IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 20 June 2023, accepted 7 July 2023, date of publication 14 July 2023, date of current version 20 July 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3295699

RESEARCH ARTICLE

Toward Practical Deep Blind Watermarking for Traitor Tracing

BOSUNG YANG, GYEONGSUP LIM, AND JUNBEOM HUR
Department of Computer Science and Engineering, Korea University, Seoul 02841, South Korea

Corresponding author: Junbeom Hur (jbhur@korea.ac.kr)

This work was supported in part by the Institute of Information and Communications Technology Planning and Evaluation (IITP) under Grant 2022-0-00411, Grant IITP-2023-2021-0-01810, and Grant IITP-2023-2020-0-01819; and in part by the National Research Foundation Grant funded by the Korea Government under Grant NRF-2021R1A6A1A13044830.

ABSTRACT Traitor tracing via blind watermarking is a promising solution to protect copyright. Recently, it was demonstrated that deep learning-based blind watermarking methods could outperform traditional watermarking methods in terms of robustness against distortions. However, we found they could sacrifice the imperceptibility as a trade-off. To handle this challenge, we propose a novel method consisting of a deep blind model and watermarking strategy. For the purpose, we first investigate the fundamental components of the basic deep blind watermarking model, and empirically show how the performance changes when each component is modified with respect to the robustness. Based on it, we construct a deep blind watermarking encoder, CFC+CONCAT, which can encode watermarks in a robust way against distortions without imperceptibility degradation. We then propose a watermarking strategy to make deep blind watermarking robust by increasing watermarking capacity (by splitting a large image into small patches), and using the effect of distortion types on the robustness we found. According to the experiments, our method achieved 5.46 higher PSNR on average than the baseline methods with comparable robustness under various distortions when watermarking in the $3 \times 256 \times 256$ image. Also, the training time and VRAM usage are reduced by less than 1/4, demonstrating our method can mitigate the trade-off between robustness and imperceptibility, and achieve lightweight training.

INDEX TERMS Traitor tracing, digital watermarking.

I. INTRODUCTION

As the digital media market gradually expands, copyright infringement of online content such as cartoons, photos, and illustrations becomes widely observable on many piracy sites [1], [2]. Copyright protection is regarded as a social challenge because of the ease of duplicating digital images and the resulting economic loss [3], [4].

Traitor tracing via blind watermarking is a promising solution to prevent such piracy and illegal dissemination [5], [6], [7]. Blind watermarking is a data hiding technique that allows the embedding of data in digital media, such as images [8], [9], [10], videos [11], [12], [13], and text [14], [15], [16], in an imperceptible manner. This is done by editing the least significant parts of the digital media, which are not visible to the human eye. Fig. 1 illustrates the traitor tracing scenario using blind watermarking. The service provider (sender) encodes the legitimate recipient's information as a fingerprint in the image. When it is necessary to track the illegal distributor for the target images which might be found at the piracy sites, the service provider decodes the suspects' information from it. By leveraging computer vision techniques such as adversarial perturbations [17], [18], [19] and data augmentation, blind watermarking based on deep learning, or deep blind watermarking, outperforms traditional blind watermarking methods in terms of robustness (bit accuracy) against distortions [20]. However, applying deep blind watermarking to the real-world traitor tracing scenario is challenging because of the trade-off between imperceptibility and robustness.

Specifically, increasing robustness of deep blind watermarking models may sacrifice imperceptibility as a trade-off. Most piracy samples are likely to be distorted by repetitive

The associate editor coordinating the review of this manuscript and approving it for publication was Ahmed Farouk.

replications, which hinders the correct decoding of watermarks. For example, the victim image may be cropped by the capturing tool, resized, or lossily compressed in the process of illegal copying and saving procedures. Additionally, the traitor may intentionally deform the image for financial gains, such as advertisement insertion [21]. In addition, multiple distortions are often applied concurrently in an orthogonal way. Even though the robust deep blind watermarking could reconstruct the hidden watermark correctly, it is likely to result in lower imperceptibility, degrading the user experience.

In this paper, we propose a novel method to mitigate this trade-off, specifically a model architecture and a watermarking strategy. First, we investigate the effect of the watermarking model's architectural difference on the robustness and imperceptibility. In particular, we examined the application target of the convolutional layer by modifying the location of the feature concatenation layer that merges the feature maps of the cover image and watermark. As a result, we found that applying convolution to a mixture of a watermark message and an image results in higher robustness but lower imperceptibility than independently applying it to each message and image, demonstrating the early-located feature concatenation layer has the benefit of robustness (*finding 1*). We also observed that the skip connection of the cover image can enhance imperceptibility but reduce the decoding robustness. Specifically, the generating method through residual connection, which adds the cover image to the residual one, shows the highest imperceptibility but the lowest robustness; and another method with feature concatenation, which concatenates the cover image into an intermediate representation, has balanced imperceptibility and robustness in our experiments (*finding 2*). As a compromise between the above two findings (*finding 1 and 2*), we propose a robust deep blind watermarking encoder, CFC+CONCAT. It applies convolution to the mixture and concatenates the cover image into an intermediate representation. As a result, CFC+CONCAT improves the robustness without imperceptibility degradation.

Second, in order to efficiently leverage the watermarking model, we investigate the changes in the watermarking capacity by varying image size, channel size, and the effect of distortion types. To this aim, we first investigate how the robustness and imperceptibility are affected by the channel size and image size. The investigation results show that the decoding robustness generally increases as the channel size and image size increase, while the imperceptibility shows no apparent changes. Thus, increasing channel or image size can be helpful to en/de-code more watermarks without degrading imperceptibility, potentially enabling the use of error correction techniques [22], [23] to improve robustness. However, we also found that the watermark capacity has an upper limit. Specifically, the robustness gain becomes smaller as more bits are encoded, although the channel or image size increases. Moreover, we examine the impact of various distortions on our deep blind watermarking model, and reveal that only partial bits are differently affected by each distortion type, rather than all bits being affected equally. For example, we observed that in the case of image cropping, the decoder can mispredict specific bits, while accurately predicting the other bits. By leveraging the observations, we propose a watermarking strategy for achieving high robustness without degradation of imperceptibility. It splits a large image into small patches, and encodes shifted watermark in each patch. Since the difference in watermark capacity of the small patch and the whole image is not so significant, we can make watermark capacity gain proportional to the number of patches. By leveraging the increased capacity to encode the circularly shifted watermark in each patch, we can make deep blind watermarking more robust. Furthermore, the training time and computation resources are also reduced, because training for a small patch is cheaper than for a large image.

The main contributions of this paper are as follows:

- We conduct a simulation that reflects distortions on real piracy sites for traitor tracing. In our simulation, we measure how the robustness is affected by the intensity of distortions, and show which distortion significantly degrades robustness when combined with the others.
- For robust deep blind watermarking against distortions without imperceptibility degradation, we investigate the encoder architecture, and find (1) the early concatenation of the cover image and watermark increases robustness, and (2) concatenating the cover image with intermediate representation enhances the imperceptibility. Based on our findings, we propose a robust deep blind watermarking encoder, CFC+CONCAT, and demonstrate it achieves increased robustness with high imperceptibility.
- By leveraging our other findings that (1) the watermark capacity of a small image is almost the same as that of a large image, and (2) only partial bits are differently affected by each distortion type, we propose a watermarking strategy that splits a large image into small patches and encodes shifted watermark into each patch. The proposed strategy could correct mispredicted bits of the watermark, which increases robustness with reduced training time.
- By leveraging the proposed model and strategy, we achieve similar robustness under various distortions and improve PSNR by 5.46 higher on average compared to the baseline methods [20], [24], [25].

II. RELATED WORK

Blind watermarking is an application of steganography for data hiding. Deep learning-based data hiding [20], [24], [25], [26], [27], [28], [29], [30], [31], [32], [33], [34], [35], [36], [37] generally contains two main parts: the encoder and the decoder. The encoder generates encoded image, I_{enc}, by hiding watermark message, $M_{in} \in \{0, 1\}^n$, in cover image, I_{cover}. The decoder reconstructs the watermark, M_{out}, from I_{enc}. The main goal of the encoder and decoder is minimizing the difference between I_{cover} and I_{enc}, and the difference between M_{in} and M_{out}.

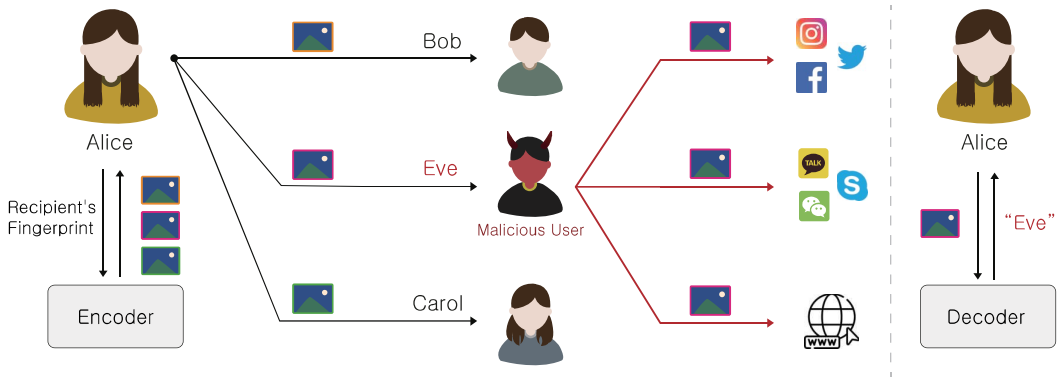

FIGURE 1. Example of traitor tracing scenario. Before Alice transmits the image to others, the watermarks as the fingerprint of each recipient are encoded in the image. Alice transmits the image to the recipients via a legitimate route (black line). Eve, a malicious user, illegally distributes the image through social networks, messenger, or the Internet (red line). When Alice recognizes that her image is illegally distributed, she can reveal that Eve has done it by decoding the fingerprint from the image.

Recent studies have focused on the robustness of decoding watermark under distortions. Zhu et al. [20] proposed HiDDeN, where the attack simulation layer is introduced between the encoder and the decoder. The attack simulation layer artificially distorts the encoded image to have an advantage of the data augmentation effect. By training with distorted images, HiDDeN predicts the invisible watermark robustly under the distorted image condition. Based on HiDDeN, several subsequent studies have focused on the design of the attack simulation layer for robust deep blind watermarking. StegaStamp [25] introduced warpping, blurring, color manipulation, and JPEG compression into the attack simulation layer. Luo et al. [24] designed the attack simulation layer to generate adversarial examples, aiming to induce misprediction of the decoder. As a result, Luo et al.'s method robustly decodes the watermark under various distortions without supervision, i.e., distortion-agnostic manner. Jia et al. [36] focused on robustness against JPEG compression by constructing an attack simulation layer with mini-batch real and simulated JPEG compression. Fang et al. [37] proposed PIMoG that simulates screen-shot distortion as an attack simulation layer.

Despite these advances of deep blind watermarking, adopting the attack simulation layer causes the inevitable degradation of imperceptibility. Therefore, in this paper, we investigate the novel properties of deep blind watermarking, which can contribute to solving the challenge, and propose a watermarking model and strategy based on it.

III. ROBUST DEEP BLIND WATERMARKING ENCODER

In this section, we empirically investigate the application target of the convolutional layer, and the connection method between the cover image and the intermediate feature map. The brief summary of our findings is as follows:

- **Location of the feature concatenation layer (Section III-A2).** Earlier placement of the feature concatenation layer results in higher robustness.

TABLE 1. Robustness of each encoder architecture.

	None	Ad insertion	Crop	Resize	JPEG compression
ICC	0.9911	0.5935	0.5933	0.5392	0.5670
MCC	0.9782	0.6497	0.5691	0.5682	0.6042
BCC	0.9644	0.6861	0.6448	0.6156	0.6355
CFC	0.9923	0.9247	0.7910	0.8454	0.7861

- **Concatenation of the cover image (Section III-A3).** Concatenating the cover image with the intermediate feature map shows higher imperceptibility without robustness degradation.

Next, we propose a robust deep blind watermarking encoder by leveraging the above findings.

A. INVESTIGATING ENCODER ARCHITECTURE

1) EXPERIMENTAL SETUP AND EVALUATION METRICS

In order to design the encoder architecture, we implemented the target of convolution and the skip connection method of the I_{cover} based on HiDDeN [20] using PyTorch [38] on NVIDIA 3090. We use 10,000 images of the MS COCO dataset [39] for training the models, and 1,000 images for testing. For gradient descent, Adam [40] is applied with a learning rate of 0.001. The loss functions for deep blind watermarking are the same as HiDDeN.

We utilize the Bit Error Rate (BER), ratio of incorrectly decoded bits to encoded ones, to evaluate the watermark decoding robustness of a deep blind watermarking model, which is defined as a follow.

$$BER = \frac{1}{len}\sum_{n=1}^{len} e_n \oplus o_n, \tag{1}$$

$$Robustness = 1 - BER, \tag{2}$$

where len denotes the length of the watermark message, e_n and o_n denote n-th bit of M_{out} and M_{in}, respectively. The imperceptibility is measured using PSNR [41] and SSIM [42] that calculate the similarity between the I_{cover} and the I_{enc}.

TABLE 2. Imperceptibility of each encoder architecture.

	ICC	MCC	BCC	CFC
PSNR	**37.51**	31.79	30.84	29.64
SSIM	**0.9893**	0.9621	0.9231	0.9353

TABLE 3. Comparison of ICC, ICC+CONCAT (HiDDeN), and ICC+RESI.

	Robustness	Imperceptibility (PSNR)	Imperceptibility (SSIM)
ICC	**0.9911**	37.51	0.9893
ICC+CONCAT	0.9743	39.46	0.9873
ICC+RESI	0.8801	**43.29**	**0.9929**

Specifically, PSNR is the metric for similarity at the pixel level, and SSIM evaluates the similarity between two images in Luminance, Contrast, and Structure.

2) APPLICATION TARGET OF CONVOLUTION

For analyzing the performance changes affected by the target of convolution, we generate four encoder architectures based on HiDDeN by varying the target of convolutional layers:

- Image-Conv-Concat (ICC): The convolution layers operate on the image features.
- Message-Conv-Concat (MCC): The convolution layers operate on the message features.
- Both-Conv-Concat (BCC): The convolution layers operate on both the image and message features.
- Concat-First-Conv (CFC): The image and message features are first concatenated, and then the convolution layers operate on the concatenated features.

These architectures spatially repeat M_{in} to generate the message volume as a preprocessing, and then apply Conv-BN-ReLU blocks to generate the intermediate representations.

ICC is similar to HiDDeN, but the skip connection of the I_{cover} is removed to compare with the other encoders under the same condition. MCC applies four Conv-BN-ReLU blocks to the message volume, and concatenates it with I_{cover}. BCC respectively applies four Conv-BN-ReLU blocks to the message volume and I_{cover}, and concatenates them. The concatenated intermediate representation of these three encoders applies one Conv-BN-ReLU block and 1×1 convolutional layer to generate I_{enc}. On the other hand, CFC first concatenates the message volume and I_{cover}, and applies five Conv-BN-ReLU blocks. It then applies a 1×1 convolution to generate I_{enc}.

Table 1 shows the evaluation results on the robustness of the four encoders under various distortions, i.e., advertisement insertion, crop, resize, and JPEG compression. For advertisement insertion, we insert a $3 \times 13 \times 13$ image into $3 \times 128 \times 128$ I_{enc}. For crop and resize, the distorted image's size is reduced to 90% of its original size. For JPEG compression, the quality factor is set to 90. Our empirical analysis results show that applying Conv-BN-ReLU blocks to the mixture of message volume and I_{cover} can mitigate the robustness degradation under various distortions. Thus, this architectural change itself can be useful to enhance robustness even without any data augmentation, which was the way of the previous works [20], [26], [29] for the same purpose. However, Table 2 shows that the architectural change inevitably incurs a significant degradation of imperceptibility as a trade-off, which should be appropriately handled.

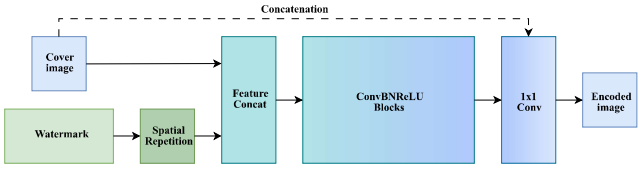

FIGURE 2. Overview of CFC+CONCAT (proposed encoder).

3) SKIP CONNECTION

Inspired by HiDDeN and RMFEN [29], we investigate how to use skip connection of the I_{cover} to improve the imperceptibility. Based on ICC, we explore the following two ways. First, we concatenate I_{cover} with the antepenultimate intermediate representation (this method is called CONCAT). Second, we make the encoder generate a residual image, and add it to I_{cover} (this method is called RESI). Table 3 shows the comparison results with different skip connections. The two ways of leveraging skip connection, that is ICC+CONCAT and ICC+RESI, increase imperceptibility, while decreasing robustness compared to ICC. Specifically, ICC+RESI shows significant changes in robustness and imperceptibility, while ICC+CONCAT shows a more balanced performance.

B. CFC+CONCAT

Based on the investigations of the encoder architectures in Section III-A, we propose a deep blind watermarking encoder, called CFC+CONCAT, aiming to enhance both robustness and imperceptibility. CFC+CONCAT concatenates the antepenultimate intermediate representation with I_{cover} to mitigate the imperceptibility degradation of CFC while maintaining robustness. Fig. 2 shows the overview of CFC+CONCAT. Although generating residual image and adding to the I_{cover} achieves higher imperceptibility than the concatenation, it decreases robustness because of the dilution of message's representation [29]. Thus, we utilize concatenating technique for enhancing imperceptibility of I_{enc}.

IV. STRATEGY FOR DEEP BLIND WATERMARKING

In this section, we investigate how the channel size of convolutional layers and image size affect the capacity of watermarks in deep blind watermarking, and what bits are mispredicted by distortions. The brief summary of our findings is as follows:

- **Upper limit of watermarking capacity (Section IV-A, and IV-B).** Deep blind watermarking has a capacity limitation. Specifically, the watermark capacity of a small image is almost the same as that of a large image.

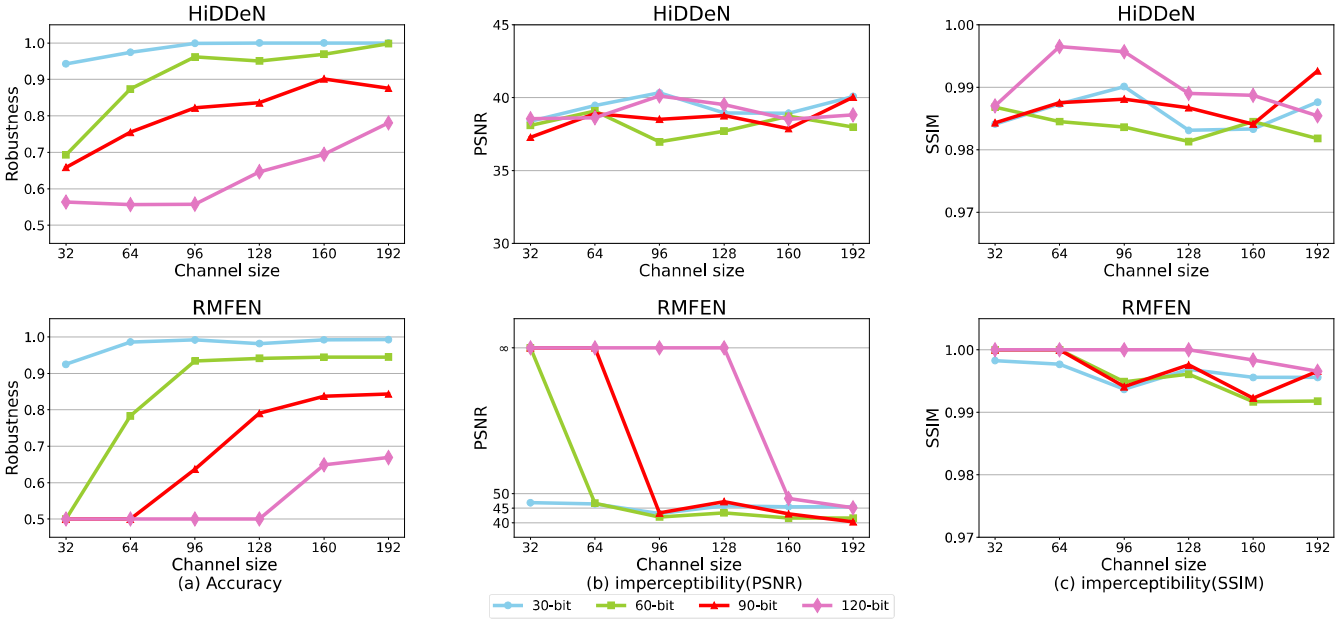

FIGURE 3. Evaluation of robustness and imperceptibility with different channel size.

- **Misprediction tendency according to distortion type (Section IV-C).** Deep blind watermarking tends to differently mispredict certain bits of the hidden watermark according to the type of distortion.

Based on these findings, we propose a watermarking strategy that can make our deep blind watermarking method more robust.

A. CHANNEL SIZE OF CONVOLUTIONAL LAYERS

We first investigate the changes in robustness and imperceptibility with different channel sizes of the convolutional layers in the same experiment setting as that in Section III-A1. As shown in Fig. 3 (a), the robustness of two models, HiDDeN [20] and RMFEN [29], increases as the channel size of each convolutional layer increases from 32 to 192. Interestingly, we observe that the robustness increases conspicuously when the channel size becomes larger than the M_{in}'s length. For example, when en/de-coding 60-bit, decoding robustness with 64-channel considerably increases than with 32-channel. Although the RMFEN with 128-channel did not reconstruct 120-bit M_{in}, its robustness also significantly increases with 160-channel. Since M_{in} is spatially repeated, each channel of the message volume represents each bit of M_{in}. Thus, the larger channel size than the message's length provides redundant effects that improve robustness [24]. On the other hand, the smaller channel size causes the information loss of M_{in}, leading to robustness degradation.

Fig. 3 (b) and (c) show the imperceptibility of the encoded image with different numbers of channels. As shown in the figures, there are no remarkable changes in imperceptibility in both PSNR and SSIM as the channel size changes, once the training of the models is properly converged (this is the case for all except the one where PSNR is evaluated as ∞). When the training is prematurely converged on the local minimum, the robustness equals 0.5 (Fig. 3 (a)) and PSNR is infinity (Fig. 3 (b)) because the training only improves imperceptibility, not robustness.

In this experimental result, increasing channel size seems to enhance the robustness without loss of imperceptibility. Unfortunately, however, we observed that it has a limitation in terms of watermark capacity. Even though the larger channel size incurs higher robustness, the amount of increase in robustness becomes smaller as the channel size increases. In addition, the larger channel size requires more training time. For example, when encoding 60-bit watermark with 128 and 160 channels in HiDDeN, robustness gain is only 0.0185, while training time increases from 15 to 21.5 hours per 300 epochs. Even when the channel size is 256, the robustness for decoding 60-bit is 0.9662. It is similar to the robustness that can be achieved when the channel size is 160, but it takes 1.65 times more training time. Moreover, we found that if M_{in}'s length is larger than 90-bit, the robustness cannot reach 0.9 even if the channel size increases. When en/de-coding 90-bit, there is no robustness difference between training with 160-channel and 256-channel. Therefore, increasing only the channel size cannot be an appropriate solution to enhance watermark capacity.

B. SIZE OF COVER IMAGE

We next investigate the changes in robustness and imperceptibility by varying the size of I_{cover}. As shown in Fig. 4 (a), as the size of I_{cover} increases, the robustness also tends to increase in most of the cases, because of the increased spatial redundancy of M_{in}. In HiDDeN, we observe an robustness drop with $3 \times 224 \times 224$ images because of unstable training. However, it is important to note that en/decoding with

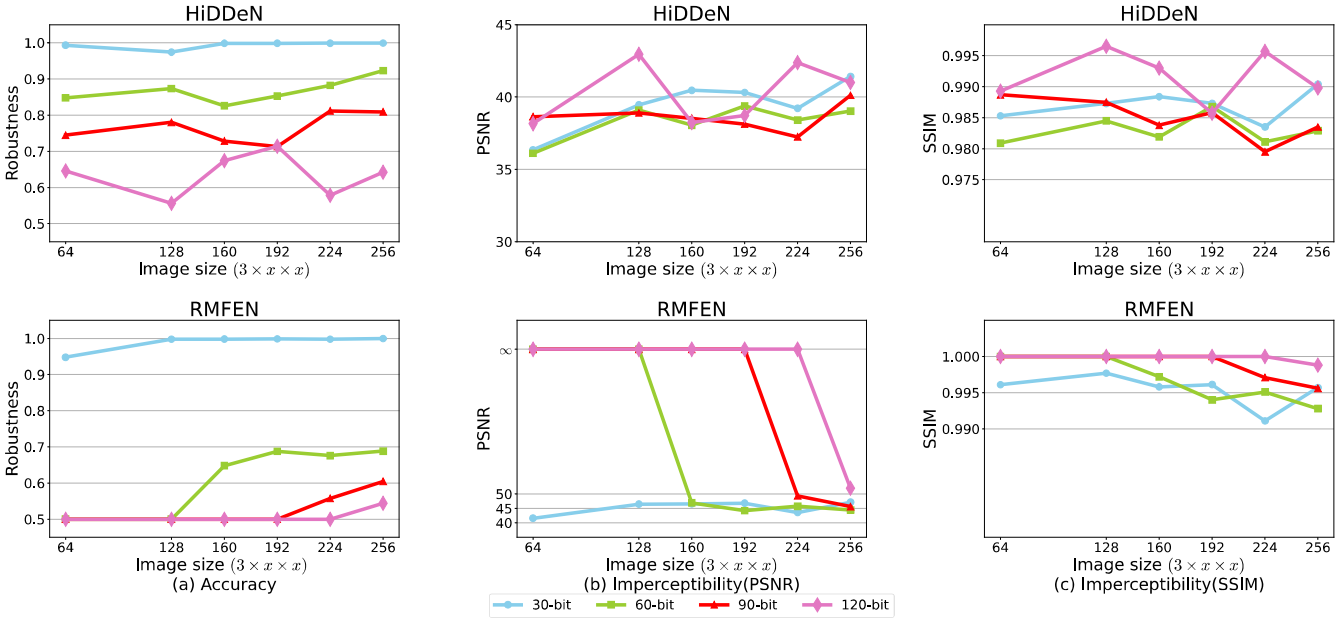

FIGURE 4. Evaluation of robustness and imperceptibility with different size of I_{cover}.

$3 \times 256 \times 256$ images eventually achieves higher robustness than en/de-coding with $3 \times 64 \times 64$ images. Unlike traditional blind watermarking methods in which the watermark length is proportional to the image size, however, the robustness gain from the increased size of I_{cover} is not so significant. For example, in the traditional watermarking method such as LSB [43], the robustness of decoding 30-bit from $3 \times 128 \times 128$ images is the same as decoding 120-bit from $3 \times 256 \times 256$ images. On the other hand, the robustness significantly drops in deep blind watermarking models when trained to en/de-code 120-bit in $3 \times 256 \times 256$ images, compared to training to en/de-code 30-bit in $3 \times 128 \times 128$ images. In our experimental result, the robustness improvement becomes smaller, when I_{cover} is larger than $3 \times 192 \times 192$.

Fig. 4 (b) and (c) show the imperceptibility changes by varying the size of I_{cover}. Since M_{in} is repeated spatially as much as the image size, the imperceptibility is not affected by the size.

When it comes to computational overhead, training watermarking model for larger images requires more computation, leading to the increase in training time. As demonstrated in Section IV-A, robustness gain from the increased I_{cover} size becomes smaller, while the training time becomes comparatively much longer. Thus, deep blind watermarking methods are not scalable with regard to image size, deteriorating their practicality in practice.

According to our analysis in Section IV-A and IV-B, we found that en/de-coding 30-bit watermark message is appropriate for robust watermarking (stably higher robustness than 0.98).

C. PROPOSED WATERMARKING STRATEGY

For practical traitor tracing, directly applying a deep blind watermarking to a large image is not cost-efficient in the aspect of training time and computation, due to the upper limit in the trainable watermark capacity as we demonstrated. Thus, we propose a novel watermarking strategy leveraging image-splitting and bit shifting techniques to solve this problem, while enhancing the robustness. Fig. 5 shows an overview of the proposed watermarking strategy. It first splits a large I_{cover} into multiple small patches, encodes M_{in}, and then merges each encoded patch. When decoding it, our method splits I_{enc} into small encoded patches and reconstructs M_{out}. Because the watermark capacity of a small patch is almost the same as that of a large image, we can en/de-code as many watermarks as the number of patches.

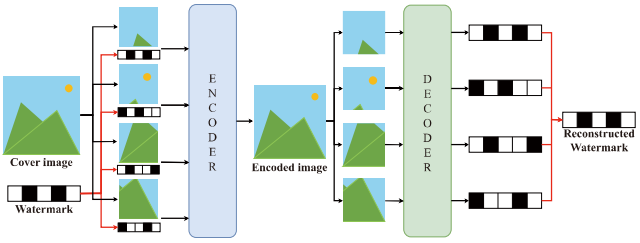

FIGURE 5. Overview of our watermarking strategy.

In order to enhance robustness, we leverage the increased watermark capacity obtained by image-splitting technique as redundant data. Specifically, we can consider channel coding [22], [23] and voting-based methods [44] to correct the mispredicted data by utilizing redundant data. However, channel coding methods cannot successfully correct the errors when the robustness is lower than 0.95 [25]. Thus, channel coding methods are unsuitable for deep blind watermarking because the reconstruction robustness would be decreased under high-intensity or multiple distortions. Voting-based methods encode the same watermark in each small patch. When decoding the watermark, the final

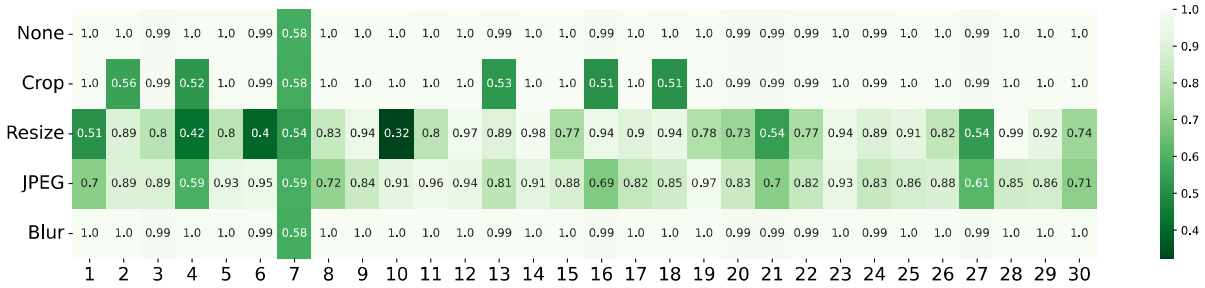

FIGURE 6. Visualized robustness per each bit under distortions. Each column represents the robustness of each bit of the watermark.

Algorithm 1 Watermark Encoding Algorithm

```
procedure Encode(coverImage, watermark, n)
    patches ← SplitImageIntoPatches(coverImage)
    for all patch ∈ patches do
        encodedPatch ← Encode(patch, watermark)
        watermark ← ShiftLeft(watermark, n)
    end for
    encodedImage ← CombinePatches(encodedPatch)
end procedure
```

Algorithm 2 Watermark Decoding Algorithm

```
procedure Decode(encodedImage, n, array)
    patches ← SplitImageIntoPatches(encodedImage)
    for all patch ∈ patches do
        watermarkPatch ← DecodePatch(patch)
        watermark ← ShiftRight(watermarkPatch, n)
        Push(array, watermark)
    end for
    decodedWatermark ← MajorityVoting(array)
end procedure
```

prediction of each bit of the watermark is decided by soft voting. However, voting from the same watermark in each small patch is not effective in terms of error correction when decoding the watermark under distortions. Since distortions tend to affect only specific bits depending on their types, not the whole data equally (see Fig. 6), if one bit is predicted incorrectly from one patch, the prediction from the other patches containing the same data would be incorrect too. Therefore, it is challenging to correct the mispredicted bit.

To address the misprediction problem of the voting method under distortions, we encode different representations of the watermark per each patch as shown in Algorithm 1. Specifically, watermark representations are encoded with a different degree of shifting in each patch. Such a bit-shifting technique prevents the mispredictions of specific bits by shifting the bits to other locations. For example, if a specific bit of the watermark is mispredicted under the cropped image, the mispredicted bit could be corrected by the other patches having the shifted watermarks, when the decoder correctly predicts the shifted watermark from them. As a result of Algorithm 2, the misprediction effect of distortions can be mitigated when the number of patches that correctly decode the watermark is larger than that of patches that mispredict the watermark by voting their prediction values.

The proposed strategy requires one deep blind watermarking model for a small patch, thereby reducing the training time and computational resources compared to training a model for a large image.

V. EXPERIMENTS

In our experiments, we encode a 30-bit watermark in the $3 \times 256 \times 256$ cover images. The hyperparameters and training environment are the same as Section III-A1. Our method consists of a deep blind watermarking model that leverages CFC+CONCAT as an encoder, and the watermarking strategy. It adopts the decoder and the discriminator architecture of HiDDeN to train the deep blind watermarking model, because we empirically observed that their architectural changes do not affect performance.

In our experiment, we first compare the robustness and imperceptibility of our method with the baseline methods. Next, in order to evaluate the robustness of our method in the traitor tracing scenario reflecting the distortions of the real world piracy sites, we perform in-depth analyses to show how distortion affects the robustness of our method when the intensity of distortions becomes larger or the distortions are combined with each other (i.e., multiple distortions). Finally, we analyze the performance of the proposed watermarking strategy in terms of training time and VRAM usage.

A. COMPARISON

We compare the proposed method with the following baseline methods.

- **HiDDeN.** It has two versions: identity and combined. The identity version has no attack simulation layer, whereas the combined version utilizes it to enhance the robustness.
- **Luo et al.'s scheme [24].** It adopts an adversarial network as an attack simulation layer based on HiDDeN. The adversarial network generates distortion that interrupts the reconstruction. By training with this distortion,

TABLE 4. Robustness comparison.

Distortion	HiDDeN (Identity)	HiDDeN (Combined)	Luo et al.	StegaStamp	Ours
JPEG (Q=50)	0.5000	0.6300	0.8500	**0.9959**	0.9265
Gaussian Noise (0.1)	0.6320	0.8040	0.8950	0.9994	**0.9997**
Salt and Pepper (0.05)	**0.9910**	0.9720	0.9570	0.9396	0.9616
Resize Width (0.5)	0.6650	0.6730	0.6710	0.6468	**0.6952**

TABLE 5. PSNR comparison.

	HiDDeN (Identity)	HiDDeN (Combined)	Luo et al.	StegaStamp	Ours
PSNR	39.46	32.30	33.70	32.24	38.21
Is model robust?	No	Yes	Yes	Yes	Yes

FIGURE 7. Examples of I_{cover}, I_{enc} and their differences. To visualize the difference explicitly, we amplify it with the scaling factor of 15.

the deep blind watermarking model achieves the 'distortion agnostic' robust reconstruction.

- **StegaStamp [25].** It leverages the attack simulation pipeline that serially adopts distortions such as wrapping, blurring, color manipulation, noising, and JPEG compression to enhance robustness under various distortions.

These baselines are suitable for comparison in practical deep blind watermarking, because (1) they have lower computational costs than methods leveraging diffusion for watermarks [36], [37], (2) they provide robustness under distortions that occur in digital piracy scenarios, and (3) they provide the advantage of analysis. Specifically, they show distinct differences in robustness and imperceptibility, which are affected by the attack simulation layer. By comparing our method with these baselines, we can evaluate the architectural advantage of CFC+CONCAT, which aims to increase robustness without the attack simulation layer. Furthermore, by comparing our method with HiDDeN and Luo et al.'s scheme, we can also evaluate how effectively the spatial repetition effect can be leveraged in the proposed strategy for increasing robustness and reducing computational costs.

We train the previous models to en/de-code 30-bit watermark in the $3 \times 256 \times 256$ image. For the combined version of HiDDeN and Luo et al.'s scheme, we compared our experimental results with their results given in the original papers. Table 4 shows the comparison results of our method with the baseline methods with regard to the robustness.

The evaluation uses the same distortion types and intensities as Luo et al.'s experiment to compare under the same conditions. As shown in the table, our method records comparable robustness with the previous methods. When we

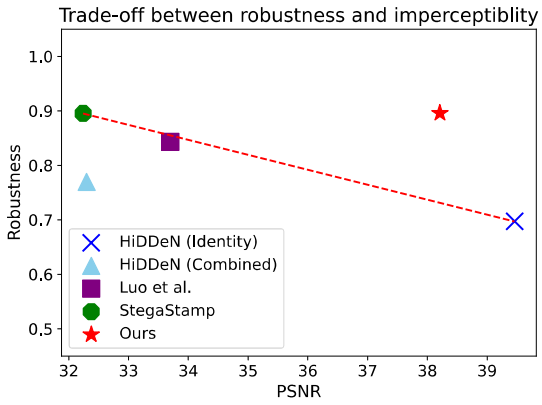

FIGURE 8. Trade-off between robustness and imperceptibility. The *y*-axis represents the average robustness of Table 4.

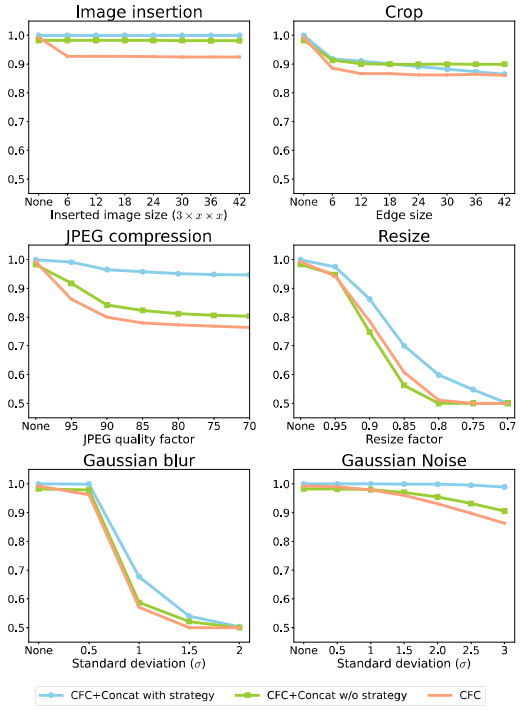

FIGURE 9. Evaluation of robustness with different distortion intensity.

compare the imperceptibility as shown in Table 5, our method achieves the highest PSNR among the methods designed to be robust, while being comparable to HiDDeN's identity version. Fig. 7 shows the examples of I_{cover} and I_{enc} of our method.

Thus, we can observe that our method mitigates the trade-off between robustness and imperceptibility, showing balanced and higher performance in terms of robustness and imperceptibility, which can be further demonstrated in Fig. 8. While the previous methods have only been able to improve robustness or imperceptibility by sacrificing the other, our method improves both robustness and imperceptibility by adopting appropriate architectural and strategic approaches.

TABLE 6. Robustness of the proposed method under multiple distortions. Color indicates the robustness of the method, with red indicating the lowest robustness and blue indicating the second lowest robustness.

Description			None	Ad insertion	Crop	Resize	JPEG compression
Single distortion			0.9998	0.9998	0.9178	0.9751	0.9909
Double distortion		Ad insertion		0.9998	0.9176	0.9724	0.9896
		Crop		0.9668	0.9309	0.8306	0.8687
		Resize		0.9040	**0.7849**	0.8031	**0.7513**
		JPEG compression		0.9789	0.8184	0.9500	0.9904
Triple distortion	Ad insertion	Ad insertion		0.9987	0.9043	0.9382	0.9426
		Crop		0.9208	0.9082	0.8030	0.8284
		Resize		0.9571	0.8212	0.9273	0.8593
		JPEG compression		0.9845	0.8197	0.9155	0.9904
	Crop	Ad insertion		0.9230	0.9088	0.8040	0.8281
		Crop		0.9070	0.9019	0.7948	0.8084
		Resize		0.7759	0.7989	0.7413	**0.6782**
		JPEG compression		0.8118	0.8003	0.7187	0.8294
	Resize	Ad insertion		0.9503	0.8220	0.9266	0.8595
		Crop		0.7901	0.8116	0.7562	0.6931
		Resize		0.8998	0.7710	0.7623	0.7895
		JPEG compression		0.8485	**0.6899**	0.8294	0.8604
	JPEG compression	Ad insertion		0.9526	0.8203	0.9147	0.9904
		Crop		0.7684	0.8007	0.7163	0.8290
		Resize		0.8959	0.7333	0.8691	0.8561
		JPEG compression		0.9874	0.8186	0.9153	0.9906

B. ROBUSTNESS ANALYSIS

1) ROBUSTNESS ACCORDING TO DISTORTION INTENSITY

In order to evaluate the robustness of our method under distortions in real world piracy sites, we manually explored 3,576 cartoon episodes distributed in the well-known three Korean cartoon piracy sites [45], [46], [47].[1] As a result, we choose the top four distortions that can be easily observed by human eyes or metadata such as image size and file format: advertisement insertion, crop, resize, and JPEG compression.

We first perform a comparative experiment to show the efficiency of the CFC+CONCAT with the proposed watermarking strategy. For the purpose, we split $3 \times 256 \times 256$ I_{cover} into four $3 \times 128 \times 128$ patches, and then encode 30-bit watermark in each patch after shifting. When reconstructing the watermark, we leverage soft-voting by averaging the prediction values.

Fig. 9 shows how the robustness (y-axis) changes with different distortion intensities (x-axis). x represents the size of the inserted image for 'Ad insertion'; the size of the cropped edge for 'crop'; the ratio of the length of one side of the distorted image to the original one for 'resize'; and compression quality factor for 'JPEG', respectively. According to Fig. 9, CFC+CONCAT achieves higher robustness than CFC. It implies applying the concatenation of I_{cover} on CFC brings about a regularization effect similar to [48], improving robustness of CFC. Furthermore, the proposed strategy achieves higher robustness in most of the distortion intensity, when combined with CFC+CONCAT. In the case of 'Ad insertion' and 'crop' that affect a certain (partial) portion of the image, the robustness remains almost constant after the first drop, demonstrating the decoding robustness is not much influenced by the distortion intensity. For example, when 25% of I_{enc} is cropped, the robustness is similar to that when 1% of the image is cropped. Since M_{in} is spatially repeated, the decoder can find the representation of M_{in} from the other parts of I_{enc}. On the contrary, for the other distortions that affect the whole image, the robustness degrades as the distortion intensity increases.

2) ROBUSTNESS UNDER MULTIPLE DISTORTIONS

We evaluate the robustness of our methods under multiple distortions of different permutations of up to 3 distortions, and show the results in Table 6. In the table, the intensity of distortions is the same as in Table 1. Interestingly, the adoption order of each distortion differently affects the robustness under the multiple-distortion conditions. For example, there is a significant difference between resize-then-JPEG compression distortion (which is 0.7513) and JPEG compression-then-resize distortion (which is 0.9500) in the double-distortion case. Fig. 10 illustrates the effect of the order in which distortions are applied on the robustness of each bit in the double-distortion case. The locations of bits affected by double distortion are similar, but the order in which the distortions are applied affects the amount of robustness degradation. In the triple-distortion case, the crop-then-resize-then-JPEG compression shows the lowest robustness, which is followed by resize-then-crop-then-JPEG compression. These results show that when distortions affecting the whole image (e.g., resize) are applied before the other distortions that partially affect the images (e.g., advertisement insertion or crop), robustness degrades more significantly than the opposite case. Thus, if the image is first distorted as a whole, the subsequent distortions' affection can be further amplified in the multiple-distortion environment,

[1] We accessed these sites on Jan. 29, 2023. Due to the nature of darkwebs, their URLs are frequently changed.

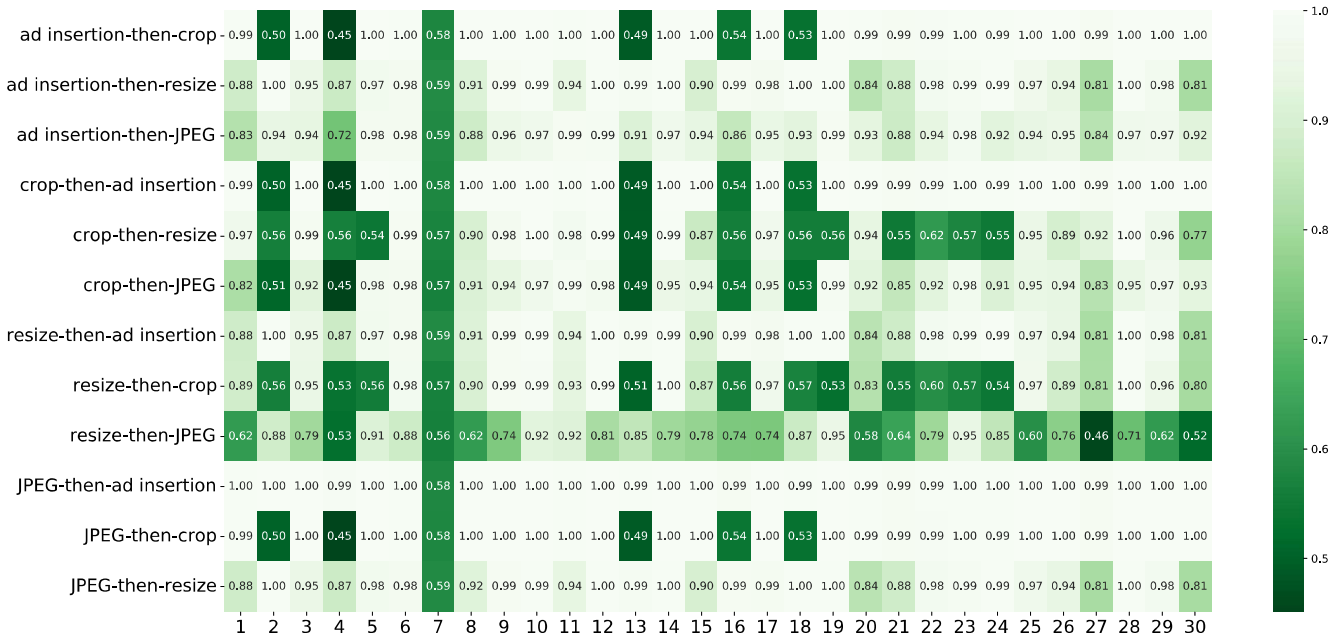

FIGURE 10. Visualized robustness per each bit under double-distortions. Each column represents the robustness of each bit of the watermark.

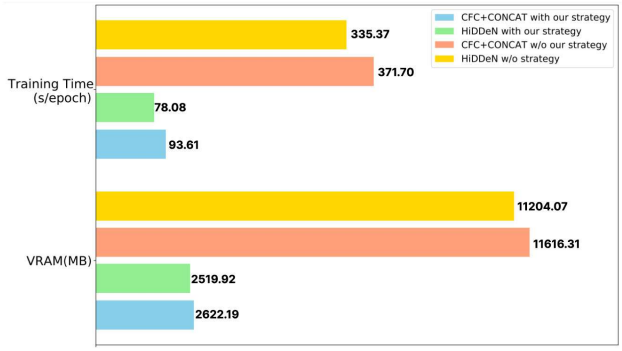

FIGURE 11. Evaluation of the training time and VRAM usage.

leading to the possible break of the watermarking technique effectively.

C. EFFICIENCY ANALYSIS

To evaluate the efficiency of the proposed watermarking strategy, we measure the training time and VRAM usage with/without our watermarking strategy, when en/de-coding 30-bit in 3 × 256 × 256 images. Since the strategy has no dependency on the model architecture, we choose HiDDeN and CFC+CONCAT in this experiment. Fig. 11 shows the training time and VRAM usage required to train each model with/without our watermarking strategy. Our strategy is effective in terms of VRAM usage or training speed, because training for small patches is cheaper than for the whole large image. Specifically, CFC+CONCAT for the small patch requires 2,622.19 MB of VRAM, while 11,616.31 MB is used to train for the whole image. Also, training with our strategy is approximately four times faster than training without it. Therefore, our watermarking strategy significantly reduces the training resources and time. Considering many of the current image sizes are likely to be larger than $3 \times 256 \times 256$ in the real world, our watermarking strategy would result in a higher performance gain in terms of training time and computational resources in practice.

VI. DISCUSSION

A. ROBUSTNESS UNDER DIFFERENT DISTORTIONS

We demonstrated our method enables the robust watermark reconstruction under the four practically representative distortions without any knowledge of them in the training procedure. However, there are also other various distortions in practice. Thus, we additionally perform a proof-of-concept experiment to evaluate the robustness of our method under the other distortions, i.e., adjust hue, adjust contrast, adjust brightness, salt-and-pepper, Gaussian noise, and Gaussian blur.

The proof-of-concept experimental result shows that our method enables robust reconstruction against almost every distortion. For adjusting hue, contrast, and brightness, for example, our method records higher robustness than 0.98 on average, regardless of their intensity. Although the robustness tends to decrease in proportion to distortion intensity in the case of salt-and-pepper, our method records the robustness of 0.9636 when the distortion rate for the image is 5%. When it comes to Gaussian noise and blur, the intensity of them is affected by standard deviation σ. Our method allows reconstruction with a robustness higher than 0.98 against Gaussian noise regardless of σ, whereas the robustness

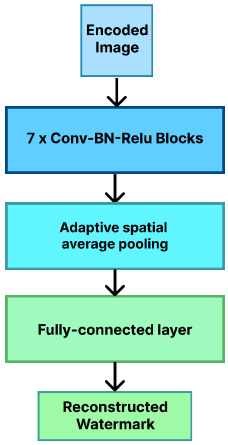

FIGURE 12. Illustration of the decoder architecture.

against Gaussian blur degrades as σ increases. We observed the robustness higher than 0.95 when σ becomes lower than 0.7. Accordingly, our method remains robust against the other diverse types of distortions.

However, how to make our method robust against special-purpose distortions such as adversarial noise [17], [18], [19] remains a challenging problem. One feasible solution to mitigate the problem is adopting adversarial training [17], [49], [50], which is an important future work.

B. EN/DE-CODING TIME

The proposed watermarking strategy has a computation overhead for en/de-coding linear to the number of small patches. On the basis of our experiment that measures CFC+CONCAT's en/de-coding time, en/de-coding for 4,000 small ($3 \times 128 \times 128$) patches takes longer than en/de-coding for 1,000 whole ($3 \times 256 \times 256$) images, when all en/de-coding procedures are performed serially. Specifically, we observe that the encoding time difference is 1.03 $ms/image$, and the decoding time difference is 3.27 $ms/image$. These differences are caused by the image splitting and merging procedures, because the total amount of computation for a large image is almost the same as that for four small patches. Further, the parallel en/de-coding process rather can reduce the en/de-coding time, but it may require more computational resources as a trade-off, which remains an open problem in the literature.

VII. CONCLUSION

In this paper, we proposed a novel deep blind watermarking method that contains a new encoder design of deep blind watermarking model and watermarking strategy to address the trade-off between robustness and imperceptibility. The proposed method achieves 5.46 higher PSNR than the baseline methods on average, guaranteeing comparable robustness. Also, our deep blind watermarking strategy reduces training time and VRAM usage by less than 1/4 when en/de-coding in $3 \times 256 \times 256$ image. Considering that many of the images' size in the real world is likely to be larger than $3 \times 256 \times 256$, our method would result in a higher performance gain in terms of training time and computational resource in practice. As a result, our method effectively mitigates the trade-off of deep blind watermarking with lightweight training.

APPENDIX
ADDITIONAL FIGURE

See Fig. 12.

REFERENCES

[1] *Intellectual Property and Youth Scoreboard*, European Union Intellectual Property Office, Alicante, Spain, Oct. 2019.
[2] United States Copyright Office. (May 2020). *Section 512 of Title 17, a Report of the Register of Copyrights*. [Online]. Available: https://www.copyright.gov/policy/section512/section-512-full-report.pdf
[3] D. Blackburn, J. A. Eisenach, and D. Harrison, *Impacts of Digital Video Piracy on the U.S. Economy*. White Plains, NY, USA: Nera Economic Consulting, Jun. 2019.
[4] Korea Creative Content Agency. (Dec. 2020). *Cartoon Industry White Paper*. [Online]. Available: https://www.kocca.kr/cop/bbs/list/B0000146.do?menuNo=201826
[5] N. Agarwal, A. K. Singh, and P. K. Singh, "Survey of robust and imperceptible watermarking," *Multimedia Tools Appl.*, vol. 78, no. 7, pp. 8603–8633, Apr. 2019.
[6] D. Boneh and J. Shaw, "Collusion-secure fingerprinting for digital data," *IEEE Trans. Inf. Theory*, vol. 44, no. 5, pp. 1897–1905, Sep. 1998.
[7] A. Fiat and T. Tassa, "Dynamic traitor tracing," *J. Cryptol.*, vol. 14, no. 3, pp. 211–223, Jun. 2001.
[8] A. Bamatraf, R. Ibrahim, and Mohd. N. B. M. Salleh, "Digital watermarking algorithm using LSB," in *Proc. Int. Conf. Comput. Appl. Ind. Electron.*, Dec. 2010, pp. 155–159.
[9] J. Abraham and V. Paul, "An imperceptible spatial domain color image watermarking scheme," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 31, no. 1, pp. 125–133, Jan. 2019.
[10] L. M. Marvel, C. G. Boncelet, and C. T. Retter, "Spread spectrum image steganography," *IEEE Trans. Image Process.*, vol. 8, no. 8, pp. 1075–1083, Aug. 1999.
[11] M. Noorkami and R. M. Mersereau, "Digital video watermarking in P-frames with controlled video bit-rate increase," *IEEE Trans. Inf. Forensics Security*, vol. 3, no. 3, pp. 441–455, Sep. 2008.
[12] X. Luo, Y. Li, H. Chang, C. Liu, P. Milanfar, and F. Yang, "DVMark: A deep multiscale framework for video watermarking," *IEEE Trans. Image Process.*, early access, Mar. 28, 2023, doi: 10.1109/TIP.2023.3251737.
[13] R. J. Mstafa and K. M. Elleithy, "Compressed and raw video steganography techniques: A comprehensive survey and analysis," *Multimedia Tools Appl.*, vol. 76, no. 20, pp. 21749–21786, Oct. 2017.
[14] O. F. A. Adeeb and S. J. Kabudian, "Arabic text steganography based on deep learning methods," *IEEE Access*, vol. 10, pp. 94403–94416, 2022.
[15] N. S. Kamaruddin, A. Kamsin, L. Y. Por, and H. Rahman, "A review of text watermarking: Theory, methods, and applications," *IEEE Access*, vol. 6, pp. 8011–8028, 2018.
[16] M. Shirali-Shahreza, "Text steganography by changing words spelling," in *Proc. 10th Int. Conf. Adv. Commun. Technol.*, Feb. 2008, pp. 1912–1913.
[17] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2014, *arXiv:1412.6572*.
[18] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," 2013, *arXiv:1312.6199*.
[19] S. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 86–94.
[20] J. Zhu, R. Kaplan, J. Johnson, and L. Fei-Fei, "HiDDeN: Hiding data with deep networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 657–672.
[21] S. K. Choi and J. Kwak, "Feature analysis and detection techniques for piracy sites," *KSII Trans. Internet Inf. Syst.*, vol. 14, no. 5, pp. 2204–2220, 2020.

[22] R. G. Gallager, "Low-density parity-check codes," *IRE Trans. Inf. Theory*, vol. 8, no. 1, pp. 21–28, Jan. 1962.

[23] R. C. Bose and D. K. Ray-Chaudhuri, "On a class of error correcting binary group codes," *Inf. Control*, vol. 3, no. 1, pp. 68–79, Mar. 1960.

[24] 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.

[25] M. Tancik, B. Mildenhall, and R. Ng, "StegaStamp: Invisible hyperlinks in physical photographs," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2114–2123.

[26] M. Ahmadi, A. Norouzi, S. M. R. Soroushmehr, N. Karimi, K. Najarian, S. Samavi, and A. Emami, "ReDMark: Framework for residual diffusion watermarking on deep networks," 2018, *arXiv:1810.07248*.

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

[28] K. A. Zhang, A. Cuesta-Infante, L. Xu, and K. Veeramachaneni, "SteganoGAN: High capacity image steganography with GANs," 2019, *arXiv:1901.03892*.

[29] Y. Liu, M. Guo, J. Zhang, Y. Zhu, and X. Xie, "A novel two-stage separable deep learning framework for practical blind watermarking," in *Proc. 27th ACM Int. Conf. Multimedia*, Oct. 2019, pp. 1509–1517, doi: 10.1145/3343031.3351025.

[30] C. Zhang, A. Karjauv, P. Benz, and I. S. Kweon, *Towards Robust Deep Hiding Under Non-Differentiable Distortions for Practical Blind Watermarking*. New York, NY, USA: Association for Computing Machinery, 2021, pp. 5158–5166, doi: 10.1145/3474085.3475628.

[31] 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. Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 10223–10234, 2020.

[32] J. Hayes and G. Danezis, "Generating steganographic images via adversarial training," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1951–1960.

[33] S.-M. Mun, S.-H. Nam, H.-U. Jang, D. Kim, and H.-K. Lee, "A robust blind watermarking using convolutional neural network," 2017, *arXiv:1704.03248*.

[34] J. Qin, J. Wang, Y. Tan, H. Huang, X. Xiang, and Z. He, "Coverless image steganography based on generative adversarial network," *Mathematics*, vol. 8, no. 9, p. 1394, Aug. 2020.

[35] D. Hu, L. Wang, W. Jiang, S. Zheng, and B. Li, "A novel image steganography method via deep convolutional generative adversarial networks," *IEEE Access*, vol. 6, pp. 38303–38314, 2018.

[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] H. Fang, Z. Jia, Z. Ma, E.-C. Chang, and W. Zhang, "PIMoG: An effective screen-shooting noise-layer simulation for deep-learning-based watermarking network," in *Proc. 30th ACM Int. Conf. Multimedia*, Oct. 2022, pp. 2267–2275.

[38] A. Paszke, "Pytorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, and R. Garnett, Eds. Red Hook, NY, USA: Curran Associates, 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[39] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 740–755.

[40] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[41] 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.

[42] 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.

[43] C.-C. Chang, J.-Y. Hsiao, and C.-S. Chan, "Finding optimal least-significant-bit substitution in image hiding by dynamic programming strategy," *Pattern Recognit.*, vol. 36, no. 7, pp. 1583–1595, Jul. 2003. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320302002893

[44] D. Opitz and R. Maclin, "Popular ensemble methods: An empirical study," *J. Artif. Intell. Res.*, vol. 11, pp. 169–198, Aug. 1999.

[45] *Newtoki*. Accessed: Jan. 29, 2023. [Online]. Available: https://newtoki215.com/

[46] *Manatoki*. Accessed: Jan. 29, 2023. [Online]. Available: https://manatoki215.net/

[47] *Hoducomics*. Accessed: Jan. 29, 2023. [Online]. Available: https://hodu292.net/toon

[48] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2261–2269.

[49] A. Shafahi, M. Najibi, M. A. Ghiasi, Z. Xu, J. Dickerson, C. Studer, L. S. Davis, G. Taylor, and T. Goldstein, "Adversarial training for free!" in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 3358–3369.

[50] T. Bai, J. Luo, J. Zhao, B. Wen, and Q. Wang, "Recent advances in adversarial training for adversarial robustness," 2021, *arXiv:2102.01356*.

BOSUNG YANG received the B.S. degree in cyber security from Ajou University, Suwon, in 2021. He is currently pursuing the M.S. degree with the Department of Computer Science and Engineering, College of Informatics, Korea University, South Korea. His research interest includes AI security and privacy.

GYEONGSUP LIM received the B.S. degree in computer science from Korea University, Seoul, South Korea, in 2021, where he is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering, College of Informatics. His research interest includes safe AI.

JUNBEOM HUR received the B.S. degree from Korea University, Seoul, South Korea, in 2001, and the M.S. and Ph.D. degrees from KAIST, in 2005 and 2009, respectively, all in computer science. He was with the University of Illinois at Urbana–Champaign, as a Postdoctoral Researcher, from 2009 to 2011. He was with the School of Computer Science and Engineering, Chung-Ang University, South Korea, as an Assistant Professor, from 2011 to 2015. He is currently a Professor with the Department of Computer Science and Engineering, Korea University. His research interests include information security, cloud computing security, network security, and applied cryptography.