 9, pp. 1014–1028, Sep. 2002.

[24] H. Cui, H. Bian, W. Zhang, and N. Yu, "UnseenCode: Invisible on-screen barcode with image-based extraction," in *Proc. IEEE Conf. Comput. Commun.*, Apr. 2019, pp. 1315–1323.

[25] S. H. Amiri and M. Jamzad, "Robust watermarking against print and scan attack through efficient modeling algorithm," *Signal Process., Image Commun.*, vol. 29, no. 10, pp. 1181–1196, Nov. 2014.

[26] 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, Jun. 2019.

[27] C. Hui, S. Liu, W. Cui, J. Zeng, F. Jiang, and D. Zhao, "Adaptive flexible 3D histogram watermarking," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2021, pp. 1–6.

[28] S. Gaj, A. K. Rathore, A. Sur, and P. K. Bora, "A robust watermarking scheme against frame blending and projection attacks," *Multimedia Tools Appl.*, vol. 76, no. 20, pp. 20755–20779, Oct. 2017.

[29] E. Wengrowski and K. Dana, "Light field messaging with deep photographic steganography," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1515–1524.

[30] H. Fang, Z. Jia, Y. Qiu, J. Zhang, W. Zhang, and E.-C. Chang, "De-END: Decoder-driven watermarking network," *IEEE Trans. Multimedia*, vol. 25, pp. 7571–7581, 2023.

[31] H. Fang, Z. Jia, H. Zhou, Z. Ma, and W. Zhang, "Encoded feature enhancement in watermarking network for distortion in real scenes," *IEEE Trans. Multimedia*, vol. 25, pp. 2648–2660, 2023.

[32] X. Zhong, P. -C. Huang, S. Mastorakis, and F. Y. Shih, "An automated and robust image watermarking scheme based on deep neural networks," *IEEE Trans. Multimedia*, vol. 23, pp. 1951–1961, 2021.

[33] S. Ge, J. Fei, Z. Xia, Y. Tong, J. Weng, and J. Liu, "A screen-shooting resilient document image watermarking scheme using deep neural network," *IET Image Process.*, vol. 17, no. 2, pp. 323–336, Feb. 2023.

[34] F. Cao, D. Guo, T. Wang, H. Yao, J. Li, and C. Qin, "Universal screen-shooting robust image watermarking with channel-attention in DCT domain," *Expert Syst. Appl.*, vol. 238, Mar. 2024, Art. no. 122062.

[35] D. Guo, X. Zhu, F. Li, H. Yao, and C. Qin, "DoBMark: A double-branch network for screen-shooting resilient image watermarking," *Expert Syst. Appl.*, vol. 246, Jul. 2024, Art. no. 123159.

[36] C. Zhang, P. Benz, A. Karjauv, and I. S. Kweon, "Universal adversarial perturbations through the lens of deep steganography: Towards a Fourier perspective," in *Proc. AAAI Conf. Artif. Intell.*, May 2021, vol. 35, no. 4, pp. 3296–3304.

[37] G. Wen, Z. Li, K. Azizzadenesheli, A. Anandkumar, and S. M. Benson, "U-FNO—An enhanced Fourier neural operator-based deep-learning model for multiphase flow," *Adv. Water Resour.*, vol. 163, May 2022, Art. no. 104180.

[38] L. Chi, B. Jiang, and Y. Mu, "Fast Fourier convolution," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 33, Dec. 2020, pp. 4479–4488.

[39] J. Zhang, Y. Liao, X. Zhu, H. Wang, and J. Ding, "A deep learning approach in the discrete cosine transform domain to median filtering forensics," *IEEE Signal Process. Lett.*, vol. 27, pp. 276–280, 2020.

[40] H. Chen, T. Zhu, Y. Zhao, B. Liu, X. Yu, and W. Zhou, "Low-frequency image deep steganography: Manipulate the frequency distribution to hide secrets with tenacious robustness," 2023, *arXiv:2303.13713*.
[41] J. Li, S. You, and A. Robles-Kelly, "A frequency domain neural network for fast image super-resolution," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2018, pp. 1–8.
[42] Z. Q. J. Xu, Y. Zhang, and Y. Xiao, "Training behavior of deep neural network in frequency domain," in *Proc. Int. Conf. Neural Inf. Process.*, Dec. 2019, pp. 264–274.
[43] J. Lu, J. Ni, W. Su, and H. Xie, "Wavelet-based CNN for robust and high-capacity image watermarking," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, 2022, pp. 1–6.
[44] B. Zheng et al., "Learning frequency domain priors for image demoireing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 11, pp. 7705–7717, Nov. 2022.
[45] Z. Liu, J. Xu, X. Peng, and R. Xiong, "Frequency-domain dynamic pruning for convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 31, Dec. 2018, pp. 1–11.
[46] Y. Xu, C. Mou, Y. Hu, J. Xie, and J. Zhang, "Robust invertible image steganography," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 7875–7884.
[47] Q. Li, L. Shen, S. Guo, and Z. Lai, "Wavelet integrated CNNs for noise-robust image classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 7245–7254.
[48] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2472–2481.
[49] C.-H. Lin and S. Lucey, "Inverse compositional spatial transformer networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2568–2576.
[50] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "BiseNet: Bilateral segmentation network for real-time semantic segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 325–341.
[51] Y.-L. Chang, Z. Y. Liu, K.-Y. Lee, and W. Hsu, "Free-form video inpainting with 3D gated convolution and temporal PatchGAN," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9066–9075.
[52] R. C. Bose and D. K. Ray-Chaudhuri, "On a class of error correcting binary group codes," *Inf. Control*, vol. 3, no. 1, pp. 68–79, 1960.
[53] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 586–595.
[54] J. Yu, H. Zhou, Y. Zhan, and D. Tao, "Deep graph-neighbor coherence preserving network for unsupervised cross-modal hashing," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 5, 2021, pp. 4626–4634.
[55] J. Korhonen and J. You, "Peak signal-to-noise ratio revisited: Is simple beautiful?" in *Proc. 4th Int. Workshop Quality Multimedia Exper.*, Jul. 2012, pp. 37–38.
[56] S. Wang, A. Rehman, Z. Wang, S. Ma, and W. Gao, "SSIM-motivated rate-distortion optimization for video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 4, pp. 516–529, Apr. 2012.

Linbo Fu received the B.E. degree from the College of Computer Science and Technology, Guizhou University, Guiyang, China, in 2021, and the M.S. degree from the College of Computer Science and Technology, Hunan University, Changsha, China, in 2024. His current research interests include watermarking.

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 is currently a Professor and the Doctoral Supervisor of Hunan University, China. He was 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, MD, USA. His current research interests include multimedia forensics, steganography, and watermarking. He is the Secretary and a member of Technical Committee (TC) on Multimedia Security and Forensics of Asia–Pacific Signal and Information Processing Association and a member of TC on Computer Forensics of Chinese Institute of Electronics and TC on Digital Forensics and Security of China Society of Image and Graphics. He is serving as an Associate Editor for the *IEEE Signal Processing Magazine*.

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

Li Dong received the B.Eng. degree from Chongqing University in 2012 and the M.S. and Ph.D. degrees from the University of 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. His research interests include statistical image modeling and processing, multimedia security, and forensic.

Zheng Qin (Member, IEEE) received the Ph.D. degree in computer software and theory from Chongqing University, China, in 2001. From 2010 to 2011, he was a Visiting Scholar with the Department of Computer Science, Michigan University. He is currently a Professor with the College of Computer Science and Electronic Engineering, Hunan University, where he is the Vice Dean. He is also the Director of Hunan Key Laboratory of Big Data Research and Application and the Vice Director of Hunan Engineering Laboratory of Authentication and Data Security. His research interests include network and data security, privacy, data analytics and applications, machine learning, and applied cryptography. He is a member of China Computer Federation (CCF).