Aparecium: Revealing Secrets from Physical Photographs

Zhe Lei[1], Jie Zhang[2*], Jingtao Li[1*], Tianwei Zhang[3], Haibin Kan[1], Weiming Zhang[4], Nenghai Yu[4]
[1] School of Computer Science, Fudan University
[2] Centre for Frontier AI Research, Agency for Science, Technology and Research
[3] School of Computer Science and Engineering, Nanyang Technological University
[4] CAS Key Laboratory of Electro-Magnetic Space Information, University of Science and Technology of China

2025 IEEE International Conference on Multimedia and Expo (ICME) | 979-8-3315-9495-4/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICME59968.2025.11209842

***Abstract*—Watermarking photographs is a crucial tool for safeguarding copyrights and can serve as a more aesthetically pleasing alternative to QR codes. In recent years, watermarking methods based on deep learning have proved superior robustness against complex physical distortions than traditional watermarking methods. However, they have some limitations that render them less effective in practice. For instance, current solutions necessitate physical photographs to be rectangular for accurate localization, can't handle physical bending or folding, and require the hidden area to be completely captured at a close distance and small angle. To overcome these challenges, we propose a novel deep watermarking framework dubbed *Aparecium*. Specifically, we preprocess secrets (*i.e.*, watermarks) into a visible pattern and then embed it into the cover image invisibly, which is symmetrical to the final decoding-then-extracting process. To capture the watermarked region from complex physical scenarios, edge distortion is also introduced. Finally, we adopt a three-stage training strategy for training convergence. Extensive experiments demonstrate that *Aparecium* is not only robust against different digital distortions, but also can resist different physical distortions, such as screen-shooting and printing-shooting, even in severe cases including different shapes, curvature, folding, incompleteness, long distances, and big angles while maintaining high visual quality. Furthermore, some ablation studies are also conducted to verify our design.**

***Index Terms*—image watermarking, physical robustness**

I. Introduction

Image watermarking is a widely-used technique for discreetly incorporating information into images without sacrificing visual quality, while still allowing for information extraction despite different distortions. According to the type of embedded secrets, watermarking can be leveraged for different purposes. For example, embedding copyright information into images can be utilized for the protection against copyright infringement, while embedding hyperlinks into images enables redirection to arbitrary information when the image is scanned (*e.g.*, by a mobile device), which can be regarded as an aesthetically pleasing alternative to unattractive QR codes. Robustness is the most significant objective for watermarking, making it effective in practical applications.

Traditional watermarking methods mainly focus on robustness against digital distortions, such as JPEG compression, Gaussian noise, affine transformation, etc. Commonly, each traditional method aims to resist a specific distortion. In recent years, deep learning-based watermarking methods [1]–[5], which simulate complex distortions in an end-to-end way, have achieved tremendous success, making it easy to obtain general robustness. However, some of these methods, such as HiDDeN [1], are only resilient to digital distortions but fragile to physical distortions, where the image is physically captured. To address it, there are some following attempts. For example, Wengrowski and Dana [2] introduce a camera-display transfer function (CDTF) to obtain robustness against screen-shooting. Afterward, the popular StegaStamp [3] is proposed which can guarantee general robustness against both screen-shooting and print-shooting. Very recently, Jia *et al*. [6] propose Offline-to-online to embed watermarks into local regions, which can further improve visual quality meanwhile preserving physical robustness. Nevertheless, there are still some limitations of StegaStamp [3] or Offline-to-online [6]: 1) it requires physical photographs to be rectangular for accurate localization; 2) it is unable to handle bending or folding, and 3) it necessitates the hidden area to be fully captured at a close distance and small angle. The above constraints degrade their effectiveness in the wild.

To effectively reveal secrets from physical photographs, this paper presents a novel robust deep watermarking named *Aparecium*, whose framework is displayed in Figure 1. Different from current methods [2], [3], [6] that map bit-string messages into noise patterns, we propose to preprocess the bit-string message to a semantic pattern by a series of transposed convolutions, and then encode it into the target cover image, which is symmetrical to the subsequent decoding-then-extracting process. We point out that diffusing information into the pattern and employing a incremental decoding-then-extracting approach can significantly enhance decoding capabilities. Besides, substituting edge distortion for the widely used edge loss in existing methods can improve both visual quality and localization accuracy. More importantly, we introduce a three-stage training strategy to incrementally train the above-mentioned five modules.

Extensive experiments demonstrate that the proposed *Aparecium* can achieve robustness against different digital and physical distortions meanwhile preserving satisfied visual quality. Importantly, *Aparecium* is able to successfully reveal secrets from the target photograph suffering various physical distortions, including different shapes, curvature, folding, incompleteness, different distances, and different angles. We also

*Corresponding author. lijt@fudan.edu.cn, zhang_jie@cfar.a-star.edu.sg.

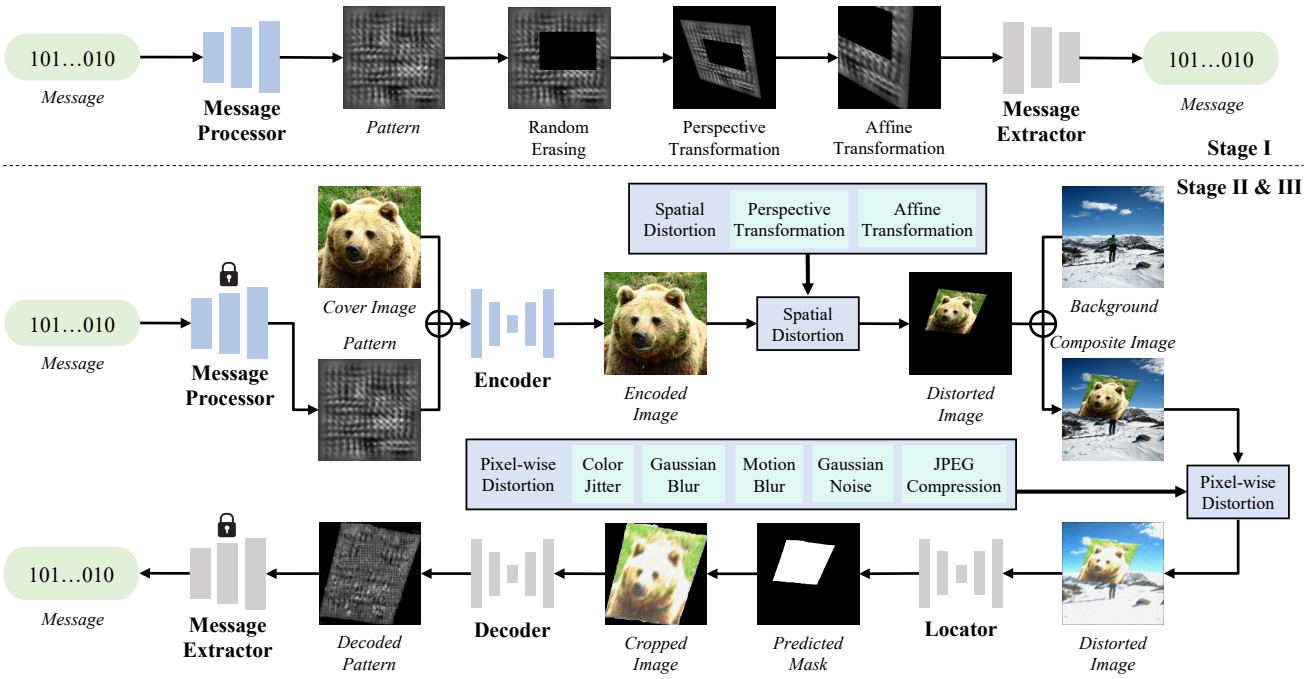

Fig. 1. The overall framework of the proposed method, where we adopt a three-stage training strategy. At Stage I, we jointly train the message processor and message extractor, which are fixed at Stage II while we optimize all models at Stage III. After training, the blue-colored models are used for watermark embedding and the gray-colored models are leveraged for watermark extracting.

conduct many ablation studies to verify our design.

II. Related Work

A. Digital Robust Deep Watermarking

Digital robust deep watermarks are particularly suitable for scenarios where images are transmitted in the digital channel. Baluja [7] proposed the first end-to-end training framework for information hiding and extraction. Based on it, HiDDeN [1] appended a noise layer between watermark embedding and extraction, which included a series of differentiable distortions, thereby obtaining the desired robustness to such distortions. Then, Luo *et al*. [8] leveraged adversarial training to improve robustness against unknown distortions. Afterward, TrustMark [9] interpolates the residual to arbitrary sizes to achieve watermark embedding in images of arbitrary resolutions. Subsequently, JigMark [10] enhances robustness against edits performed by diffusion models through comparative learning of unedited and edited image pairs derived from the diffusion model. To extract information from small watermarked areas within an image, WAM [11] conducts pixel-level detection and message extraction during the decoding process. *However, all these methods are fragile to physical distortions such as screen-shooting and print-shooting.*

B. Physical Robust Deep Watermarking

Our approach is closely related to physical robust deep watermarking, which not only ensures robustness for digital distortions but also for physical distortions. Previous studies, such as LFM [2] and PIMoG [12], have focused on enhancing the robustness against screen shooting. LFM [2] achieved robustness by training a CDTF network, while PIMoG [12] designed the most influenced distortion in the noise layer. Both methods simulated the distortion present in screen-shooting and achieved excellent robustness. However, their robustness against print-shooting is limited.

In contrast, StegaStamp [3] and Offline-to-online [6] have achieved robustness against both screen-shooting and print-shooting. Similar to HiDDeN [1], StegaStamp [3] simulates these physical distortions into the training process, which sacrifices visual quality for certain robustness. To improve visual quality, Offline-to-online [6] hides information in sub-images, namely, a local region of original images, and requires a localization network to locate the watermarked region. *However, during the watermark extracting, both methods require physical photographs to be presented in a complete and flat manner during shooting, which is not applicable in some practical scenarios.*

III. Methodology

A. The Framework of Aparecium

1) Overview: As shown in Figure 1, *Aparecium* consists of several components, including a message processor, an encoder, a locator, a decoder, and a message extractor. Given a message, the message processor transforms it into a pattern, which is then encoded into a cover image by the encoder while maintaining visual similarity. The resulting encoded image can be printed or displayed on a screen and subsequently captured. The locator component then identifies the position of the encoded image and generates a mask that is used to automatically crop out the encoded image. The pattern decoder then decodes the pattern, and the hidden message is

ultimately extracted by the message extractor based on the recovered pattern. In the following sections, we will describe each component in detail.

2) Message Processor: The message processor component is responsible for converting the message into a single-channel pattern, which enables the encoder to conceal the message within the cover image more effectively. To achieve better robustness, the message processor aims to distribute the message as evenly as possible throughout the pattern.

3) Encoder: The encoder component aims to encode the single-channel pattern into a three-channel RGB image while minimizing the visual discrepancies between the encoded image and the cover image.

4) Locator: The locator is introduced to identify the location of the encoded image within a captured physical photograph. To address situations such as incomplete image capture, we select the salient object detection network U^2-Net [13] as our locator.

5) Decoder: After the location of the encoded image has been identified by the locator, the image is automatically cropped based on the generated mask. The main function of the decoder is to decode the single-channel pattern from the cropped image. During the pattern decoding process, pixel-wise distortions such as color jitter, noise, and blur are randomly introduced to enhance the robustness against such distortions. We employ the U-Net [14] as its architecture.

6) Message Extractor: The message extractor is a critical component in retrieving hidden messages from given patterns. However, even after the pattern has been decoded, spatial distortions such as perspective transformation and affine transformation may still exhibit. To address this issue, we utilize the ConvNeXt [15] classification network as our message extractor.

B. Three-stage Training Strategy

To achieve the desired functionality of the above five modules, we adopt a three-stage training strategy, which will be introduced in the following part.

1) Training Stage I: In the first stage, our focus is on training the message processor and the message extractor. Specifically, the message processor is responsible for converting randomly generated binary strings into patterns. To ensure that the message is diffused into the pattern and that the message extraction process can withstand spatial distortions, we introduce different distortions to the pattern prior to extracting. These distortions include (a) random erasing, which involves randomly erasing a rectangular portion of the pattern; (b) perspective transformation, which simulates a scenario where the camera and the image are not aligned; and (c) affine transformation, which encompasses rotation, translation, and scaling. Ultimately, the message extractor is able to extract the hidden message from the distorted pattern. We adopt $\mathcal{L}_{\mathrm{I}}$ to constrain training stage I, *i.e.*,

$$\mathcal{L}_{\mathrm{I}} = \mathcal{L}_{msg} = \ell_{bce}\left(\mathbf{S}_{gt}, \mathbf{S}_{ext}\right), \tag{1}$$

TABLE I
COMPARISON WITH THE STATE-OF-THE-ART METHODS.

Method	PSNR	SSIM	Manual Localization
HiDDeN [1]	20.98	0.4863	✓
PIMoG [12]	34.61	0.9414	✓
StegaStamp [3]	26.46	0.8802	✗
Offline-to-online [6]	31.01	0.9648	✓
Aparecium	33.48	0.9883	✗

where $\mathbf{S}_{gt}$ denotes the secret message in the ground truth, $\mathbf{S}_{ext}$ signifies the extracted secret message.

2) Training Stage II: In this stage, we fixed the parameters of the message processor and message extractor and only train the encoder, locator, and decoder. Once the message processor generates a pattern based on the random message, the encoder encodes the pattern and localization information into the cover image. To simulate spatial distortions that may occur during the photo-taking process, we apply spatial distortions to the encoded image, including perspective transformation and affine transformation. Furthermore, since the encoder is inclined to add marks at the edges of the images for localization, the rotation and translation in the affine transformation can randomly remove portions of the image's edges. This effectively removes edge marks and enhances visual quality while diffusing the localization information throughout the image, improving its robustness in localization. After composing the spatially distorted image with the background image, we add pixel-wise distortions to the composite image, including brightness, contrast, saturation, hue, Gaussian blur, motion blur, Gaussian noise, and JPEG compression. Next, the locator locates the position of the watermarked region of the distorted composite image. Then, we will obtain the input of the following decoder after cropping. Finally, the decoder decodes the pattern from the cropped image. The loss function for stage II consists of visual loss $\mathcal{L}_{vis}$, localization loss $\mathcal{L}_{loc}$, and pattern loss $\mathcal{L}_{pat}$, *i.e.*,

$$\begin{aligned}\mathcal{L}_{\mathrm{II}} =& \lambda_1\mathcal{L}_{vis} + \lambda_2\mathcal{L}_{loc} + \lambda_3\mathcal{L}_{pat}\\ =& \lambda_1\left[\ell_{mse}\left(\mathbf{I}_{co}, \mathbf{I}_{en}\right) + \ell_{ssim}\left(\mathbf{I}_{co}, \mathbf{I}_{en}\right)\right]\\ &+ \lambda_2\ell_{bce}\left(\mathbf{M}_{gt}, \mathbf{M}_{pred}\right)\\ &+ \lambda_3\left[\ell_{mse}\left(\mathbf{P}_{gt}, \mathbf{P}_{de}\right) + \ell_{ssim}\left(\mathbf{P}_{gt}, \mathbf{P}_{de}\right)\right],\end{aligned} \tag{2}$$

where I_{co} represents the cover image, I_{en} denotes the encoded image, M_{gt} denotes the ground truth mask, M_{pred} indicates the predicted mask, P_{gt} represents the ground truth pattern and P_{de} signifies the decoded pattern.

3) Training Stage III: During the third stage, we unfix the parameters of the Message Processor and Message Extractor and fine-tune all five modules in an end-to-end way. The loss function $\mathcal{L}_{\mathrm{III}}$ can be written as follows:

$$\mathcal{L}_{\mathrm{III}} = \lambda_1\mathcal{L}_{vis} + \lambda_2\mathcal{L}_{loc} + \lambda_3\mathcal{L}_{pat} + \lambda_4\mathcal{L}_{msg}. \tag{3}$$

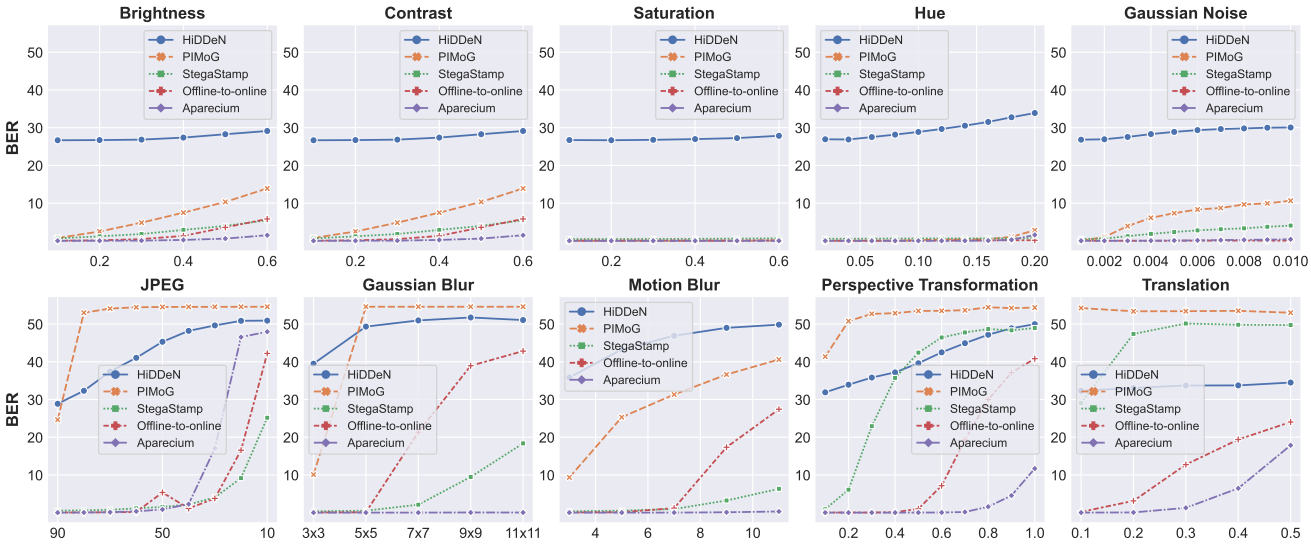

Fig. 2. The comparison of robustness against different digital distortions among different watermarking methods.

TABLE II
THE COMPARISON OF ROBUSTNESS AGAINST SCREEN-SHOOTING UNDER DIFFERENT SHOOTING DISTANCES.

Method	10cm	20cm	30cm	40cm	50cm
Offline-to-online [6]	0.28%	0.34%	0.92%	4.22%	3.21%
Aparecium	0.72%	0.33%	0.76%	2.23%	1.73%

TABLE III
THE COMPARISON OF ROBUSTNESS AGAINST SCREEN-SHOOTING UNDER DIFFERENT SHOOTING ANGLES.

Horizontal	Offline [6]	Aparecium	Vertical	Offline [6]	Aparecium
Left 60°	6.16%	1.75%	Up 60°	5.00%	1.99%
Left 45°	2.95%	1.55%	Up 45°	1.22%	0.65%
Left 30°	0.91%	0.52%	Up 30°	0.94%	0.46%
0°	1.16%	0.41%	0°	1.16%	0.41%
Right 30°	0.81%	0.19%	Down 30°	1.94%	0.49%
Right 45°	2.40%	0.55%	Down 45°	3.04%	1.09%
Right 60°	1.48%	0.53%	Down 60°	3.93%	1.83%

TABLE IV
COMPARISON OF ROBUSTNESS AGAINST PRINT-SHOOTING UNDER DIFFERENT SHOOTING DISTANCES.

Method	30cm	60cm	90cm	120cm	150cm
Offline-to-online [6]	0.33%	0.46%	1.06%	3.57%	4.23%
Aparecium	0.20%	0.32%	1.08%	1.48%	3.96%

TABLE V
THE COMPARISON OF ROBUSTNESS AGAINST PRINT-SHOOTING UNDER DIFFERENT SHOOTING ANGLES.

Horizontal	Offline [6]	Aparecium	Vertical	Offline [6]	Aparecium
Left 60°	6.75%	4.28%	Up 60°	9.89%	3.85%
Left 45°	6.18%	2.97%	Up 45°	4.20%	1.77%
Left 30°	3.15%	1.29%	Up 30°	2.93%	1.45%
0°	0.87%	0.75%	0°	0.87%	0.75%
Right 30°	2.35%	0.82%	Down 30°	1.02%	1.13%
Right 45°	4.68%	1.39%	Down 45°	2.92%	0.72%
Right 60°	9.83%	2.54%	Down 60°	6.40%	4.07%

IV. EXPERIMENT

A. Experiment Settings

We utilize the COCO dataset and compose messages consisting of 196 randomly generated binary bits for training and testing. We conduct comparative experiments with the state-of-the-art deep watermarking methods, namely HiDDeN [1], PIMoG [12], StegaStamp [3], and Offline-to-online [6]. To ensure a fair comparison, we retrain HiDDeN [1], PIMoG [12], and StegaStamp [3] with image size set to 256×256 and message length set to 196 bits.

In the digital robustness experiments, we utilize Kornia to facilitate our testing process. During the physical world experiment, we utilize a DELL S2421NX monitor for display and use an iPhone 13 Pro for shooting by default. Other devices such as a MacBook Pro 14 laptop and a Redmi 12C are also leveraged to justify the general robustness. The default shooting angle is 0°, while the default distance for screen-shooting and print-shooting are 30cm and 90cm, respectively.

B. Visual Quality

As shown in Table I, *Aparecium* exhibits lower PSNR but has a higher SSIM than PIMoG, and outperforms other methods in terms of visual quality. The reason why the visual quality of Offline-to-online [6] is lower than what they report (PSNR: 31.01 v.s. 32.95; SSIM: 0.9648 v.s. 0.9677) is that we randomly select sub-image positions for a fair comparison, instead of manually selecting high-frequency areas. Moreover, both StegaStamp [3] and *Aparecium* can automatically locate and crop the physical photographs, whereas both HiDDeN [1] and PIMoG [12] require manual perspective transformation correction for images due to the absence of locators and Offline-to-online [6] requires manual cropping before locating the sub-image, making them inefficient in practice.

C. Robustness Against Digital Distortions

To compare the digital robustness with the baseline methods, we simulate different distortions. The results are depicted in Figure 2, and it can be observed that after increasing

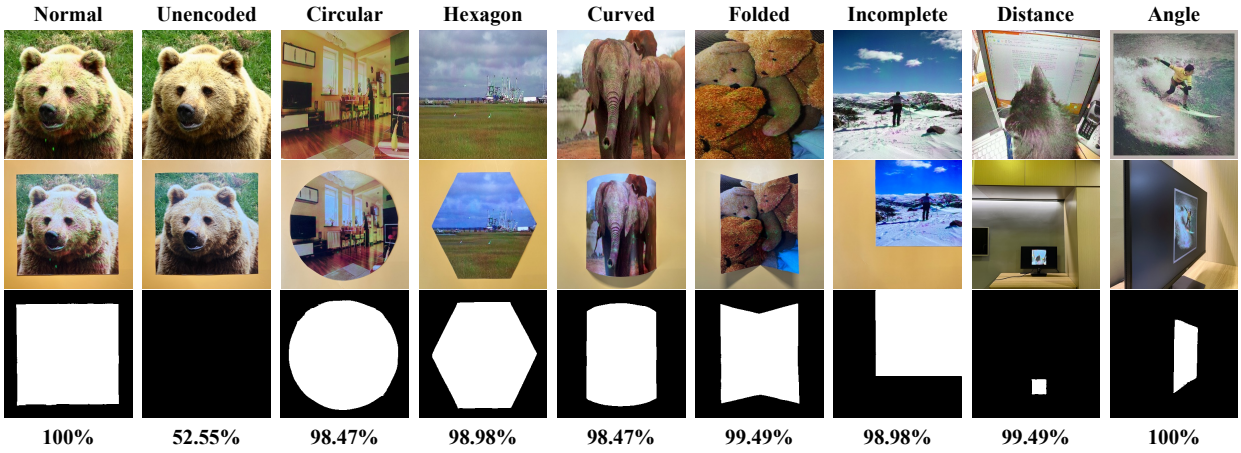

Fig. 3. The performance of *Aparecium* in the wild. We showcase different physical distortions in each column. The 2nd row is distorted images from printing-shooting or screen-shooting of the 1st row, respectively. The 3rd row displays the located watermarked areas, while the 4th row is the corresponding extraction accuracy.

TABLE VI
THE DECODING RESULTS UNDER DIFFERENT COMBINATIONS OF DISPLAY AND SHOOTING DEVICES.

Device	MacBook Pro 14	Dell S2421NX
iPhone 13 Pro	0.45%	0.40%
Redmi 12C	1.43%	1.29%

TABLE VII
THE IMPACT OF PATTERN DISTORTIONS.

Methods	PSNR	SSIM	BER
w/o Pattern Distortioin	31.13	0.9896	7.15%
w Pattern Distortioin	33.48	0.9883	5.03%

TABLE VIII
THE COMPARISON RESULTS BETWEEN DECODING-THEN-EXTRACTING MANNER AND DIRECT EXTRACTING MANNER.

Methods	PSNR	SSIM	BER
Direct Extracting Manner	32.98	0.9831	6.24%
Decoding-then-extracting Manner	33.48	0.9883	5.03%

TABLE IX
THE INFLUENCE OF λ_1.

λ_1	PSNR	SSIM	BER	JPEG			
				40	30	20	10
1	26.14	0.9619	1.36%	0.00%	0.00%	1.57%	44.62%
5	30.02	0.9800	3.76%	0.00%	2.30%	10.02%	47.28%
10	33.48	0.9883	5.03%	2.26%	17.05%	46.54%	47.96%

message length from the default 30 bits to 196 bits, the BER of HiDDeN [1] experience a significant increase. *Aparecium* obtains lower robustness to JPEG compression, which can be improved by adjusting hyper-parameter λ_1 (shown Section IV-F4). For blur and spatial distortion, *Aparecium* demonstrates superior robustness compared to the other methods.

D. Robustness Against Physical Distortions

In this experiment, we evaluate the decoding performance of screen and print shooting at different distances and angles. For screen-shooting, we display the image on a monitor and capture photos by mobile phone. The results for different distances and angles are presented in Table III and Table II, and our method achieves a lower BER than Offline-to-online [6]. However, at a distance of 40cm, we encounter severe moire patterns that affect both our method and the baseline. Subsequently, we conduct experiments on print-shooting at different distances and angles. As shown in Table IV and Table V, our method still achieves better performance. Moreover, Table VI shows the general robustness of our method on different devices.

E. Robustness in the Wild

To demonstrate the unexpected robustness of *Aparecium* in real-world scenarios, we capture a series of photographs using a handheld mobile phone. The Figure 3 showcases examples of the captured images with masks and decoding accuracy. In the 1st and 2nd columns, we capture both encoded and unencoded images, respectively. The results suggest that the locator works actually based on the embedded information rather than differences between the image and the background. According to the other results, *Aparecium* performs well in the wild, even in some severe cases, such as bending, folding, incomplete capture, long shooting distances, big shooting angles, etc.

F. Ablation Study

1) The Influence of Pattern Distortions: The robustness of patterns directly influences the subsequent extraction accuracy. In stage I, the introduction of pattern distortions affects the robustness of the pattern. Therefore, we undertook a comparative analysis to assess the impact of incorporating different pattern distortions on the ultimate robustness of the model. As shown in Table VII, enhancing the robustness of patterns by introducing stronger pattern distortions can simultaneously improve visual quality and watermark extraction capabilities.

2) The Effectiveness of the Incremental Decoding-then-extracting Manner: Current deep watermarking usually di-

TABLE X
THE VISUAL COMPARISON AMONG DIFFERENT STRATEGIES ON FOR LOCALIZATION.

Methods	PSNR	SSIM	IoU
Marks	29.42	0.9828	0.7826
Edge Loss	32.65	0.9849	0.8592
Edge Distortion	33.48	0.9883	0.9654

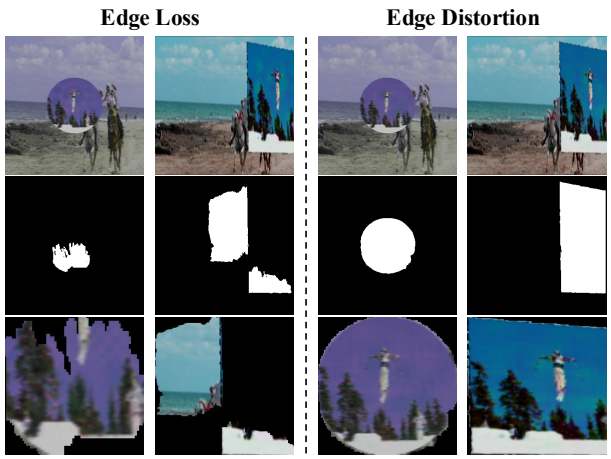

Fig. 4. The comparison of localization performance of different strategies. The first row represents distorted images, the second row represents the corresponding masks for localization, and the third row represents cropped images.

rectly extracts the bit-string message from the watermarked image. In comparison, we adopt a decoding-then-extracting manner to extract messages, namely, using a decoder to reveal patterns from captured images and then a message extractor to extract messages from the decoded patterns. To demonstrate the superiority of our incremental manner, we also try the direct extracting manner. As shown in the Table VIII, the decoding-then-extracting manner can achieve higher robustness with better visual quality.

3) The Effect of Edge Distortion: Without additional constraints for the localization, the encoder tends to append marks (*e.g.*, colorful taints) at the edges of images for localization purposes. In StegaStamp [3] and Offline-to-online [6], an edge loss is employed to eliminate such artifacts. In this paper, to mitigate the impact of these marks on image quality and to diffuse localization information across the image for improving robustness, we introduce edge distortion, specifically by rotation and translation, which involve randomly removing portions of the image. After applying edge loss or edge distortion, the marks are eliminated. To validate the effectiveness of our approach, we conducted a comparative analysis with the aforementioned techniques. In the Table X, our method exhibits superior visual quality and higher localization accuracy under combined distortion. Furthermore, we present the localization outcomes of superimposing the encoded image onto a background in Figure 4, which demonstrates our superior performance on localization.

4) The Influence of Hyper-parameter λ_1: During the training process in the third stage, we set $\lambda_1 = 10$. We also train models with $\lambda_1 = 5$ and $\lambda_1 = 1$ to investigate the impact of λ_1. As shown in the table Table IX, lower λ_1 leads to poorer visual quality but better extraction accuracy and robustness.

V. CONCLUSION

This paper proposes a novel deep watermarking framework, *Aparecium*, which is well-suited for real-world applications. Specifically, we investigate an efficient approach for processing watermark information and opt to use a series of transposed convolutions to diffuse the message into a pattern. In the decoding process, we distinguish pixel-wise distortion from spatial distortion, which significantly enhances the decoding performance. Finally, we adopt a three-stage training strategy to ensure training stability. Experiments demonstrate the proposed method achieves remarkable robustness against both digital distortions and physical distortions while preserving satisfied visual quality. We hope this work can shed some light on achieving robustness against complex physical distortions in more applications.

REFERENCES

[1] Jiren Zhu, Russell Kaplan, Justin Johnson, and Li Fei-Fei, "Hidden: Hiding data with deep networks," in *ECCV*, 2018, pp. 657–672.

[2] Eric Wengrowski and Kristin Dana, "Light field messaging with deep photographic steganography," in *CVPR*, 2019, pp. 1515–1524.

[3] Matthew Tancik, Ben Mildenhall, and Ren Ng, "Stegastamp: Invisible hyperlinks in physical photographs," in *CVPR*, 2020, pp. 2117–2126.

[4] Han Fang, Dongdong Chen, et al., "Deep template-based watermarking," *TCSVT*, 2020.

[5] Liuhao Zhu, Yixiang Fang, Yi Zhao, Yi Peng, Junxiang Wang, and Jiangqun Ni, "Lite localization network and due-based watermarking for color image copyright protection," *TCSVT*, 2024.

[6] Jun Jia, Zhongpai Gao, Dandan Zhu, Xiongkuo Min, Guangtao Zhai, and Xiaokang Yang, "Learning invisible markers for hidden codes in offline-to-online photography," in *CVPR*, 2022, pp. 2273–2282.

[7] Shumeet Baluja, "Hiding images in plain sight: Deep steganography," *NeurIPS*, vol. 30, 2017.

[8] Xiyang Luo, Ruohan Zhan, Huiwen Chang, Feng Yang, and Peyman Milanfar, "Distortion agnostic deep watermarking," in *CVPR*, 2020, pp. 13548–13557.

[9] Tu Bui, Shruti Agarwal, and John Collomosse, "Trustmark: Universal watermarking for arbitrary resolution images," *arXiv preprint arXiv:2311.18297*, 2023.

[10] Minzhou Pan, Yi Zeng, Xue Lin, Ning Yu, Cho-Jui Hsieh, Peter Henderson, and Ruoxi Jia, "Jigmark: A black-box approach for enhancing image watermarks against diffusion model edits," *arXiv preprint arXiv:2406.03720*, 2024.

[11] Tom Sander, Pierre Fernandez, Alain Durmus, Teddy Furon, and Matthijs Douze, "Watermark anything with localized messages," *arXiv preprint arXiv:2411.07231*, 2024.

[12] Han Fang, Zhaoyang Jia, Zehua Ma, Ee-Chien Chang, and Weiming Zhang, "Pimog: An effective screen-shooting noise-layer simulation for deep-learning-based watermarking network," in *TCSVT*, 2022, pp. 2267–2275.

[13] Xuebin Qin, Zichen Zhang, et al., "U2-net: Going deeper with nested u-structure for salient object detection," *Pattern recognition*, vol. 106, pp. 107404, 2020.

[14] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, "U-net: Convolutional networks for biomedical image segmentation," in *MICCAI*. Springer, 2015, pp. 234–241.

[15] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie, "A convnet for the 2020s," in *CVPR*, 2022, pp. 11976–11986.