Fixing the Double Agent Vulnerability of Deep Watermarking: A Patch-Level Solution Against Artwork Plagiarism

Yuanjing Luo, Tongqing Zhou, Shenglan Cui, Yunfan Ye, Fang Liu, and Zhiping Cai, *Member, IEEE*

***Abstract*— Increasing artwork plagiarism incidents stresses the urgent need for proper copyright protection on behalf of the creators. The latest development in this context focuses on embedding watermarks via deep encoder-decoder networks. However, we find that deep watermarking has a serious vulnerability on its robustness when facing deliberate plagiarism. To manifest it, we construct an attack that misuses watermarking encoder as a plagiarism lookout for bypassing copyright detection. As a remedy, we propose a patch-level deep watermarking framework (DIPW) to retain copyright evidence in essential patches with plagiarism resistance, inspired by a user study observation that subject elements in artworks are the principal plagiarism entities. Technically, DIPW adaptively finds the embedding patches by identifying a subset of non-overlapping and feature-rich objects; and tailors the model with dual-distortion losses and adversarial plagiarism noise injection for robustness. Experimental results demonstrate the superiority of DIPW in facilitating better robustness, secrecy, and imperceptibility with acceptable time burden.**

***Index Terms*— Deep watermarking, artwork copyright protection, plagiarism resistance, convolutional neural networks.**

I. Introduction

THE pervasive network access has significantly accelerated artworks' online propagation, particularly in the form of high-quality images (e.g., photographs and paintings). These digital products are usually copyrighted, even taken as non-fungible tokens, for conveying the efforts or thoughts of the owners [1]. Yet, the widely spread also exposes them to increasing risks of plagiarism [2], which, even though explicitly forbidden by the law, can hardly be convicted due to the lack of evidence.[1]

To mitigate the plagiarism issues for copyright protection, existing efforts rely on either passive plagiarism detection [3], [4], [5], [6] or active watermark embedding [7], [8]. The former line of work performs matching or classification for a target artwork based on a reference copyrighted image dataset, which is not usually available or necessarily contains a 'preimage' for the artwork. On the other hand, exclusive watermarks are traditionally constructed as the unique information extracted via image transformation [9], [10], [11], but may be criticized to cause significant visual distortion that degrades the viewing experience. For imperceptible information hiding and plagiarism forensics, deep watermarking [8], [12], [13], [14], [15], [16], [17], [18], [19], [20] has come to the spotlight recently. Based on end-to-end training, it tailors an encoder-decoder framework that conforms to symmetric watermark embedding and extraction.

Although providing an imperceptible way for copyright statement, we find that deep watermarking could, unfortunately, *be turned into a 'double agent' when facing artwork plagiarism*. Like an attacker who tries to launch oracle attacks without knowledge about the algorithm only by using watermarked content [21], in practice, a wise plagiarizer wouldn't honestly duplicate the slightly distorted image (i.e., transmission noise), as assumed in existing proposals. Instead, as revealed by our user study with 500 designers, artwork plagiarism is usually characterized by deliberate processing of the original artwork, including cropping (the principal form), stretching, rotation, etc. Repeating these processes provides the plagiarizer with valuable knowledge regarding the detection algorithm operation. Wherein, the encoder of deep watermarking can be misused by the plagiarizer to figure out whether the watermark is within the detection region and then guide the adversarial processing toward invalid watermark extraction.

Specifically, we propose an iterative challenge-and-response attack to manifest such concerns,[2] as shown in Fig. 1. First, a watermark detection classifier is constructed by tailoring a dedicated deep model based on the input-output pairs of various public-available deep watermarking encoders. In this way, the encoders for copyright embedding are tuned into plagiarizers' lookout (with >98.5% watermark detection accuracy). Then the plagiarizer could progressively modify the

Manuscript received 16 February 2023; revised 2 May 2023 and 17 June 2023; accepted 10 July 2023. Date of publication 17 July 2023; date of current version 7 March 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62102425, in part by the Science and Technology Innovation Program of Hunan Province under Grant 2022RC3061 and Grant 2021RC2071, and in part by the Postgraduate Research and Innovation Project of Hunan Province under Grant CX20220049. This article was recommended by Associate Editor R. Du. *(Yuanjing Luo and Tongqing Zhou contributed equally to this work.) (Corresponding author: Fang Liu.)*

Yuanjing Luo, Tongqing Zhou, Yunfan Ye, and Zhiping Cai are with the College of Computer, National University of Defense Technology, Changsha 410073, China (e-mail: yjluo@nudt.edu.cn; zhoutongqing@nudt.edu.cn; yeyunfan@nudt.edu.cn; zpcai@nudt.edu.cn).

Shenglan Cui and Fang Liu are with the School of Design, Hunan University, Changsha 410082, China (e-mail: cuisl@hnu.edu.cn; fangl@hnu.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCSVT.2023.3295895.

Digital Object Identifier 10.1109/TCSVT.2023.3295895

[1] www.creativebloq.com/features/how-can-designers-deal-with-plagiarism

[2] Traditional oracle attack [22] cannot be implemented for such copyright attacks as watermark decoding is not accessible.

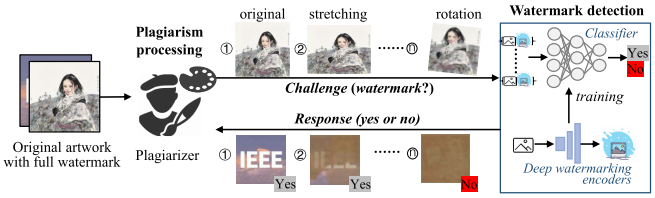

Fig. 1. Iterative challenge-and-response attack based on the double agent vulnerability for artwork plagiarism.

artwork image and ask the detector whether a watermark can be recovered after each move. Such an attack constitutes serious robustness/double agent vulnerability to existing deep watermarking techniques.

This work focuses on designing robust watermarking against deliberate artwork plagiarism. The basic design intuition comes from an observation in our exploratory study on plagiarism skills, that is, *subject elements usually attract viewers' attention quickly and are favored by plagiarizers*. Arguably, the image patches of the subject elements (e.g., characters) would be retained during plagiarism, otherwise, the plagiarism (e.g., copying background) is meaningless and cannot be effectively judged [23]. With this insight, we propose a patch-level deep watermark embedding framework (DIPW) for robustness against plagiarism. Given an artwork, DIPW adaptively finds the highlighted patches as embedding masks by fusing high-level object detection and low-level density estimation. We realize that a powerful plagiarizer may manage to bypass DIPW by brutally playing the challenge-and-response attack on different patches and performing plagiarism processing on the watermark-retained patches. For this, besides adopting image and watermark distortion losses, we introduce robustness-aware training in DIPW to explicitly tailor it with common distortions and plagiarism operations.

Interestingly, our patch-level design requires no resizing and maintains the overall image representation, thus endowing DIPW with better imperceptibility compared with existing full-image embedding techniques. Overall, we make the following contributions:

- We discover and implement the double agent vulnerability of misusing deep watermarking encoders to detect watermarks and bypass watermark extraction with deliberate distortions. This vulnerability can also be treated as a generic tool for watermarking imperceptibility evaluation.
- Our user study highlights the principal plagiarism skill of copying the subject elements in artworks. A dedicated patch-level framework is designed with this insight by jointly accommodating adaptive patch-finding and various noise resistance for watermarking robustness.
- Experimental results on 4 datasets demonstrate DIPW's better robustness (12%↓ watermark distortion rate), secrecy (77.6%↓ detection rate), and imperceptibility (25%↓ image distortion rate), compared with three well-studied deep watermarking methods.

II. Related Work

A. Plagiarism Detection

This form of work retrieves visually similar images via matching or classification to detect plagiarism. Digital artwork share sites (e.g., DeviantArt Protect [3]) and search engines (e.g., Tineye [4]) can help us to discover identical or partially modified images, but the results are generally not very satisfactory. Plagiarized-Search-Net (PS-Net) [5] is considered as the first specialized work for plagiarized retrieval tasks. It can identify plagiarized clothes that are usually modified in a certain region on the original design. Attribute-Specific Embedding Network (ASEN) [6] jointly learns multiple attribute-specific embeddings in an end-to-end manner for better representation of fine-grained similarity. However, *these approaches rely on copyrighted image dataset for detection, which is usually unavailable, and thus inscalable to general copyright protection practice*.

B. Watermark Embedding Techniques

Watermarks embedded into the image can be used for copyright statement [24]. Traditional watermarking algorithms typically embed information using image transformations, such as DCT [25], [26], DWT [27], [28], etc. Yet, such algorithms are poorly scalable and difficult to be reproduced due to the complicated encoding mechanisms [29], [33], [34]. Due to the subjectivity and inflexibility of traditional algorithms, watermarking methods combined with deep learning have gradually occupied the mainstream of this field, such as introducing auxiliary enhancing and classification subnetworks [30], strength factors [31], and texture value [32]. However, *these algorithms are still not suitable for plagiarism detection as their robustness comes mainly from the specific attack-resistant design that is vulnerable to unknown attacks*. The straightforward way of designing an algorithm for each type of plagiarism action and the possible combination is not feasible [11].

Recently, the encoder-decoder framework has been of interest for watermarking with impressive results by jointly training the encoder and decoder with various training strategies. HiDDeN [12] is the first deep learning solution for 'robust' image watermarking. Given an input message and cover image, the encoder produces a visually indistinguishable encoded image from which the decoder can recover the original message. Based on it, several excellent deep watermarking schemes have been proposed [13], [17], [19], [35], [36], [37]. Baluja simultaneously trained deep neural networks to create the hiding and revealing processes, realizing hiding a full-color image inside another of the same size with minimal quality loss to either image [38]. Liu et al. proposed a novel two-stage separable deep learning framework for practical blind watermarking, improving the decoder's robustness [14]. Luo et al. presented a distortion-agnostic watermarking framework, where the image distortion is not explicitly modeled during training, and the robustness comes from adversarial training and channel coding [8]. Jia et al. propose a novel invisible information-hiding architecture for display/print-camera scenarios, hiding information in a sub-image rather than the entire image [39]. Adversarial learning is combined with the attention mechanism in deep watermarking to help to generate a better embedded image by endowing different importance to different pixels [15], [16]. These schemes

Fig. 2. Examples for common plagiarism actions.

all require adaptive encoding of the watermark through the cover image, collectively known as Dependent Deep Hiding (Ddh) [20]. To simplify the encoder-decoder framework, the meta-architecture of Universal Deep Hiding (Udh) [18] disentangles the encoding of the watermark from the cover image, explored the possibility to hide an image in a cover-agnostic manner, opening the possibility for future work. Yet, existing efforts on deep watermarking would face serious vulnerability in front of artwork plagiarism, as we will reveal in the next section. *Briefly, deliberate plagiarism processing actions and the side-channel information from deep watermarking encoders will together render existing watermark embedding techniques vulnerable.*

III. Watermarking Vulnerability on Plagiarism

To dig into the vulnerability of deep watermarking in front of plagiarism, we first show the deliberate image processing of artwork plagiarism with a user study and then construct a technical attack to manifest the concern.

A. Understanding Artwork Plagiarism

In practice, plagiarism is very complicated and occurs in a wide variety of forms [5]. To dig into the patterns of image plagiarism, we initially conduct thematic research on image plagiarism by organizing a focus group and issuing a questionnaire[3] to ask the following question: "Different from simple duplication or network transmission, what skills or processing on images are we expecting when talking about plagiarism?"

Before participation, each participant is informed of the intention of this user study. We have 20 students majoring in design in a focus group. During the discussion of the focus group, five plagiarism processing actions are raised, including image cropping, image stretching, adding or deleting patterns (e.g., covering), color adjustment, and angle adjustment (a.k.a., rotation), wherein cropping is identified (9/10) as the principal form. Fig. 2 gives an example for each form.

We continue to collect 207 questionnaires with anonymous responses from different participants, more than 50% of which are students majoring in design or engaged in design-related fields. Given the above processing actions, we study typical plagiarism skills that the participants have experienced or would take as plagiarism. From the statistics, *subject elements copy* (73.47% positive[4]) is significantly preferred among other options, e.g., composition copy (55.1% positive), color copy (28.57% positive). By consulting (online and face-to-face) the reasons for such plagiarism characteristics with designers, they put that one would try to reuse the most interesting part in a design and minimize the efforts of such processes if s/he intended to produce his/her own work via plagiarism.

Remark 1: Subject elements (e.g., the character in Fig. 3) in an artwork imply the design highlights, which would attract a viewer's attention quickly with the inherent complex texture and color combinations and are thus the favored plagiarized contents.

The cropping action is the basic way to attain the subject elements, as stated in a representative comment of one participant (#16) in the survey, *"If I were a plagiarizer, I would crop the interesting parts from artworks and integrate them into my work"*. Since an encoded watermark is cascaded on all regions of a cover image, cropping it during plagiarism would cause significant global information loss with the watermark being hard to be recovered.

B. Attack on Deep Watermarking

Existing deep watermarking could, unfortunately, be turned into a 'double agent' when facing artwork plagiarism. The underlying reason is that deep watermarking encodes a watermark as high-frequency (HF) information and embeds it straightforwardly onto the full cover image. Given that such encoding and its decoding are opulent to users, a plagiarizer can simply run the decoder and manually check whether the output is a watermark to decide if an artwork has copyright protection (i.e., as if s/he wanted to check copyright like the watermark extraction module in Fig. 3).

Note that this type of oracle attack is not a threat to copyright protection because the watermark decoder is not publicly available [40]. To technically implement such a vulnerability, we first build a generic deep watermark detector. Note that existing steganalysis is designed for specific image variations and yields low detection accuracy of 50% for deep watermarking [12]. Instead, we construct a watermark classifier with ResNet-34 [41] as the backbone. It is tailored based on the input-output samples (i.e., cover image and watermark-embedded image pair) generated and combined from various public-available deep watermarking encoders (e.g., Ddh [12], [15], Udh [18]). Totally 24,752 images from MS COCO are used for training, half of which are with embedded watermarks (positive samples) and the rest are cover images (negative samples). The testing results on 2000 artwork images show the classifier can determine whether an image contains a watermark with an accuracy of over 98.5%. A smart plagiarizer could then misuse such a dedicated classifier as its lookout for performing plagiarism actions, to ensure as many interesting contents are retained and no valid watermark can be recovered from its modified copy.

Remark 2: A plagiarizer can launch a challenge-and-response attack by iteratively modifying the artwork image and asking the classifier whether a watermark can be recovered after each move. This in turn, unfortunately, makes deep watermarking a tool useful for plagiarism (see Fig. 1).

[3]Questionnaire link: https://www.wjx.cn/vm/YHsyFXH.aspx

[4]Positive means a score>3 in our 5-level Likert scale.

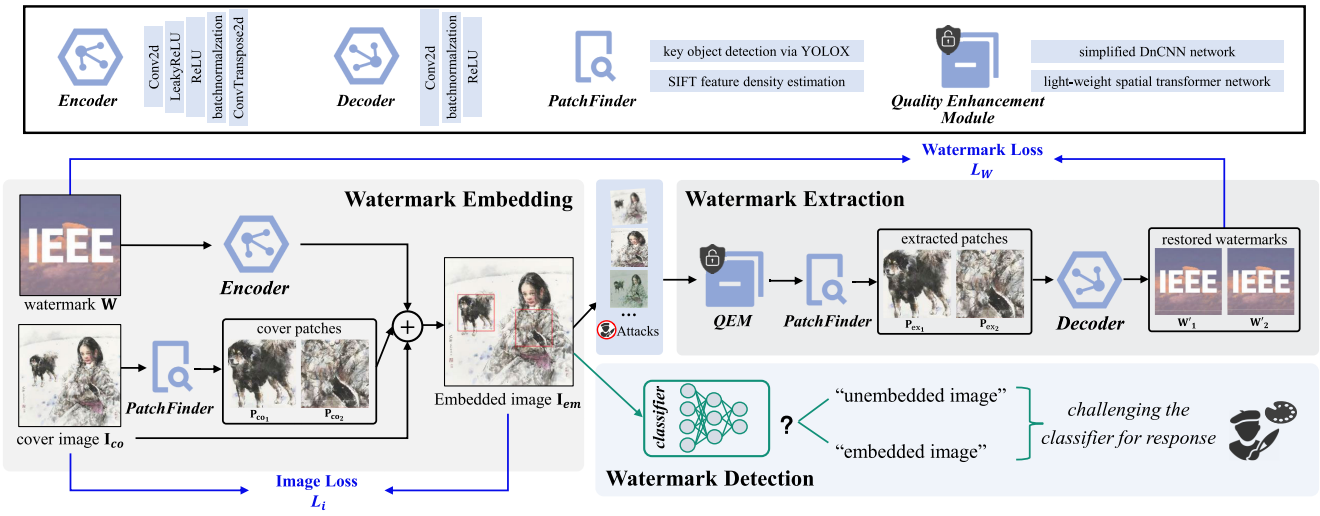

Fig. 3. The framework of DIPW. The watermark is subtly embedded in images' highlighted patches and can be accurately restored even under plagiarisms.

IV. Design of DIPW

A. Motivation and Preliminary Validation

Inspired by the above insights, *we are motivated to embed the watermark in patches that show higher interestingness*, as these patches would be retained during plagiarism. We argue that copying some minor elements (e.g., the background) is meaningless for plagiarizers and cannot be effectively judged as plagiarism [23]. We consider this the paradox of plagiarism, i.e., some essential elements must be retained during plagiarism. Intuitively, patch-level embedding is expected to block the challenge-and-response attack from revealing watermarks for the plagiarizer. However, we also realize that a powerful plagiarizer may manage to brutally play the challenge-and-response attack on different patches of its targeted artwork and perform plagiarism processing on possible watermark-retained patches. We emphasize that adversarial training on such extreme actions would help to mitigate them.

To observe the feasibility of this idea, we further fine-tune a patch-level watermark classifier with 5000 positive samples that embed watermarks on some random regions of images (e.g., with 1/4 the sizes of cover images). The test results on another 5000 images showcase poor detection accuracy of this classifier (i.e., accuracy <6%), which means an invalid double agent in the patch-level context (see §VI for more details).

B. Framework Overview

Fig. 3 demonstrates the overall framework of the proposed DIPW. In the watermark embedding phase, a cover/original image $\mathbf{I}_{co}$ is fed to PatchFinder, yielding two highlighted patches: $\mathbf{P}_{co_1}$ and $\mathbf{P}_{co_2}$. A watermark $\mathbf{W}$ is encoded by Encoder and then added to $\mathbf{P}_{co_1}$ and $\mathbf{P}_{co_2}$ in $\mathbf{I}_{co}$, resulting in an embedded image $\mathbf{I}_{em}$. We mark the embedded patches $\mathbf{P}_{em1}$ and $\mathbf{P}_{em_2}$ in $\mathbf{I}_{em}$ with red boxes for visualization. In the watermark extraction process, the embedded image $\mathbf{I}_{em}$ is first fed into the QEM to eliminate the influence of possible distortions brought about by plagiarism attacks. Then, PatchFinder is used to predict and extract $\mathbf{P}_{ex_1}$ and $\mathbf{P}_{ex_2}$ from $\mathbf{I}_{em}$, and $\mathbf{W}'$ can be restored from $\mathbf{P}_{ex_1}$ or $\mathbf{P}_{ex_2}$ by Decoder. The Encoder and Decoder are jointly trained to minimize loss $\mathcal{L}_i$ from the difference between $\mathbf{I}_{co}$ and $\mathbf{I}_{em}$ and loss $\mathcal{L}_w$ from $\mathbf{W}$ and $\mathbf{W}'$. A quality enhancement module (QEM) is utilized between the encoder and decoder to eliminate the influence of distortion/small perspective changes in $\mathbf{I}_{em}$ brought about by plagiarism. QEM consists of a DnCNN [42] and a spatial transformer network (STN) [43], which should be trained during Stage II training for better performance. For a distorted $\mathbf{I}_{em}$, the former can help it to keep spatial transform invariance, and the latter can efficiently provide denoising. Where the Batch-Normalization layer in the DnCNN is removed for a lighter and suitable structure. For watermark detection, the trained classifier determines whether the embedded image $\mathbf{I}_{em}$ contains a watermark from the perspective of active plagiarizers.

C. How to Find Proper Patches

1) Basic Idea: As mentioned, subject elements that quickly capture viewers' attention are usually retained during plagiarism. However, it is a very subjective task to find these image patches, a.k.a., highlighted patches that correlate with the ones that the plagiarizer actually operates. Existing studies pointed out that elementary shapes (e.g., rectangles, ellipses, etc.) are commonly seen as the basis for representing things in artworks [44], and objects (a more generalized term of elements) in foreground and regions with contrasting colors usually gain more visual attention [45], [46], [47]. Given these insights, we believe it is helpful to rely on object detection [48] and hand-crafted feature points [49] (e.g. Harris, SIFT, SURF, ORB) to locate images' highlight patches. Among them, SIFT feature points are adopted by us due to their superior performance, see Tab. I and Fig. 4. Meanwhile, we have to note that: 1) the interesting objects that exist in the artwork are almost sure to occupy an irregular region. To ensure that the network can batch the embedded region during training and then guarantee the extraction of the watermark, we need to constrain the size of $\mathbf{P}_{co}$; 2) an image's highlighted patches

TABLE I
IMAGE KEY POINTS DETECTORS COMPARISON

Feature Points	Speed	Angle invariance	Scale invariance
Harris	low	medium	low
SIFT	medium	high	high
SURF	high	medium	medium
ORB	high	high	medium

(a) Detecting objects in the image (b) Locating SIFT feature points in the image

Fig. 4. (a) Object detection can help quickly parse the foreground of an image and (b) SIFT feature points can help locate the highlights by representing the richness of color texture in different regions.

are non-invariant when it is modified due to the instability of object detection and feature points. We assume that embedding the watermark in multiple non-overlapping patches can improve a watermark's survivability.

2) The Design of PatchFinder: We design PatchFinder based on high-level key object detection and low-level SIFT feature density estimation to find proper patches. It exploits the merits of different patches on attention during object detection by using an attention mechanism to elaborate eye-catching elements. Given a cover image $\mathbf{I}_{co}$, PatchFinder first divides $\mathbf{I}_{co}$ into patches according to the object detection bounding box or evenly into nine patches if no object is detected:

$$\mathbf{P} = Divide(\mathbf{I}_{co}) = (\mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_n), \quad 1 \leq n. \tag{1}$$

Next, PatchFinder estimates the density of sift feature points in $\mathbf{P}_i$, $i \in [1, n]$ and selects *top-k* patches in descending order as candidate patches $\mathbf{P}_{ca} = (\mathbf{P}_{ca_1}, \mathbf{P}_{ca_2}, \ldots, \mathbf{P}_{ca_k}), k \leq n$ where $Sift(\mathbf{P}_{ca_1}) > Sift(\mathbf{P}_{ca_2}) > \ldots > Sift(\mathbf{P}_{ca_k})$ and $Sift(\cdot)$ indicates the operation of calculating the number of sift feature points. With this step, we get k candidate patches with varying sizes that should be uniformly sized to fit the size requirements of the network's batchsize. Therefore, we use the centroid of $\mathbf{P}_{ca}$ as the basis and cut size-fixed patches from $\mathbf{I}_{co}$ as the e highlighted $\mathbf{P}_{co}$ for watermark embedding:

$$\mathbf{P}_{co_i} = Centercut_{m \times m}(\mathbf{P}_{co} | \mathbf{P}_{ca_i}), \quad i \in [1, k]. \tag{2}$$

where $Centercut(\cdot)$ indicates the operation of cutting based on the centroid, $m \times m$ denotes the patch size. Since embedding watermarks to overlapping patches would cause blurs in extraction, we further find the non-overlapping patch subset with the maximum coverage:

$$\max(\sum_{i=1}^{k} |\mathbf{P}_{co_i}|) \quad s.t., \mathbf{P}_{co_i} \cap \mathbf{P}_{co_j} = \oslash, \quad (i \neq j \in [1, k]), \tag{3}$$

where $|\mathbf{P}_{co_i}|$ denotes the patch area. The pipeline of PatchFinder is shown in Fig. 5, where $k = 2$, and the rightmost column shows that the obtained patches effectively match the desired attention-grabbing regions (visualized by ResNet-50+Grad-Cam [50]), namely, the attained $\mathbf{P}_{co}$ is the subjectively interesting region during plagiarism. Note that when extracting the watermark, all detectable regions are treated as highlighted patches $\mathbf{P}_{ex}$ to avoid inaccuracies in patch locating due to plagiarism, which is slightly different from the embedding process.

D. Robustness-Aware Training

1) Noise Layers: Adversarial learning on well-designed noise layers can contribute to the robustness of the embedded watermarks [51]. We purposefully adopt two types of noise layers between the encoder and decode.

a) Common distortions processing: These distortions have been considered in previous work [12] and [18].

- Dropout and Cropout undo some of the changes made by the encoder, producing the distorted image by combining pixels from the $\mathbf{I}_{co}$ and $\mathbf{I}_{em}$ with $p = 0.3$.
- Gaussian blurs $\mathbf{I}_{em}$ with a Gaussian kernel $\sigma = 2.0$.
- Crop produces a random square $H' \times W'$ crop of $\mathbf{I}_{em}$, the image sizes ratio $\frac{H' \times W'}{H \times W}$ is $p = 0.035$.
- JPEG compresses $\mathbf{I}_{em}$ with quality factor $Q = 50$.

b) Plagiarism action-incurred noises: As these processing actions are not completely predictable in real-world, the following adversarial samples (outlined in Fig. 2) are evenly divided for $\mathbf{I}_{em}$ in each training batch to cover all forms of noise.

- 80%-90% random Cropping.
- 110%-120% random Stretching.
- Rotation with random angles of 5° -10°.
- random-placed Covering with a 5×5 white patch.
- Color-changing with degrees randomly chosen from $(-5, +5)$.

2) Dual-Distortion Loss: The total loss function is composed of image distortion and watermark distortion.

a) Image loss: For the viewing quality, the embedded image should look visually similar to the cover image. This requires explicit embedding performance of the encoder and can be measured by the $L2$ distance between $\mathbf{I}_{co}$ and $\mathbf{I}_{em}$:

$$\mathcal{L}_i(\mathbf{I}_{co}, \mathbf{I}_{em}) = \|\mathbf{I}_{co} - \mathbf{I}_{em}\|_2^2 / (C \cdot H \cdot W)) \tag{4}$$

b) Watermark loss: For facilitating robust copyright evidence, the extracted watermark shouldn't have many changes compared with the embedded version (i.e., restoring performance). Accordingly, this loss term is defined as:

$$\mathcal{L}_w(\mathbf{W}, \mathbf{W}') = \|\mathbf{W} - \mathbf{W}'\|_2^2 / (C \cdot H \cdot W) \tag{5}$$

wherein, C, H, and W denote the number of channels, height, and width, respectively.

c) Total loss function: By jointly considering these two dimensions of distortion losses, we attain the optimization goal for the watermark embedding and extraction branch, minimizing the following loss:

$$\mathcal{L}_{total} = \lambda_1 \cdot \mathcal{L}_i(\mathbf{I}_{co}, \mathbf{I}_{em}) + \lambda_2 \cdot \mathcal{L}_w(\mathbf{W}, \mathbf{W}'), \tag{6}$$

where λ controls the relative weights of the losses.

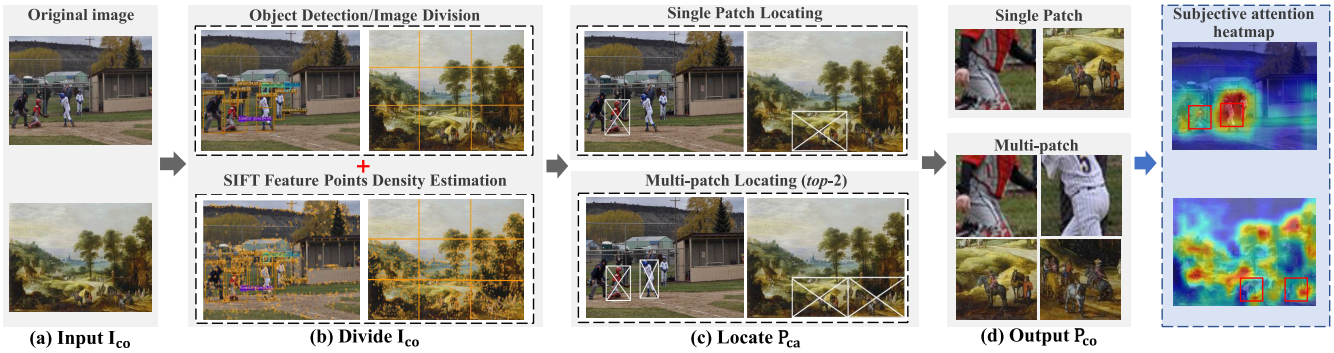

Fig. 5. Illustration of adaptively finding proper patches for watermark embedding, where the elaborated patches share large intersections with attention-grabbing regions of the cover images (indicated with heatmaps).

3) Two-Stage Training Strategy: To acquire a powerful encoder and fully exploit the potential of the decoder, we train our model in two stages.

Stage I: The end-to-end training phase autonomously embeds the watermark into the input image, and then directly restores it without any noise. The optimization objective is to minimize $\mathcal{L}_{total}$. As the encoder is trained without being affected by decoder training, which is able to maintain higher quality of the encoded image with better stability and faster speed of convergence.

Stage II: Through Stage I, we get an encoder responsible for watermark embedding and a pretrained decoder. After this, the noise processing is introduced, and the parameters of the encoder are no longer modified, but the decoder is fine-tuned depending on different noise based on the pretrained decoder. The optimization objective is to minimize $\mathcal{L}_w$. Note that loading the decoder weights obtained from the Stage I as the pretrained weights can significantly accelerate Stage II.

The training of DIPW is summarized as Algorithm. 1.

Algorithm 1 DIPW Training

Require:
Training set of cover images and watermarks
Ensure:
Trained Encoder and Decoder
Initialize Encoder and Decoder with random value.
Initialize PatchFinder with pre-trained YOLO-X.
if Stage I training **then**
  while $Step < max_{steps}$ **do**
    -Compute Patch: $\mathbf{P}_{co} = PatchFinder(\mathbf{I}_{co})$
    -Embed Watermark: $\mathbf{I}_{em} = Encoder(\mathbf{P}_{co}, \mathbf{W})$
    -Compute Patch: $\mathbf{P}_{ex} = PatchFinder(\mathbf{I}_{em})$
    -Extract Watermark: $\mathbf{W}' = Decoder(\mathbf{P}_{ex})$
    -Update Encoder and Decoder: minimize $\mathcal{L}_{total}$
  end while
end if
if Stage II training **then**
  while $Step < max_{steps}$ **do**
    -Add noise to $\mathbf{I}_{em}$
    -Enhance $\mathbf{I}_{em}$ via QEM
    -Compute Patch: $\mathbf{P}_{ex} = PatchFinder(\mathbf{I}_{em})$
    -Extract Watermark: $\mathbf{W}' = pre\text{-}Decoder(\mathbf{P}_{ex})$
    -Update QEM and Decoder: minimize $\mathcal{L}_w$
  end while
end if

V. Experiments

In experiments, we want to investigate two issues:

- DIPW performance experiments and the corresponding comparison with state-of-the-art methods.
- Ablation study.

where performance experiments are evaluated on five recognized axes: *robustness*, the degree of successful watermark recovery under noise; *secrecy*, the difficulty of detecting embedded watermarks; *imperceptibility*, the visual similarity between the cover/original images and the embedded images; *capacity*, the number of message bits which can be hidden per image bit; *efficiency*, the time cost on embedding and extracting watermarks.

A. Basic Setup

1) Datasets: The MS COCO and VOC 2007 training datasets are used for training DIPW, in which half is randomly selected as cover patches and the remained are watermark patches. The testing datasets include 8,000 cover images selected from Wiki Art [52] and the test sets of MS COCO and VOC 2007, and 8,000 watermarks in LOGO_DIPW (logo image sets collected online in this paper).

2) Implementation: DIPW is implemented with PyTorch 1.10.0, Intel(R) Core(TM) i7-11700K @ 3.60GHz, 32.00 GB RAM, and an Nvidia GeForce GTX 3080 Ti GPU is used for acceleration. We use the Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.99$, a learning rate of 0.0001 and a mini-batch of size 16 to train DIPW. The training patch size $C \cdot H \cdot W = 3 \times 512 \times 512$, and the number of total iteration is $10K$. For PatchFinder, the number of candidate patches $k = 2$, and the patchsize $m \times m = 128 \times 128$. For the loss, the corresponding weight factors in Stage I $\lambda_1 : \lambda_2 = 1 : 1.25$, while in Stage II $\lambda_1 : \lambda_2 = 0 : 1$.

3) Metrics: The main metrics includes: *Robustness* is measured by watermark distortion rate; *secrecy* is measured by detection rate; *imperceptibility* is measured by image distortion rate; *capacity* is measured by Bits Per Pixel (BPP); *efficiency* is measured by Seconds Per Image (SPI). The distortion rate

includes four evaluation indicators, i.e., Peak Signal to Noise Ratio (PSNR) [53], Structural Similarity (SSIM) [54], Learned Perceptual Image Patch Similarity (LPIPS) [55], Average Pixel Discrepancy (APD), and Bit Error Rate (BER):

$$PSNR(x, y) = 10\log_{10}(\frac{(MAX_I)^2}{MSE(x, y)}), \tag{7}$$

where MAX_I is the maximum possible pixel value of images x and y. $MSE(x, y)$ represents the Mean Squared Error (MSE) between images x and y.

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}, \tag{8}$$

where μ_x and μ_y represent the average grey values of images. Symbol σ_x and σ_y represent the variances of images. Symbol σ_{xy} represents covariance between images. C_1 and C_2 are two constants which are used to prevent unstable results when either $\mu_x^2 + \mu_y^2$ or $\sigma_x^2 + \sigma_y^2$ is very close to 0.

$$LPIPS(x, y) = \sum_l \frac{1}{H_l W_l} \sum_{h,w} \left\| w_l \odot (\hat{x}^l - \hat{y}^l) \right\|_2^2, \tag{9}$$

where $\hat{x}^l, \hat{y}^l \in \mathbb{R}^{H_l \times W_l \times C_l}$ for layer l is designated from unit-normalized feature stack in the channel dimension, which are extracted from L layers of VGG. vector $w^l \in \mathbb{R}^{C_l}$ scales the activations channel-wise. $w_l = 1 \forall l$ is computing cosine distance.

$$APD(x, y) = \ell_1(x, y) = (x - y)^2, \tag{10}$$

$$BER = \frac{n_{err}}{len(str)}, \tag{11}$$

where n_{err} is the number of error bits, $len(str)$ represents the length of hidden messages. Higher is better for PSNR and SSIM, and lower is better for APD, LPIPS, and BER.

4) Baselines: We choose several SOTA methods as baselines, including two well-studied deep watermarking benchmarks: HiDDeN [12] and Udh [18]; and a region-embedded method proposed by Jia et al. [39]. HiDDeN and [39] are dependent models, Udh and our DIPW are universal models. Note that the original HiDDeN [12] model can only embed messages. To make it consistent with the image watermark embedding in this paper, we also slightly modified its output into images and then re-trained the model.

B. Robustness

Strong robustness requires that the watermark extracted under noise be as close to the original watermark as possible (*low watermark distortion rate*) [38]. In real life, artworks are inescapably suffered noises (e.g., transmission distortions and plagiarisms), affecting the restoration of the watermark. In this section, we subject the model to varying considered distortions during testing to evaluate robustness in three main ways: "Comparison with HiDDeN and Udh", "Comparison with Jia et al. [39]", and "Against real-world attacks". Remark: 1) The embedded watermarks are lightly-colored logo images from LOGO_DIPW with *very low fault-tolerant rate*, and slight distortion will be evident, which can better expose the extraction effect of the watermark; 2) DIPW embeds the watermark in two patches, and the best-restored watermark is the result used for comparison.

1) Comparison With HiDDeN and Udh: In HiDDeN, Udh, and DIPW, the embedded watermarks are all color images, thus PSNR and SSIM are chosen as metrics in this section as they can best reflect the subjective perception of human eyes. The results are shown in Tab. II. For "Identity", "Cropout", "Dropout", "Gaussian" and "JPEG", it can be found that although DIPW is slightly inferior to HiDDeN and Udh in terms of distortion rate (as DIPW additionally requires patch localization, which introduces little instability), all three combined distortion trained models are equally robust and maintain good extraction performance (see Fig. 6). For "plagiarism actions", We observe that DIPW is significantly more robust: 1.44× and 1.41× increase over HiDDeN and Udh in PSNR, respectively, and 1.16× and 1.23× increase over HiDDeN and Udh in SSIM, respectively. This is thanks to two strategies: 1) DIPW only embeds the watermark in images' highlighted patch instead of the full image, results under cropping prove the superiority of DIPW, as cropping inevitably destroys the content of the full-embedded image rather than our highlighted patches. The robustness results of the target patch being attacked also favor the validity of our patch-level design; 2) DIPW is trained with specific plagiarism noise layers and allows highlighted patches to hold up well under multiple plagiarisms.

2) Comparison With Jia et al. [39]: In our approach, we chose the logo image as the watermark since the image is a more straightforward way to prove authorship, while in Jia et al. [39], the embedded watermark is the barcode. For a more intuitive comparison with Jia et al. [39], we try to embed barcode. By dividing the barcode into $m \times n$ patches with the content of 0 or 255, it can be equivalent to $m \times n$ bits of information via calculating the average value of each patch, e.g., classifying the output to 1 if the average value is higher than 128, otherwise 0. The embedded results are shown in Fig. 11 and Tab. II, where BER is as metric. We observe that, compared to Jia et al. [39], DIPW gains lower BER under most plagiarism attacks (except for rotation since image division is not sturdy to angle changes). This is because the embedding region of [39] is not the plagiarism preference region and the noise it considers focuses only on screenshots rather than plagiarism. In addition, the results imply that*our approach also supports barcode-based watermarking*. Remark that DIPW only trains the model with images, and retraining categorically for barcodes may result in improved performance.

3) Against Real-World Attacks: To better approximate the actual plagiarism of each image subjected to different actions, five more sophisticated plagiarism actions are also set up to test the performance (see Fig. 8). Experimental results demonstrate that DIPW is able to covertly embed the watermark with good reconstruction capability (average image SSIM=0.99) and effectively restore them (average watermark SSIM=0.87). This is because DIPW can find the artworks'soul-highlighted patch, which is most likely to be cropped and integrated into new artwork under plagiarism, i.e., the difference between the embedded patch and the extracted patch is minimal.

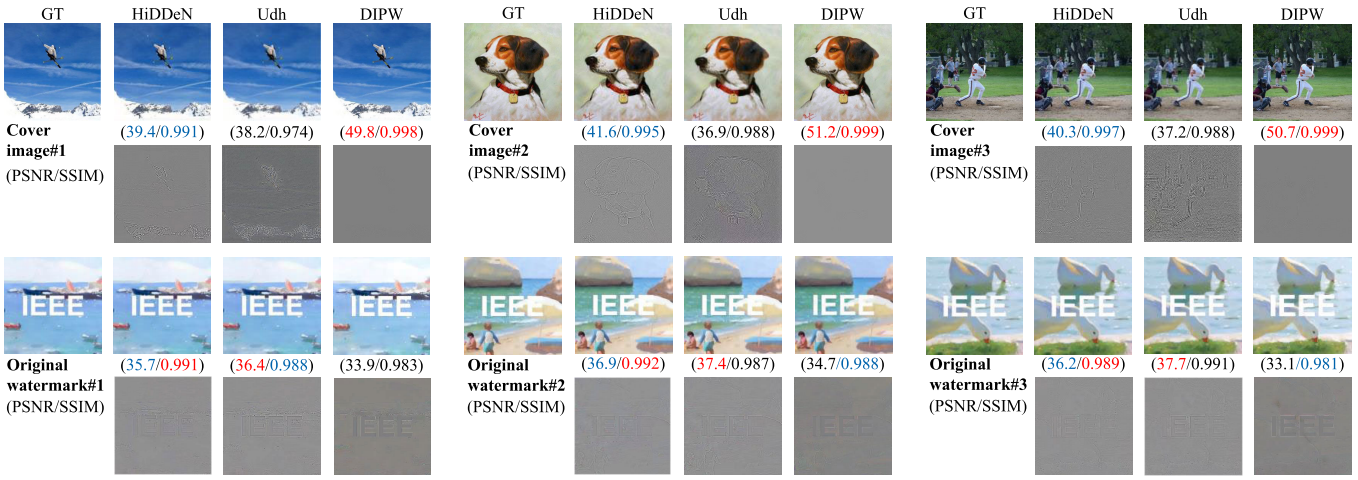

Fig. 6. Visual comparisons of embedded images and restored watermarks images of our DIPW and the comparison methods HiDDeN and Udh. The upper two rows show the embedded images, while the lower two rows show the restored watermarks of different methods.

TABLE II

ROBUSTNESS COMPARISON. "IDENTITY": NO NOISE, "DIPW (P)": EMBEDDED PATCH

Noise	PSNR (dB) ↑				SSIM ↑				BER (%) ↓ (length=256 bits)		
	HiDDeN	Udh	DIPW	DIPW (P)	HiDDeN	Udh	DIPW	DIPW (P)	Jia *et al.* [39]	DIPW	DIPW (P)
Identity	34.7	**35.9**	32.9	34.2	**0.992**	0.987	0.983	0.988	0	0	0
Cropout	26.5	**29.2**	26.7	29.1	0.891	0.896	**0.912**	0.877	20.5	6.8	**5.9**
Dropout	26.4	27.9	23.9	**28.1**	**0.868**	0.861	0.863	0.861	30.2	7.3	**6.7**
Gaussian	28.2	**29.7**	25.8	29.4	**0.871**	0.871	0.869	0.869	0	1.7	**0.7**
JPEG	**23.7**	21.1	21.4	21.3	0.721	0.74	**0.741**	0.736	20.6	15.5	**10.4**
Cropping	14.1	13.4	**23.6**	13.6	0.582	0.554	**0.884**	0.552	**13.5**	15.1	27.8
Streching	20.9	20.5	**28.7**	20.4	0.875	0.875	**0.985**	0.877	4.7	**0**	0
Covering	20.7	23.3	**28.3**	21.9	0.939	0.868	**0.972**	0.891	10.5	**0**	28.1
Rotation	10.6	11.7	**20.6**	10.3	0.683	0.615	**0.851**	0.614	54.8	**48.6**	50.2
Coloring	22.9	22.3	**27.5**	24.1	0.911	0.855	**0.967**	0.855	0	0	0
Average	22.8	23.5	**25.9**	23.2	0.833	0.812	**0.902**	0.812	15.4	**9.5**	12.9

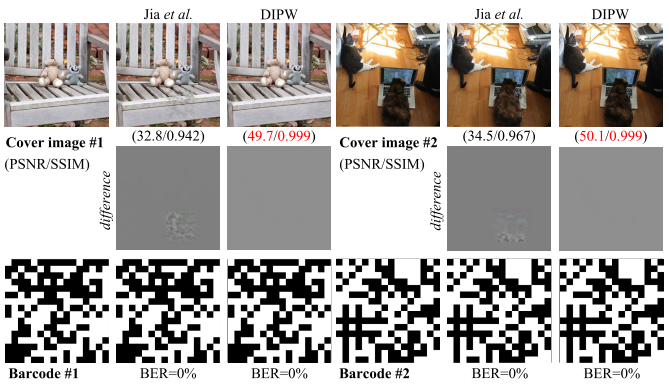

Fig. 7. Visual comparisons of embedded images and restored watermarks images of our DIPW and the comparison method Jia et al. [39]. The upper two rows show the embedded images, while the lower one show the restored barcodes of different methods.

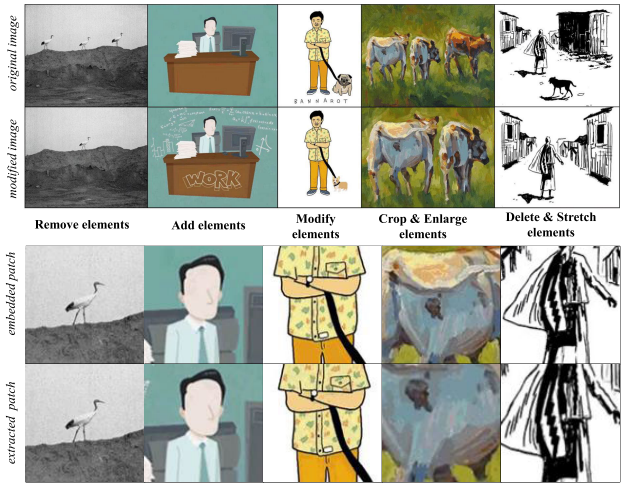

Fig. 8. Samples of artwork images with different painting styles and common plagiarism processing actions. The key to the success of DIPW is that the embedded patches are similar to the extracted patches from modified images.

C. Secrecy

1) Classifier as an Indicator of Secrecy: High security requires that the embedded watermark cannot be detected by other tools (*low detection rate*) to avoid plagiarists launching iterative challenge-and-response attacks. Intuitively, judging watermark existence is a prerequisite of watermark extraction under traditional encoder-decoder watermarking framework. In practice, since the decoder is undisclosed to the public as technique property, plagiarizers cannot directly use the decoder to guide their plagiarism operations. Instead, one could train

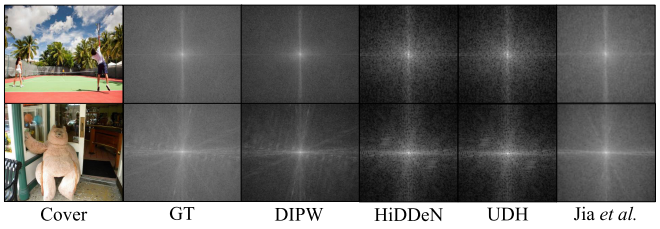

Fig. 9. Fourier analysis of cover image and embedded images from different methods, showing that watermark embedding causes frequency discrepancy.

a classifier with the input-output pair of the encoder to tell whether a watermark exists in an image.

Compared with existing steganalysis, the well-trained classifier is a more effective detector for deep watermarking, and its test-set accuracy is the detection rate, which is 85.2%, 83.7%, 30.4% and **6.8%** on HiDDeN, Udh, [39], and DIPW, respectively, indicating that the watermarks embedded via DIPW are hard to be detected. This is because: 1) DIPW jointly exploits dual-distortion-loss of patch and watermark in optimization and watermark-sensitive classifier; 2) as mentioned, existing deep watermarking techniques rely on the image's HF information [16], and the resulting frequency discrepancy is the key to the success of classifier. While DIPW does not much affect the frequency domain of the whole cover image since we only embed the watermark in the images' highlighted patch, thus having better undetectability. Note that although [39] is also region-embedded, the selected sub-image position can be the same each time, so its frequency domain information varies regularly and has a higher probability of being detected than DIPW. As shown in Fig. 9, the performed Fourier analysis [56] confirms this conclusion.

2) Watermark Decoder Cross-Test: Since DIPW and Udh both belong to the universal model, there is a certain chance that the decoders will be generic. Therefore, we perform a cross-test for Udh and DIPW, the output of the Udh encoder is set as the input of the DIPW decoder, and the output of the DIPW encoder is set as the input of the Udh decoder. As shown in Fig. 10, from the first two columns, it can be seen that the embedded image via the UDH encoder narrowly evades the DIPW decoder, and from the last two columns, it can be seen that the embedded image via the DIPW encoder are completely unhackable by the UDH decoder since the UDH decoder cannot locate the embedded patches. Unsurprisingly, DIPW is more secure.

3) Watermark Extraction After Challenge-and-Response Attack: For the traditional framework, plagiarizers could implement plagiarism operations until the classifier cannot detect any watermarks (i.e., challenge-and-response attack), in which case the decoder would also fail to recover the watermark after such operations (if the classifier is well trained). To test the impact of such attacks on watermark extraction, we have presented an illustrative example where the embedded images of Udh and DIPW undergo four challenge-and-response attacks, each yielded a plagiarized image that can pass the watermark detector. Then we show the extracted watermarks of Udh decoder and DIPW decoder on these

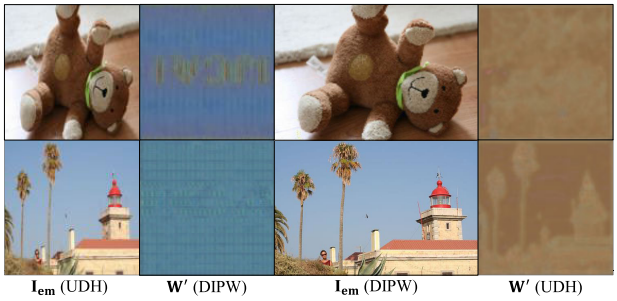

Fig. 10. Cross-test with UDH and DIPW. The four columns from left to right indicate Embedded image from UDH encoder; Restored watermark from DIPW decoder; Embedded image from DIPW encoder; Restored watermark from UDH decoder.

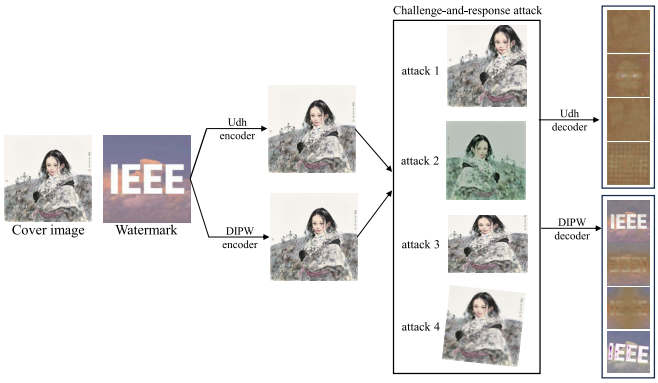

Fig. 11. Watermark extraction of the Udh decoder and the DIPW decoder for the deliberate plagiarized image (i.e., images processed with challenge-and-response attack to be classified as of no watermark).

four images. As shown in Fig. 11, when the classifier considers an embedded image has no watermark after plagiarism operations, the Udh decoder cannot recover visible watermarks. In contrast, DIPW could still recover visible watermark after the plagiarizer thinks s/he has succeeded (no watermark detected). We owe this secrecy superiority to the interestingness-aware patch-based embedding in DIPW, which misleads the classifier to believe no watermark exists.

D. Imperceptibility

High imperceptibility requires that the embedded image is highly similar to the cover image (*low image distortion rate*). In addition to objective metrics, we also invited 50 volunteers to evaluate the subjective quality of embedded images via a user study. In our experiment, five images with watermarks embedded by HiDDeN, Udh, [39], DIPW (P), and DIPW, respectively, and one original image are shown to the volunteers. They are required to mark the suspected abnormal image as 1 and the others as 0 without knowing the attributes of these images. We count the final Mean Opinion Scores (MOS) [57] are results. Fig. 6, Fig. 11 and Tab. III visualize the results of these qualitative comparisons. It can be found that the images embedded via DIPW are closer to the cover images, 1.1×, 1.5×, and 1.28× increase over HiDDeN, Udh, and [39], respectively (average of PSNR and SSIM). This is because 1) both HiDDeN and Udh embed

TABLE III
IMPERCEPTIBILITY COMPARISON. "DIPW (P)": EMBEDDED PATCH

	PSNR(dB)↑	SSIM↑	LPIPS↓	APD↓	MOS↓
HiDDeN	41.4	0.991	0.00183	3.255	0.42
Udh	37.6	0.974	0.00011	2.807	0.52
Jia *et al.* [39]	32.4	0.937	0.00101	4.692	0.52
DIPW (P)	37.9	0.971	0.00098	2.991	0.67
DIPW	**49.7**	**0.998**	**0.00001**	**0.175**	**0.2**

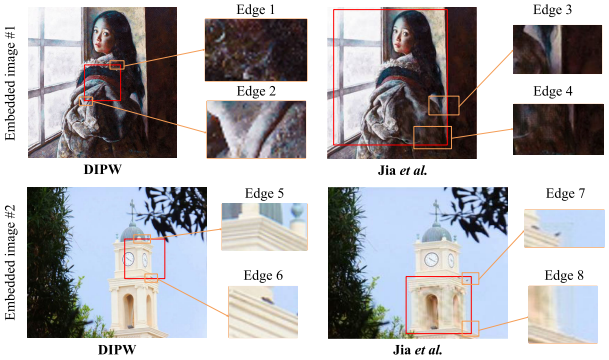

Fig. 12. Samples of embedded images and their embedded patch edges are zoomed for better visualization, where the embedding region of [39] is randomly selected by its localization network.

the watermarks on the downsized full cover image. Instead, DIPW selects images' highlighted patches for embedding, maintaining the overall visual representation of the cover image by constraining patches' size; 2) although [39] is also the region-embedded method, it resizes the cropped regions to 256×256 when embedding, whereas DIPW directly crops out patches with 128×128 size in the original image. Not only that, we find that DIPW maintains well imperceptibility even when compared at patch-level, which is thanks to the encoder's high reconstruction performance obtained through Stage I training.

Moreover, as the human eye is sensitive to edge contours [58], the impact of our patch-based embedding on image views can be visually studied by digging into the edge information loss and local degradation. On one hand, in Fig. 12, considering that visual differences of the cover-embedded pair are mainly reflected in the embedded area edge [59], we zoom the cross areas near the embedded area edge of our embedded image and [39]'s embedded image, to highlight the visual differences between the embedded region and the cover image. It reveals that the difference degree of our method is unrelated to the complexity of backgrounds, embedding on the feature-intensive area (Edges #1, 2) and otherwise (Edges #5, 6) are both blend perfectly, i.e., predominant edge information is retained without clear margin. In contrast, due to the compression of the sub-images, [39] has some slight blurring on cross areas near the embedded area edge after watermark embedding (Edges #3, 4, 7, 8). On the other hand, we generate the edge maps of some embedded images and the corresponding cover images, where the cover images' edge maps are regarded as the ground truth (GT). As shown in Fig. 13, the embedded image inevitably loses information, with the edge map of DIPW being the closest to GT, due to its visual differences of the cover-embedded pair are mainly reflected in the embedded patch [59], which is almost invisible. Whereas for HiDDeN, Udh, and [39], due to the compression, more blurring is introduced and most of the image information is lost. In summary, our embedded patch only slightly reduces the visual aesthetics of the image, possessing a higher imperceptibility.

TABLE IV
CAPACITY COMPARISON. "N/A": NOT APPLICABLE

Bpp ↑	Patch Size					
($\times 10^{-3}$)	Identity	2×2	4×4	8×8	16×16	32×32
HiDDeN	1.8	N/A	N/A	N/A	N/A	N/A
Udh	N/A	250	62.5	15.6	3.9	0.9
Jia *et al.* [39]	2.99	N/A	N/A	N/A	N/A	N/A
DIPW	N/A	$250\times k$	$62.5\times k$	$15.6\times k$	$3.9\times k$	$0.9\times k$

E. Capacity

High capacity means that more of the original author's copyright is embedded in the image, allowing better proof of copyright ownership. Bpp, the message bits hidden per pixel of the encoded image, is the most common metric used to evaluate capacity, which is calculated based on the embedded bits. In the original HiDDeN and [39], the embedded watermark is a 30-bit message and 196-bit message, respectively. In DIPW and Udh, the embedded watermarks are images with the content of $128 \times 128 \times 3$ bytes. To visualize the capacity with bpp, before calculating the capacity, we should transform the embedded byte information into bit information: dividing the watermark image into patches, and setting the patch pixel intensity lower than 128 as bit 0 and that higher than 128 as bit 1, as in Udh. The capacity comparison of the original HiDDeN, Udh, [39], and DIPW is shown in Tab. IV, where "Identity" indicates that no modification is made to the watermark information, while Udh and DIPW are unable to calculate bpp in this case (expressed as N/A); when we calculate bpp under different patch sizes, the original HiDDeN and [39] are N/A; "k" indicates the number of patches embedded with watermarks in DIPW, interestingly, we also find that Udh can be seen as a special case of DIPW in single patch embedding. With these results, it can be concluded that DIPW's capacity is remarkable with better utilization in the spatial dimension.

F. Efficiency

Efficient watermarking technology is consequential in this era with vast amounts of images/videos as people don't want to spend a lot of time processing data [60]. In DIPW, on one hand, the framework is efficient, i.e., requiring only one simple summation to watermark an image; on the other hand, we need to extract patches repeatedly, introducing time overhead. To objectively evaluate the efficiency of DIPW, we randomly select 500 images for watermark embedding and extraction through HiDDeN, Udh and DIPW respectively, using seconds per image (SPI) to calculate the number of seconds it takes to process each image.

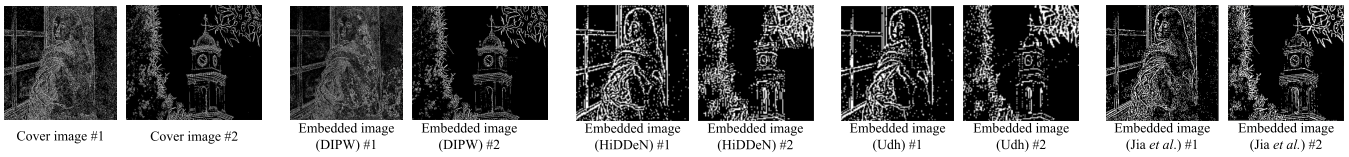

Fig. 13. Edge detection comparison on the examples in Fig. 12.

TABLE V
EFFICIENCY COMPARISON. "DIPW (S)": SINGLE PATCH EMBEDDING, "DIPW (M)": MULTI-PATCH EMBEDDING

SPI ↓	HiDDeN	Udh	Jia *et al.* [39]	DIPW (S)	DIPW (M)
Embedding	1.26	0.71	2.5	0.82	1.49
Extraction	0.96	0.64	2.29	0.76	1.34

The experimental results are shown in Tab. V. We find that the watermark embedding takes more time than extraction, which is because the embedding process needs to process the watermark and the cover image, whereas only one embedded image is required in the extraction process. Among the four methods, Udh is the most efficient as Udh requires only summation of the watermark and cover image; HiDDeN is moderately efficient as HiDDeN requires watermark-cover co-encoding; [39] and DIPW are less efficient as they have to pre-process the cover image to get the sub-image for embedding. In addition, DIPW's efficiency decreases as the number of patches increases. Nevertheless, the SPI of DIPW (S) is acceptable, even approximating real-time, proving the efficiency of DIPW in practice.

G. Ablation Study

To further demonstrate the effectiveness of PatchFinder, Plagiarism-Resistant (PR) training, and Quality Enhancement Module (QEM), we randomly select 2700 images for the ablation study. These images are evenly divided to cover each form of noise for watermark distortion rate testing.

1) Effect of PatchFinder: As shown in Tab. VI, PatchFinder plays an important role in improving the performance of DIPW: the detection rate with PatchFinder decreases by 76.1% as the frequency variation of the cover image caused by watermark embedding is weakened by patch-embedding, making the watermark more undetectable; the highlighted patch-embedding helps the watermark to be maintained under noise and maintains the overall visual representation of the full image, the PSNR value increases by 0.2dB and 12.9dB for watermark and image distortion rates, respectively.

In addition, more analysis of PatchFinder component can increase the enlightenment of our work. We hypothesize that 1) embedding watermarks in multi-patches can help at least one watermark handle distortion since the highlighted patches are not invariant all the time; 2) the patch with larger sizes is more stable. Thus, we test the performance with different numbers k and sizes$m \times m$ of selected patches. From Fig. 14, we find that the robustness improves somewhat with increasing k and m, but this comes at the expense of imperceptibility, fortunately, the embedded image remains well-viewable.

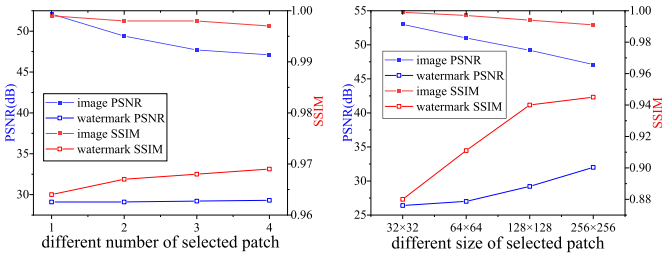

Fig. 14. Ablation study on selected patches, indicates that the robustness improves somewhat with increasing the selected number and patch size, while imperceptibility decreases.

In practical applications, the robustness benefits and quality degradation should be balanced. Besides, essential elements vary from artwork to artwork, i.e., some artworks may have only one real highlighted patch while others may have more, and it is possible that some local patches are essential but no watermark is embedded into them. Therefore, we need to set a reasonable threshold to select patches that make the best performance of watermarking. From the experimental results in this paper, we set $k = 2$ and $m \times m = 128 \times 128$ to achieve this goal.

2) Effect of Plagiarism-Resistant Training: As shown in Tab. VI, the PSNR value with plagiarism-resistant training increases by 4.2dB for watermark distortion rates, demonstrating that such training increase DIPW's noisy factors tolerance. This is thanks to: well-designed noise layers that forces the model to learn encodings that can survive distortions; 2 stage training strategy that helps us fully exploit the potential of the decoder under multiple noise.

3) Effect of Quality Enhancement Module: The QEM module employs the DnCNN-like network and STN to perform pre-processing/optimisating over the distorted embedded image to eliminate the effect of plagiarism actions, and the results in Tab. VI favor its involvement.

VI. DISCUSSION

A. Why Our Classifier (Detector) Works?

As emphasized, the key to the success of deep watermarking is the frequency discrepancy caused by watermark embedding [18], which is also the breakthrough point for our classifier. Since most existing deep watermarking methods are based on the basic backbone framework, we choose the widely applicable and open-source framework (Ddh and Udh) to generate the dataset for classifier training, which can guarantee the classifier's success to some extent, even use it to detect a new method. Note that there is an easily overlooked problem, the classifier may be used for watermark adversarial

TABLE VI
EFFECTIVENESS OF PATCHFINDER, PR TRAINING, AND QEM. THE FOURTH ROW REPRESENTS OUR DIPW

PatchFinder	PR Training	QEM	Image Distortion Rate				Watermark Distortion Rate				Detection
			PSNR(dB)↑	SSIM↑	LPIPS↓	APD↓	PSNR(dB)↑	SSIM↑	LPIPS↓	APD↓	Rate(%) ↓
✗	✓	✓	36.9	0.974	0.00021	2.664	28.1	0.971	0.10033	12.711	82.9%
✓	✗	✓	49.8	0.999	0.00001	0.211	24.1	0.942	0.09194	15.596	6.9%
✓	✓	✗	49.8	0.999	0.00001	0.188	27.7	0.953	0.06772	9.270	6.9%
✓	✓	✓	**49.8**	**0.999**	**0.00001**	**0.176**	**28.3**	**0.974**	**0.05012**	**8.656**	**6.8%**

training to make the generated watermark hard to be detected. Don't worry, adversarial learning can not destroy the fact that "*deep watermarking model tends to embed information on the high-frequency areas, resulting in frequency differences*" [16]. A typical example is HiDDeN has been adversarially for steganalysis, while it is still recognized by our classifier due to frequency discrepancy. For such a problem, we can retrain a similar classifier to perform the detection task. In summary, we provide a straightforward idea for watermark detection and demonstrate its effectiveness beyond just a classifier.

B. Why Patch-Level Embedding Is Effective?

Patch-level embedding mitigates the threat of double agent vulnerability mainly through improving robustness and security. For robustness, as we embed the watermark in the patch with the largest retained possibility during plagiarism, in this way, the patch-level watermark information is expected to be inherited from the original image to the 'created' image, providing robustness against the processing actions. Meanwhile, it is difficult to find the same embedded patches in the case of plagiarism, all detectable patches will be considered as extraction regions to improve accuracy. For security, patch-level embedding greatly reduces the frequency discrepancy brought by the watermark embedding, as the embedding area is smaller than the full image and the location is not fixed in each image, thus making the watermark difficult to be detected. Of course, the adversarial relationship between the classifier and watermarking model still exists, with a small probability of being detected by the classifier (see the 2^{nd}-4^{th} row of Tab. VI). After all, if the frequency discrepancy is completely eliminated, it may lead to the failure of watermark extraction.

C. Limitations

In DIPW, PatchFinder is shown to be stable on the identified embedding regions under different distortions. When facing severe distortions, the re-detected region may be very different (i.e., unsatisfactory detection), then unfortunately the watermark cannot be recovered. However, severe distortion also means the artwork is significantly modified, plagiarizers are usually not motivated to do so, otherwise, does it still commit plagiarism? We consider it an open question.

D. Implication

Open algorithms would inevitably be misused, making them double-edged swords. The revealed double agent vulnerability is an illustrative example in the context of watermarking. Fortunately, a proper remedy could be found by looking into the nature of the plagiarism, for the case of this work, it is the plagiarism characteristics/paradox that in turn keeps 'intact' copyright evidence.

VII. CONCLUSION

This work investigates the limitations of existing watermarking techniques in view of copyright protection against plagiarism. We have then proposed a dedicated design for such contexts using an adaptive patch-level watermarking framework. The novel framework integrates dual-loss function in optimization and watermark-sensitive classifier for gaining imperceptibility and secrecy, introduce an object-attention heuristic in patch elaboration and specific plagiarism-resistant learning for attaining watermark extraction robustness. Experiments and visualization analysis demonstrate the effectiveness and performance of our proposal.

REFERENCES

[1] M. J. Wilber, C. Fang, H. Jin, A. Hertzmann, J. Collomosse, and S. Belongie, "BAM! The Behance artistic media dataset for recognition beyond photography," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1211–1220.

[2] C. Joyce, T. T. Ochoa, M. W. Carroll, M. A. Leaffer, and P. Jaszi, *Copyright Law*. Durham, NC, USA: Carolina Academic Press, 2016.

[3] A. A. Salah et al., "DeviantArt in spotlight: A network of artists," *Leonardo*, vol. 45, no. 5, pp. 486–487, Oct. 2012.

[4] K. Kousha and M. Thelwall, "TinEye searches for image impact assessment," in *Proc. ICSTI*. Leiden, The Netherlands: Leiden Univ., 2010, p. 153.

[5] Y. Lang, Y. He, F. Yang, J. Dong, and H. Xue, "Which is plagiarism: Fashion image retrieval based on regional representation for design protection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2592–2601.

[6] Z. Ma et al., "Fine-grained fashion similarity learning by attribute-specific embedding network," in *Proc. AAAI*. New York, NY, USA: Hilton New York Midtown, 2020, pp. 11741–11748.

[7] R. Liu and T. Tan, "An SVD-based watermarking scheme for protecting rightful ownership," *IEEE Trans. Multimedia*, vol. 4, no. 1, pp. 121–128, Mar. 2002.

[8] X. Luo, R. Zhan, H. Chang, F. Yang, and P. Milanfar, "Distortion agnostic deep watermarking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 13545–13554.

[9] I. Cox, M. Miller, J. Bloom, J. Fridrich, and T. Kalker, *Digital Watermarking and Steganography*. San Mateo, CA, USA: Morgan Kaufmann, 2007.

[10] H. Berghel and L. O'Gorman, "Protecting ownership rights through digital watermarking," *Computer*, vol. 29, no. 7, pp. 101–103, Jul. 1996.

[11] S.-M. Mun, S.-H. Nam, H. Jang, D. Kim, and H.-K. Lee, "Finding robust domain from attacks: A learning framework for blind watermarking," *Neurocomputing*, vol. 337, pp. 191–202, Apr. 2019.

[12] J. Zhu, R. Kaplan, J. Johnson, and L. Fei-Fei, "Hidden: Hiding data with deep networks," in *Proc. ECCV*, Munich, Germany, 2018, pp. 657–672.

[13] M. Ahmadi, A. Norouzi, N. Karimi, S. Samavi, and A. Emami, "ReDMark: Framework for residual diffusion watermarking based on deep networks," *Expert Syst. Appl.*, vol. 146, May 2020, Art. no. 113157.

[14] Y. Liu, M. Guo, J. Zhang, Y. Zhu, and X. Xie, "A novel two-stage separable deep learning framework for practical blind watermarking," in *Proc. 27th ACM Int. Conf. Multimedia*, Oct. 2019, pp. 1509–1517.

[15] C. Yu, "Attention based data hiding with generative adversarial networks," in *Proc. AAAI*. New York, NY, USA: Hilton New York Midtown, 2020, pp. 1120–1128.

[16] H. Zhang, H. Wang, Y. Cao, C. Shen, and Y. Li, "Robust watermarking using inverse gradient attention," in *Proc. CVPR*, Seattle, WA, USA, Nov. 2020, pp. 2592–2601.

[17] 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.

[18] C. Zhang, P. Benz, A. Karjauv, G. Sun, and I. S. Kweon, "UDH: Universal deep hiding for steganography, watermarking, and light field messaging," in *Proc. NeurIPS*, 2020, pp. 10223–10234.

[19] 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.

[20] C. Zhang, C. Lin, P. Benz, K. Chen, W. Zhang, and I. So Kweon, "A brief survey on deep learning based data hiding," 2021, *arXiv:2103.01607*.

[21] M. Tanha, S. D. S. Torshizi, M. T. Abdullah, and F. Hashim, "An overview of attacks against digital watermarking and their respective countermeasures," in *Proc. Title, Int. Conf. Cyber Secur., Cyber Warfare Digit. Forensic (CyberSec)*, Jun. 2012, pp. 265–270.

[22] A. Cohen, J. Holmgren, R. Nishimaki, V. Vaikuntanathan, and D. Wichs, "Watermarking cryptographic capabilities," in *Proc. 48th Annu. ACM Symp. Theory Comput.*, Jun. 2016, pp. 1115–1127.

[23] S. Cui, F. Liu, T. Zhou, and M. Zhang, "Understanding and identifying artwork plagiarism with the wisdom of designers: A case study on poster artworks," in *Proc. 30th ACM Int. Conf. Multimedia*, Oct. 2022, pp. 1117–1127.

[24] H. Tao, L. Chongmin, J. M. Zain, and A. N. Abdalla, "Robust image watermarking theories and techniques: A review," *J. Appl. Res. Technol.*, vol. 12, no. 1, pp. 122–138, Feb. 2014.

[25] M. Barni, F. Bartolini, V. Cappellini, and A. Piva, "A DCT-domain system for robust image watermarking," *Signal Process.*, vol. 66, no. 3, pp. 357–372, May 1998.

[26] M. Ali, C. W. Ahn, and M. Pant, "A robust image watermarking technique using SVD and differential evolution in DCT domain," *Optik*, vol. 125, no. 1, pp. 428–434, Jan. 2014.

[27] P. Bao and X. Ma, "Image adaptive watermarking using wavelet domain singular value decomposition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 1, pp. 96–102, Jan. 2005.

[28] X. Kang, J. Huang, Y. Q. Shi, and Y. Lin, "A DWT-DFT composite watermarking scheme robust to both affine transform and JPEG compression," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 8, pp. 776–786, Aug. 2003.

[29] N. Nikolaidis and I. Pitas, "Robust image watermarking in the spatial domain," *Signal Process.*, vol. 66, no. 3, pp. 385–403, May 1998.

[30] H. Fang et al., "Deep template-based watermarking," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 4, pp. 1436–1451, Apr. 2021.

[31] B. Wang, Y. Wu, and G. Wang, "Adaptor: Improving the robustness and imperceptibility of watermarking by the adaptive strength factor," *IEEE Trans. Circuits Syst. Video Technol.*, early access, Apr. 10, 2023, doi: 10.1109/TCSVT.2023.3265970.

[32] Y. Huang, H. Guan, J. Liu, S. Zhang, B. Niu, and G. Zhang, "Robust texture-aware local adaptive image watermarking with perceptual guarantee," *IEEE Trans. Circuits Syst. Video Technol.*, early access, Feb. 15, 2023, doi: 10.1109/TCSVT.2023.3245650.

[33] H. Kim, "Robust image watermarking using local invariant features," *Opt. Eng.*, vol. 45, no. 3, Mar. 2006, Art. no. 037002.

[34] E. Nezhadarya, Z. J. Wang, and R. K. Ward, "Robust image watermarking based on multiscale gradient direction quantization," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 4, pp. 1200–1213, Dec. 2011.

[35] C. Wang et al., "RD-IWAN: Residual dense based imperceptible watermark attack network," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 11, pp. 7460–7472, Nov. 2022.

[36] Z. Jia, H. Fang, and W. Zhang, "MBRS: Enhancing robustness of DNN-based watermarking by mini-batch of real and simulated JPEG compression," in *Proc. 29th ACM Int. Conf. Multimedia*, Oct. 2021, pp. 41–49.

[37] Y. Luo, T. Zhou, F. Liu, and Z. Cai, "IRWArt: Levering watermarking performance for protecting high-quality artwork images," in *Proc. ACM Web Conf.*, Apr. 2023, pp. 2340–2348.

[38] S. Baluja, "Hiding images within images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 7, pp. 1685–1697, Jul. 2020.

[39] J. Jia, Z. Gao, D. Zhu, X. Min, G. Zhai, and X. Yang, "Learning invisible markers for hidden codes in Offline-to-online photography," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 2263–2272.

[40] J. P. M. G. Linnartz and M. Van Dijk, "Analysis of the sensitivity attack against electronic watermarks in images," in *Proc. IH*. Portland, OR, USA, Apr. 1998, pp. 258–272.

[41] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[42] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, Jul. 2017.

[43] M. Jaderberg, K. Simonyan, and A. Zisserman, "Spatial transformer networks," in *Proc. NeurIPS*, 2015.

[44] K. Baek and H. Shim, "Commonality in natural images rescues GANs: Pretraining GANs with generic and privacy-free synthetic data," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 7844–7854.

[45] R. Yu and J. Gero, "The effect of digital design representation on designers' visual attention," in *Proc. DRS, Synergy*, Sep. 2020, pp. 657–672.

[46] M. Tichindelean, M. T. Tichindelean, I. Cetină, and G. Orzan, "A comparative eye tracking study of usability—Towards sustainable web design," *Sustainability*, vol. 13, no. 18, Sep. 2021, Art. no. 10415.

[47] Y. Deng et al., "Exploring the representativity of art paintings," *IEEE Trans. Multimedia*, vol. 23, pp. 2794–2805, 2021.

[48] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "YOLOX: Exceeding YOLO series in 2021," 2021, *arXiv:2107.08430*.

[49] L.-C. Chiu, T.-S. Chang, J.-Y. Chen, and N. Y. Chang, "Fast SIFT design for real-time visual feature extraction," *IEEE Trans. Image Process.*, vol. 22, no. 8, pp. 3158–3167, Aug. 2013.

[50] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 618–626.

[51] C. Song, S. Sudirman, M. Merabti, and D. Llewellyn-Jones, "Analysis of digital image watermark attacks," in *Proc. 7th IEEE Consum. Commun. Netw. Conf.*, Jan. 2010, pp. 1–5.

[52] S. Mohammad and S. Kiritchenko, "Wikiart emotions: An annotated dataset of emotions evoked by art," in *Proc. LREC*, Miyazaki, Japan, 2018, pp. 1–14.

[53] Q. Huynh-Thu and M. Ghanbari, "Scope of validity of PSNR in image/video quality assessment," *Electron. Lett.*, vol. 44, no. 13, pp. 800–801, Jun. 2008.

[54] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[55] 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.

[56] S. A. Broughton and K. Bryan, *Discrete Fourier Analysis and Wavelets: Applications to Signal and Image Processing*. Hoboken, NJ, USA: Wiley, 2018.

[57] R. C. Streijl, S. Winkler, and D. S. Hands, "Mean opinion score (MOS) revisited: Methods and applications, limitations and alternatives," *Multimedia Syst.*, vol. 22, no. 2, pp. 213–227, Mar. 2016.

[58] M. G. Martini, C. T. E. R. Hewage, and B. Villarini, "Image quality assessment based on edge preservation," *Signal Process., Image Commun.*, vol. 27, no. 8, pp. 875–882, Sep. 2012.

[59] R. Maini and H. Aggarwal, "Study and comparison of various image edge detection techniques," *Int. J. Image Process.*, vol. 3, no. 1, pp. 1–11, Feb. 2009.

[60] V. Sharma and R. N. Mir, "An enhanced time efficient technique for image watermarking using ant colony optimization and light gradient boosting algorithm," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 34, no. 3, pp. 615–626, Mar. 2022.

Yuanjing Luo received the bachelor's degree from Hainan Normal University and the master's degree from the Central South University of Forestry and Technology. She is currently pursuing the Ph.D. degree with the National University of Defense Technology, China. Her research interests include digital watermarking, steganography, deep learning, and information security.

Tongqing Zhou received the B.S., M.S., and Ph.D. degrees in computer science and technology from the National University of Defense Technology (NUDT), Changsha, in 2012, 2014, and 2018, respectively. He is currently an Assistant Researcher with the College of Computer, NUDT. His main research interests include network measurement, crowdsensing, and data privacy. He was a recipient of the Outstanding Ph.D. Dissertation Award of Hunan Province, China, and the Outstanding Post-Doctoral Award of Hunan Province, China.

Shenglan Cui received the bachelor's degree from Northeastern University, China. She is currently pursuing the Ph.D. degree with Hunan University, China. Her research interests include artwork copyright protection, AI aid interaction design, and human–computer interaction.

Yunfan Ye received the bachelor's degree from Xiamen University and the master's degree from the Stevens Institute of Technology. He is currently pursuing the Ph.D. degree with the National University of Defense Technology, China. His research interests include edge detection, computer vision, graphics, and their applications.

Fang Liu received the B.S. and Ph.D. degrees in computer science from the National University of Defense Technology, Changsha, China, in 1999 and 2005, respectively. She is currently a Full Professor with the School of Design, Hunan University. Her main research interests include edge computing, data storage and management, and intelligent design.

Zhiping Cai (Member, IEEE) received the B.Eng., M.A.Sc., and Ph.D. degrees in computer science and technology from the National University of Defense Technology (NUDT), China, in 1996, 2002, and 2005, respectively. He is currently a Full Professor with the College of Computer, NUDT. His current research interests include artificial intelligence, network security, and big data. He is a Senior Member of the CCF.