OXFORD

The Computer Journal, 2024, 1–13
https://doi.org/10.1093/comjnl/bxae014
Original Article

SEDD: Robust Blind Image Watermarking With Single Encoder And Dual Decoders

Yuyuan Xiang[1,2], Hongxia Wang[1,2,*], Ling Yang[1,2], Mingze He[1,2] and Fei Zhang[1,2]

[1]School of Cyber Science and Engineering, Sichuan University, Chengdu 610065, China
[2]Key Laboratory of Data Protection and Intelligent Management (Sichuan University), Ministry of Education, 610207, China

*Corresponding author: hxwang@scu.edu.cn

Blind image watermarking is regarded as a vital technology to provide copyright of digital images. Due to the rapid growth of deep neural networks, deep learning-based watermarking methods have been widely studied. However, most existing methods which adopt simple embedding and extraction structures cannot fully utilize the image features. In this paper, we propose a novel Single-Encoder-Dual-Decoder (SEDD) watermarking architecture to achieve high imperceptibility and strong robustness. Precisely, the single encoder utilizes normalizing flow to realize watermark embedding, which can effectively fuse the watermark and cover image. For watermark extraction, we introduce a parallel dual-decoder to improve the imperceptibility and extracting ability. Extensive experiments demonstrate that better watermark robustness and imperceptibility are obtained by SEDD architecture. Our method achieves a bit error rate less than 0.1% under most attacks such as JPEG compression, Gaussian blur and crop. Besides, the proposed method also obtains strong robustness under combined attacks and social platform processing.

Keywords: Blind watermarking; Deep learning; Robustness

1. INTRODUCTION

With the rapid development of multimedia technology and the widespread use of online social platforms such as WeChat and Twitter, it has become increasingly easy and convenient for people to share various forms of digital media, including images, audio and videos. However, due to the ease of copying and distributing digital media, the copyrights of original works are vulnerable to infringement. For content creators, unauthorized use of digital media always leads to financial losses and reputation risks. Therefore, the need for copyright protection has become more essential than ever. Digital watermarking technology provides an effective solution by embedding identification information into digital media to achieve copyright protection and leak source tracing [1–4]. Specifically, digital image watermarking aims to embed watermark into cover image invisibly, where the watermark can be correctly extracted under image distortions.

Traditional methods [5–10] usually hide information by modifying the coefficients in the handcrafted features of the cover image. Based on different embedding domains, existing research on traditional watermarking algorithms mainly focused on transform domain watermarking algorithms [5–7] and moment-based watermarking algorithms [8–10]. Recently, various deep learning-based methods [11–20] have been proposed to solve watermarking problem, aiming to extract better image features. Zhu *et al.* [11] proposed HiDDeN, an end-to-end image watermarking framework that achieves robustness against various attacks by introducing a noise layer between the encoder and decoder. This framework also introduced adversarial training to improve the imperceptibility for the first time. Ahmadi *et al.* [12] proposed a watermarking scheme based on residual network, which further enhanced the robustness. However, the above methods struggle to resist high-intensity JPEG compression. In order to address the non-differentiable problem of JPEG compression in the end-to-end network, Liu *et al.* [15] proposed a two-stage separable method that separates the training process of the encoder and the decoder. This framework is also robust against black-box noises. Besides, Jia *et al.* [16] proposed a novel Mini-Batch of Real and Simulated JPEG Compression method, which significantly enhances the robustness against JPEG compression. Luo *et al.* [17] used convolutional neural networks to generate adversarial samples to replace the distorted watermarked images attacked by noise layer, aiming for improving the robustness against unknown attacks. Fernandez *et al.* [18] proposed a self-supervised watermarking algorithm that can achieve watermark embedding in the latent spaces of any pre-trained network. Ma *et al.* [19] introduced the invertible neural network into blind watermarking, and achieved high imperceptibility and strong robustness at the same time by combining invertible and non-invertible mechanisms. Fang *et al.* [20] proposed a decoder-driven watermarking network called DeEND, which could effectively couple the encoder and decoder. This architecture can be applied to existing watermarking networks and improve performance without changing the backbone.

Furthermore, there are works researching deep learning-based steganography and watermarking [21–23] against cross-media distortions, such as print-capture and screen-shooting. What is more, there are methods [24, 25] proposed to apply neural style transfer to image steganography, which realized information hiding by transferring the secret information as the content to style transfer images. All these deep learning-based methods achieve great performance in terms of image quality and robustness, but their performance is limited by simple watermarking embedding structures. Besides, most of them do not consider the effect of decoder on imperceptibility. This paper mainly focuses on the embedding and extraction schemes of the watermark. For the encoder, we improve the fusion method of the cover image and

Received: July 13, 2023. **Revised:** January 11, 2024. **Accepted:** January 17, 2024

watermark to achieve deeper information fusion. In addition, we design a dual-decoder architecture to influence the results of the encoder in reverse, optimizing the visual quality of the watermarked image.

The rest of the paper is organized as follows. The research gaps and contributions are described in Section 2. The details of the proposed architecture are introduced in Section 3. The results from experiment are presented in Section 4. Finally, conclusions are drawn in Section 5.

2. RESEARCH GAPS AND CONTRIBUTIONS

2.1. Research Gaps

Existing deep learning-based image watermarking schemes have not only achieved end-to-end watermark embedding and extraction, improving the efficiency of watermark schemes, but have also obtained robustness against various image distortions through differentiable noise layer. Although current deep learning-based robust image watermarking methods have achieved good performance, there are still two problems that require further optimization. Firstly, the current watermark fusion operation which uses a single concatenation for watermark embedding is so simplistic that lacks the ability to fully fuse the information of the cover image and the watermark. In addition, existing methods often improve visual quality by optimizing the encoder or reducing the embedding strength, while the architecture of the decoder and the choice of noises in the noise layer also greatly affect the morphological character of the watermark information in the watermarked image. Due to the complexity of image distortion, a combined noise layer composed of various noises is generally used for training. In this case, existing decoding methods that only perform downsampling through convolution layers will inevitably result in corner artifacts on the watermarked image, which causes the reduction of visual quality.

Recently, the normalizing flow has been widely used in various fields, such as image super-resolution [26], image scaling [27] and image compression [28], and has achieved excellent results. The normalizing flow was first introduced [29, 30] to transform a simple probability distribution into a complex distribution by a sequence of invertible mappings. Subsequently, Dinh *et al.* [31] proposed RealNVP, which combines additive and multiplicative coupling layers to form a generalized affine coupling layer. This method also introduced convolution layers in the coupling model to better handle image tasks. Glow [32] introduces 1x1 invertible convolution and actnorm layers on the basis of RealNVP to achieve better generation results. In addition, many studies applied normalizing flow to image steganography task [33–35], achieving outstanding imperceptibility with high payload capacity. HiNet [34] treated hiding and revealing procedure as forward and backward processes of the normalizing-flow model. To improve hiding capacity and reconstruction quality, Xu *et al.* [35] proposed content-aware noise projection based on conditional flow block, which can provide effective guidance for revealing, thereby retaining more high-frequency information of the secret image. The structure of the normalizing flow corresponds to the process of watermark embedding. Specifically, the normalizing flow has two inputs, and its process involves the repeated fusion of features from these inputs. Therefore, we propose to apply normalizing flow to the watermark network as a fundamental component for the encoder, enabling better fusion of the cover image and watermark.

2.2. Contributions

To solve the above problems, we propose a new image watermarking structure named Single Encoder Dual Decoder (SEDD). Different from the encoder in existing schemes, we use the flow-based model to realize the watermark embedding, which fully fuses the information of the watermark and the cover image through the combination of coupling layers. The feature reuse in coupling layers enables the watermark to be embedded into the cover image in a more redundant form, further enhancing the robustness of the model. Since the entire network is trained end-to-end, the gradient information of the decoder can also influence the encoder, which enables the encoder to learn how to modify its output to minimize the image loss and message loss. Therefore, improving the structure of the decoder can also enhance the imperceptibility. To mitigate the corner artifacts, we construct a parallel dual-decoder, which introduces global information through fully connected (FC) layers to break through the locality of convolution layers. By incorporating FC layers, the decoder can access and utilize information from a wider context, prompting the encoder to embed the watermark information in a more global manner and thereby improving the visual quality of the watermarked image. What is more, distributing the task of decoding to two decoders can reduce the burden of each decoder and improve robustness.

In summary, we list the contributions in this paper as below:

(1) We propose a novel architecture SEDD for image watermarking, which can fully exploit image features to obtain better watermarked image and extracted watermark.
(2) We present a flow-based encoder that can effectively couple the watermark and cover image, enabling a more redundant approach for watermark embedding.
(3) We propose a parallel dual-decoder to improve the visual quality and extraction ability. It is able to take full advantage of the global information and local information of the image.

3. METHOD

3.1. Overall Architecture

As shown in Fig. 1, this SEDD architecture includes five components: (1) Watermark Processor *WP*, which receives the watermark message $M \in \{0,1\}$ with length L as input and outputs the watermark feature $M_{en} \in \mathbb{R}^{3\times H\times W}$. (2) Encoder E, which is fed with the cover image $I_{co} \in \mathbb{R}^{3\times H\times W}$ and the watermark feature M_{en}, and produces the watermarked image $I_{en} \in \mathbb{R}^{3\times H\times W}$. (3) Noise layer N, which receives I_{en} as input and distorts the watermarked image to generate the noised image I_{no}. (4) Decoders D, consisting of the convolution decoder D_{conv} and the FC decoder D_{fc}. Both D_{conv} and D_{fc} receive the noised image I_{no} as input and produce the decoded watermarks M_{de_1} and M_{de_2}, respectively. And generating the final decoded watermark M_{de} by fusing M_{de_1} and M_{de_2}. (5) Adversary discriminator Ad, which receives the image I_{co} or I_{en} to judge whether the input image is a watermarked image or not.

3.2. Watermark Processor

In order to better realize the fusion of the watermark and cover image, we propose the watermark processor *WP* to process the watermark. The watermark is a binary message $M \in \{0,1\}$ of length L. Figure 2 shows the processing of the watermark. To align the watermark with the number of channels of the cover image, we first use three different FC layers to generate redundant watermarks with length $\hat{L} = 256$. Subsequently, reshape the

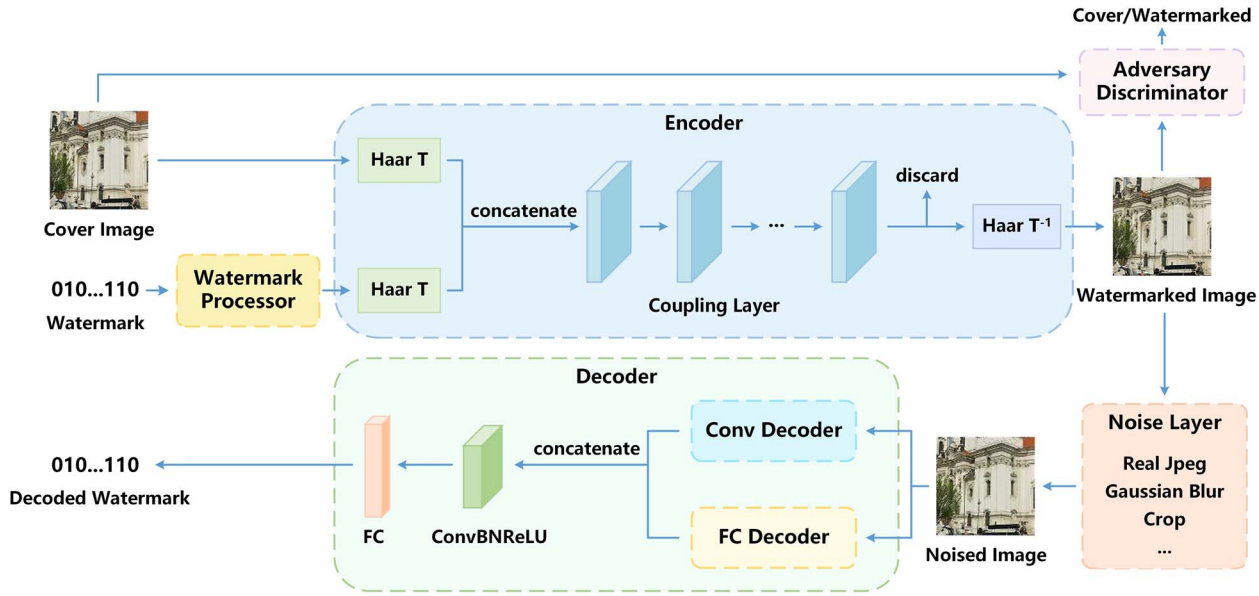

Figure 1. Overview of our proposed SEDD watermarking framework. The watermark processor diffuses the watermark to the same size of the cover image. The encoder embeds the watermark feature into the cover image using coupling layers. The noise layer includes a variety of attacks. The decoder uses two branches for watermark extraction and fuses their outputs as the final extracted watermark. The adversary discriminator is used to distinguish the cover image and the watermarked image.

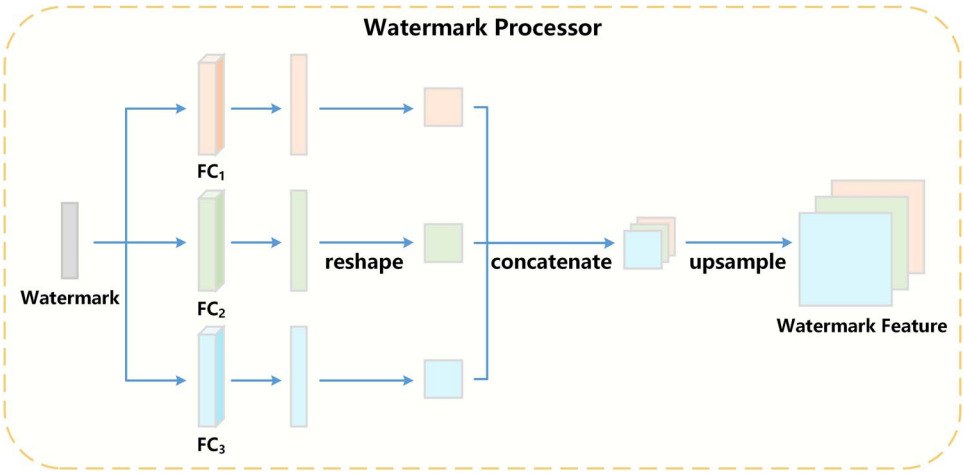

Figure 2. The procedure of the watermark processor.

redundant watermarks to size 16×16. Then the three watermarks are concatenated to produce three-dimensional feature. Finally, it is upsampled to $3 \times H \times W$ by two-dimensional nearest interpolation, where the H, W are the height and width of the cover image. The watermark feature $M_{en} \in \mathbb{R}^{3\times H\times W}$ is then fed into the encoder to be embedded in the cover image. For watermark processing operations:

$$WP(M) = O_{up}(O_{cat}(\Gamma_{fc_1}(M), \Gamma_{fc_2}(M), \Gamma_{fc_3}(M))), \tag{1}$$

where Γ_{fc}, O_{cat} and O_{up} refer to operations FC, Concatenate and Upsample, respectively.

3.3. Encoder

Due to the powerful representation ability and the architecture, normalizing flow is naturally and intuitively suitable for the image watermarking task. Therefore, we use flow-based encoder to embed the watermark message M into the cover image in a more imperceptible and robust way.

In Fig. 3, we build up flow blocks based on IRN [27]. We first transform the M_{en} and the I_{co} into frequency domain by Haar transform, and both outputs' sizes are $12 \times H/2 \times W/2$. Taking the feature maps after the Haar transform as input, N coupling layers are used to fuse the watermark and cover image. For the i-th coupling layer, the inputs are x_m^i and x_{co}^i, and the outputs x_m^{i+1} and x_{co}^{i+1} are formulated as

$$x_{co}^{i+1} = x_{co}^{i} + \phi(x_m^{i+1}) \tag{2}$$

$$x_m^{i+1} = x_m^i \odot \exp(\rho(x_{co}^{i+1})) + \eta(x_{co}^{i+1}), \tag{3}$$

where $\exp(\cdot)$ is exponential operator, and $\odot$ is the element-wise product. Here, $\phi(\cdot)$, $\rho(\cdot)$ and $\eta(\cdot)$ are arbitrary functions and we employ dense block in [36] to represent them. After the last coupling layer, we obtain the outputs x_m^{N+1} and x_{co}^{N+1}. Finally, the

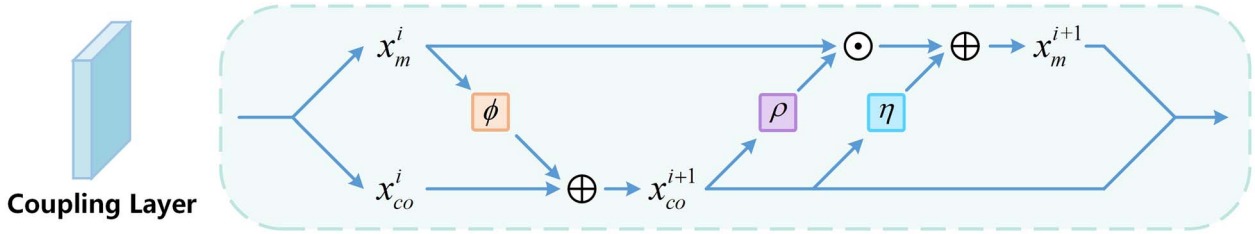

Figure 3. The structure diagram of the coupling layer.

watermarked feature x_{co}^{N+1} go through an inverse Haar transform to generate the watermarked image $I_{en} \in \mathbb{R}^{3\times H\times W}$.

We can regard the operation of the coupling layer as two interactive branches. On one hand, the watermark is embedded into the cover image through additive coupling. Meanwhile, the information of the cover image is also incorporated into the watermark features through affine coupling [31]. And the corresponding outputs will be fed into the next layer as the new cover image and watermark. During the coupling progress, the information of the watermark and image is fully integrated with each other.

To make watermarked image and cover image as visually similar as possible, the objective of encoder is to minimize the L_2 distance between I_{co} and I_{en} by updating θ_E with

$$L_E = \|I_{co} - I_{en}\|_2^2 = \|I_{co} - E(\theta_E, I_{co}, M)\|_2^2, \tag{4}$$

where I_{co} is the cover image, I_{en} is the watermarked image, M is the watermark message and $\theta_E \in \mathbb{R}$ represents learnable parameters in the encoder.

3.4. Noise layer

To ensure robustness of the watermark, we utilize a noise layer to simulate possible distortions in a real scene. Due to the variety of image distortions, we train with an extensive family of attacks including identity, real JPEG, JPEG-Mask, crop, Gaussian blur and Gaussian noise, where JPEG-Mask is the simulated differentiable JPEG noise [16].

3.5. Decoder

In deep learning-based watermarking architectures, decoder is also a key component affecting robustness and imperceptibility.

In the watermark processor, we perform diffusion and reshape operations on the watermark message, which obtain 16 × 16 diffused watermark. In the subsequent upsample process, the nearest interpolation will lead to a block distribution of the watermark information. Each watermark block represents 1 bit information of the diffused watermark. Since there is no direct correlation between the values of the diffused watermark bits, there are usually clear boundaries between blocks. When we use several ConvBNReLU blocks (consist of convolution layer, batch normalization and ReLU activation) to downsample the noised image to extract the watermark, the convolution operation is mainly performed within each watermark block because the size of the convolution kernel is smaller than the size of the watermark block. Thus, the convolution layer mainly represents the information of the local watermark block, resulting in the lack of global information. Optimizing with the constraint of image loss, in order to reduce the visual impact of the watermark on the watermarked image, the network tends to concentrate the information of the block into the corner, forming corner artifacts. In the early stage of training, the watermark is mainly presented in the form of brightly colored irregular blocks. During the training progress, the network gradually aggregates this color information to the corners of watermark blocks to improve invisibility. Figure 4 shows the morphological changes of watermark features during the training process.

To mitigate the corner artifacts, we construct a second decoder by fully connected layers to introduce global information. Figure 5 shows the detailed architecture of the two decoders. The convolution decoder D_{conv} consists of seven ConvBNReLU blocks, where the downsample operation is carried out three times in the blocks. For the FC decoder D_{fc}, corresponding to watermark diffusion, we process RGB channels through three different FC layers, compressing the image information from size $3 \times H \times W$ to $3 \times 16 \times 16$. Therefore, each input pixel of the noised image is related to the output watermark bit, and impelling each pixel of the watermarked image to be related to the global watermark information as much as possible. As shown in Fig. 1, the two decoders generate $3 \times 16 \times 16$ watermark features, and then fuse the two features through concatenation and a convolution layer. Finally, a linear layer is utilized to obtain the extracted watermark message M_{de}. Among them, the intervention of the FC layers breaks through the limitation that the operations of the convolution layer are concentrated in the block, resulting in better visual quality of the watermarked image. Besides, dual decoder can reduce the decoding burden of single decoder, resulting in an improvement in the extraction capability of the model. For extracting operations in Decoder:

$$D(I_{no}) = \Gamma_{FC}(\Gamma_{ConvBlk}(O_{cat}(\Gamma_{D_{conv}}(I_{no}), \Gamma_{D_{fc}}(I_{no})))), \tag{5}$$

where $\Gamma_{D_{conv}}$, $\Gamma_{D_{fc}}$ and $\Gamma_{ConvBlk}$ refer to convolution decoder, FC decoder and ConvBNReLU, respectively.

To ensure the accuracy of watermark extraction, the object of decoder training is to minimize the L_2 distance between M and M_{de} by updating θ_D with

$$L_D = \|M - M_{de}\|_2^2 = \|M, D(\theta_D, I_{no})\|_2^2, \tag{6}$$

where M is the watermark message, M_{de} is the decoded watermark and $\theta_D \in \mathbb{R}$ represents learnable parameters in the decoder.

3.6. Adversary discriminator

To acquire better imperceptibility of watermark, an adversarial network is adopted in the framework, which consists of several convolution layers and a global average pooling layer. The discriminator performs as an adversary of encoder and tries to correctly

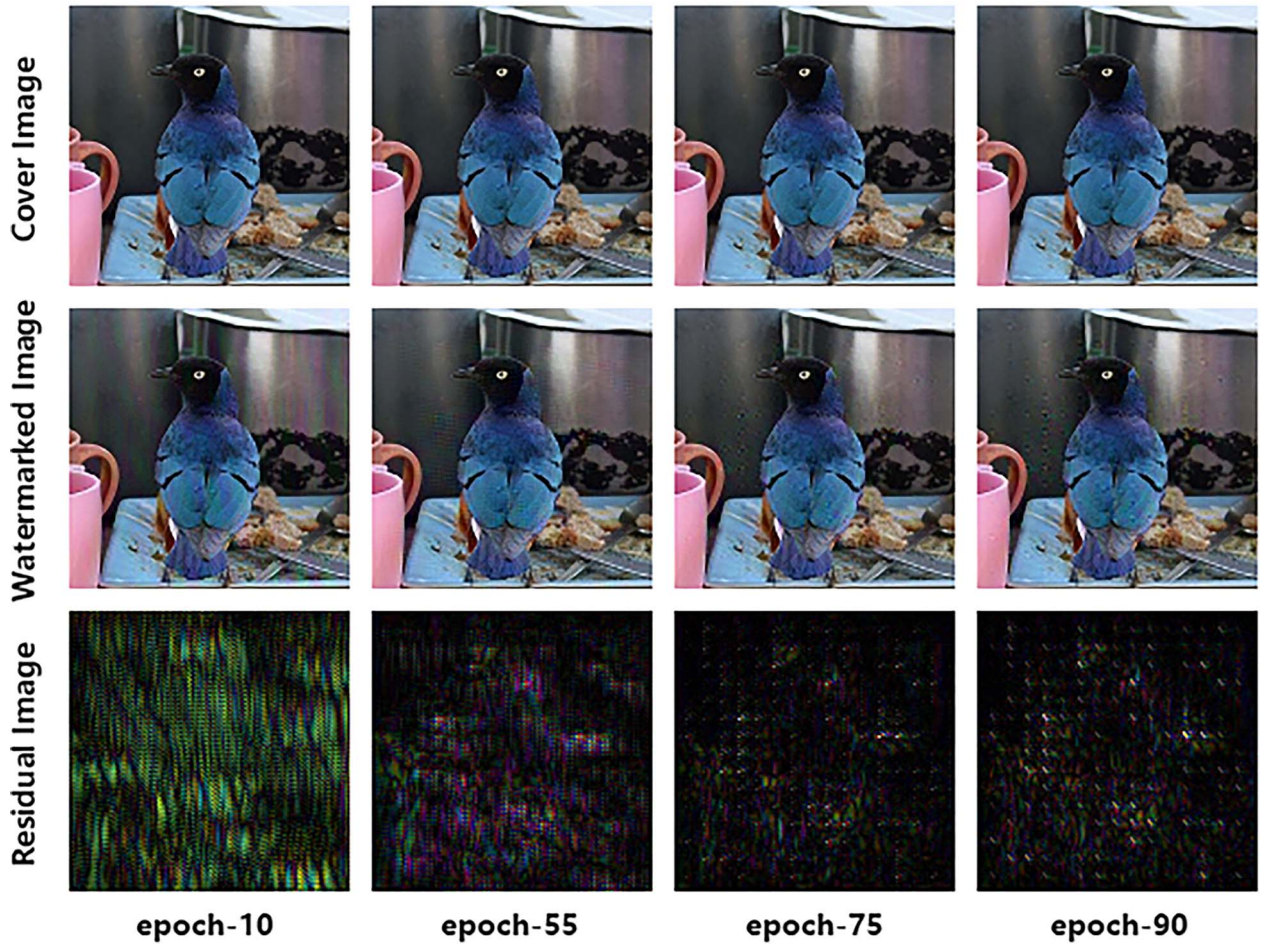

Figure 4. Visual quality of the watermarked images generated by convolution decoder-based model. From left to right are the results of different epochs. Top: the cover image I_{co}. Second row: the watermarked image I_{en}. Bottom: the magnified residual $|I_{co} - I_{en}|$.

distinguish between I_{co} and I_{en}, which is realized by updating θ_{Ad} with

$$L_{Adv} = \log(1 - Ad(\theta_{Ad}, E(I_{co}, M))) + \log(Ad(\theta_{Ad}, I_{co})), \tag{7}$$

where $\theta_{Ad} \in \mathbb{R}$ denotes the learnable parameters in the adversary discriminator.

In the meanwhile, the watermarked image attempts to confuse the discriminator, which will result in a similar image as the cover image. We use the L_{Ad} loss to constrain this process by updating θ_E with

$$L_{Ad} = \log(Ad(\theta_{Ad}, I_{en})) = \log(Ad(\theta_{Ad}, E(\theta_E, I_{co}, M))) \tag{8}$$

3.7. Loss Function

In total, the target loss L of the whole network consists of image loss L_E, message loss L_D and adversarial loss L_{Ad}, which can be formulated as

$$L = \lambda_E L_E + \lambda_D L_D + \lambda_{Ad} L_{Ad}, \tag{9}$$

where λ_E, λ_D and λ_{Ad} are weight factors (λ_E, λ_D and $\lambda_{Ad} > 0$). The total loss L is for the encoder and decoder, and loss L_{Adv} for the adversary discriminator.

Table 1. The details of the experimental environment.

Environment	Configuration
Operating system	Windows 10
RAM	32GB
GPU	Nvidia GeForce RTX 2080 Ti
Programming language	Python 3.7
Development framework	Pytorch 1.10.2

4. EXPERIMENTS

4.1. Implementation Details

To verify the effectiveness of the proposed SEDD, we utilize the COCO dataset [37] for training and evaluation. All models are trained on 10 000 image, and evaluated on the other 5000 images. Images are rescaled to size 128 × 128. Watermark messages are sampled randomly at each bit. The watermark length is 30 bits. For the weight factors of the loss function, we choose $\lambda_E = 2$, $\lambda_D = 1$ and $\lambda_{Ad} = 0.001$. The whole framework is implemented by PyTorch [38] and executed on NVIDIA RTX 2080ti. Table 1 indicates the details of the experimental environment. The Adam [39] optimizer is adopted with standard hyperparameters. The learning rate is set to be 1×10^{-4}, and the batch size is set to be 8.

For the imperceptibility of the watermarked image, we use peak signal-to-noise ratio (PSNR) [40] and structural similarity (SSIM)

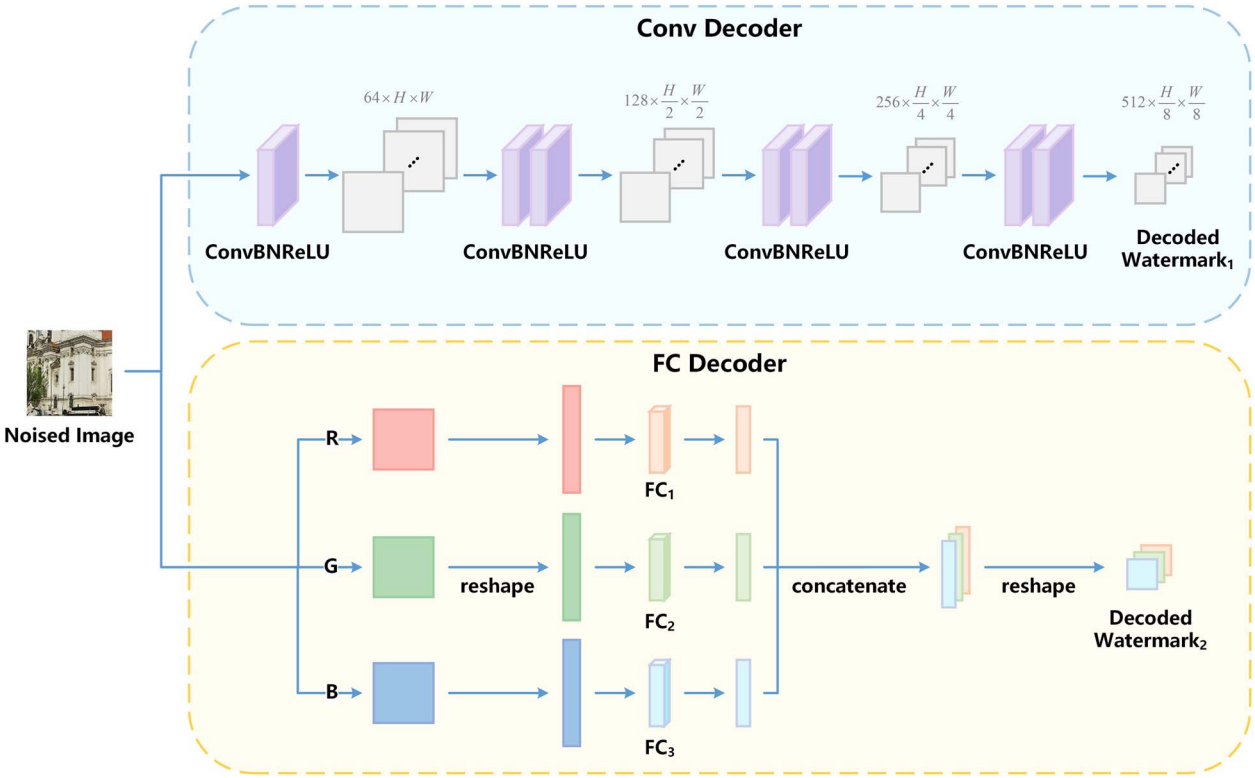

Figure 5. The dual decoder which consists of the convolution decoder and the fully connected decoder.

[41] as metrics. To measure the robustness of our proposed framework, we utilize bit error ratio (BER) and file-level correctness ratio (FCR) for evaluation. In practical applications, if errors occur in the extracted watermark, even if there is only one wrong bit, the watermark is meaningless to the image. As a result, we define FCR as the ratio of the number of images that can fully extract the watermark to the number of all test images:

$$FCR = \frac{NCI}{TNI} \times 100\%, \tag{10}$$

where NCI is the number of images that can fully extract the watermark (the 30 bits watermark extracted from the watermarked image has no errors) and TNI is the total number of test images.

4.2. Visual Quality

For the combined model of our framework, the average PSNR and SSIM reach 35.79dB and 0.9074, respectively. To intuitively illustrate imperceptibility, we show the visual results of the watermarked image in Fig. 6. The magnified residual $|I_{en} - I_{co}|$ is also shown to more clearly visualize the difference between the cover image I_{co} and the watermarked image I_{en}. We can see that our model achieves excellent imperceptibility. Furthermore, the residual signal shows that the watermark is adaptively embedded in the texture area of the cover image.

4.3. Robustness Comparison with State-of-the-art Methods

In this section, we compare our method with four SOTA methods: HiDDeN [11], MBRS [16], SSL [18] and CIN [19]. We test MBRS, SSL and CIN with the open source pre-trained models. For HiDDeN, we train combined model with noise layer including JPEG-Mask, crop, cropout, dropout and Gaussian blur. For a fair comparison, we adjust the PSNR of all methods to 35.7dB. For SSL, we set the target PSNR to 35.7dB. For the other three methods, we adjust the PSNR by the strength factor. All models are trained and evaluated with the same image size 128 × 128 × 3 and the same watermark length 30 bits.

4.3.1. *Robustness against Individual Attacks*

We first show the robustness against individual attacks of the proposed method. Figure 7 shows the visual results against various distortions. Each column indicates the result against a specific attack.

Table 2 reports the BER of each method on the individual attacks. Overall, our model reaches optimal robustness against most attacks while maintaining high visual quality. Especially for JPEG compression, our method achieves a significant reduction in the BER, which is approaching 0%. Besides, under two different types of blurring, our model can obtain excellent extraction results. Table 3 shows the comparison results of FCR. Our method achieves an FCR above 0.98 on individual attacks except for median blur, which means that the vast majority of watermarked images can still fully extract the watermark after being attacked. Table 4 gives a more comprehensive comparison across a range of attack intensities. Against JPEG compression, our model outperforms other methods at different compression intensities. Against Gaussian blur, our model reaches comparable performance of CIN at standard deviation of 2 and below. For Gaussian noise, our model performs better at high noise intensities. For Brightness, factors below 1 and above 1 represent dimming and brightening, respectively. It is evident that our model has a BER of 0% against dimming, but it is slightly less robust against brightening. It can also be seen from Table 2 that the BER of all schemes are

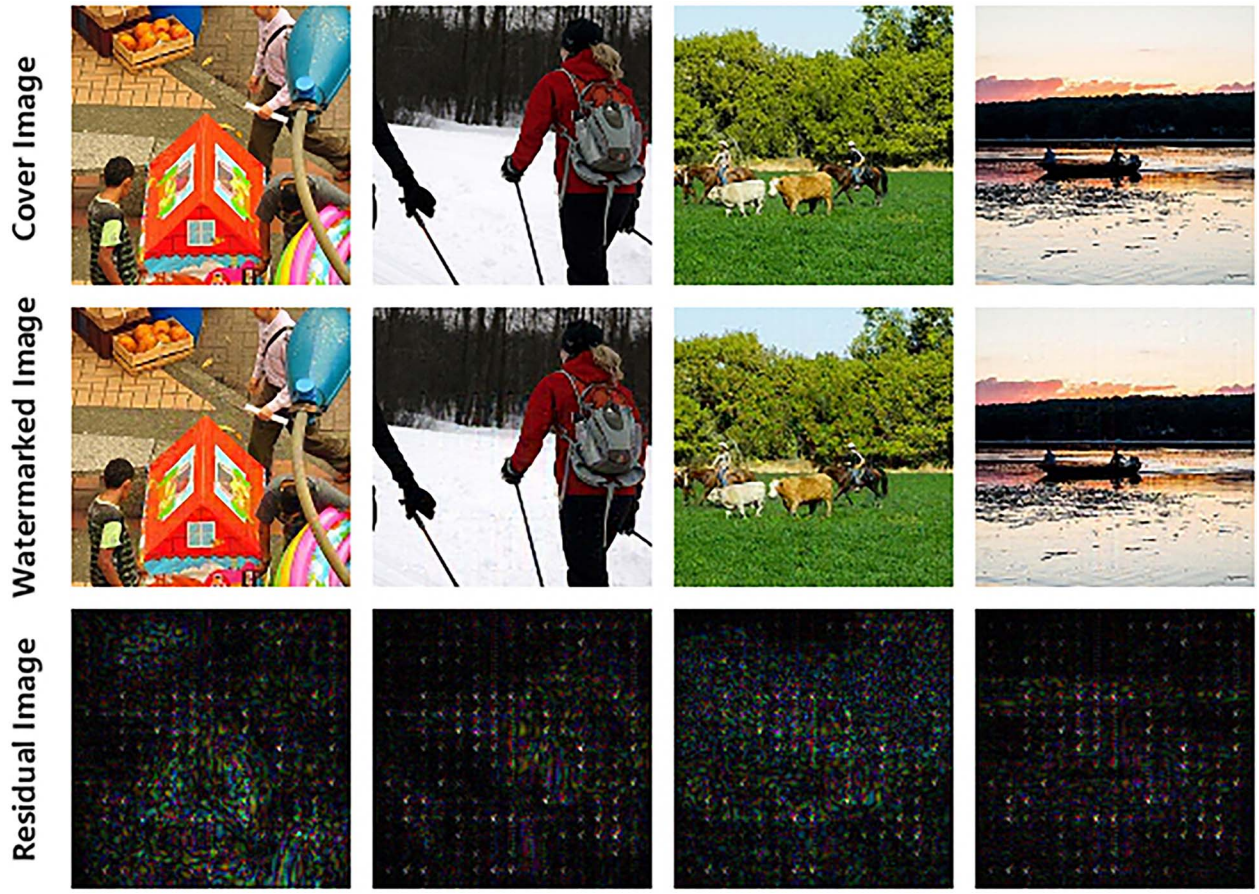

Figure 6. Visual quality of the watermarked images. Top: the cover image I_{co}. Second row: the watermarked image I_{en}. Bottom: the magnified residual $|I_{co} - I_{en}|$ between cover image and watermarked image.

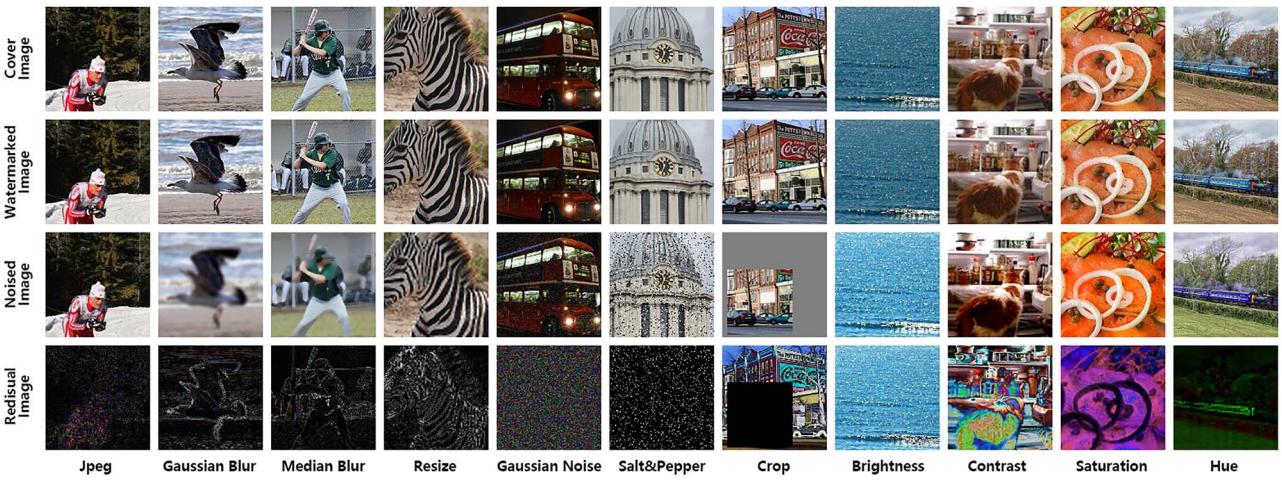

Figure 7. Visual results of the noised images with different attacks. Top: the cover image I_{co}. Second row: the watermarked image I_{en}. Third row: the attacked image I_{no}. Bottom: the magnified residual $|I_{en} - I_{no}|$ between watermarked image and attacked image.

increasing under the brightening distortion, which shows that the image loses more information after brightening than dimming.

4.3.2. *Robustness against Combined Attacks*

It is an ideal situation that watermarked images are subject to only one type of attack. In real-world scenarios, images are usually propagated several times, which means that images are always subject to more than one attack. In this section, we mainly discuss the robustness of the proposed method against combined attacks.

To test the application scenarios, we design four groups of combined attacks by combining different types of individual attacks including compression, crop, noise, blur and brightness. Table 5 and Table 6 show the BER and FCR of each method under different combined attacks and intensities, respectively. Under high-intensity combined attacks, our model maintains an extremely low BER which is lower than 1%. This is because our model has strong and balanced robustness against various attacks. In particular, for combined attacks involving JPEG compression (QF=50),

Table 2. The BER (%) values of each method against various attacks. The best and second best result are highlighted in bold and underlined, respectively.

Attack	Factor	Methods				
		HiDDeN [11]	MBRS [16]	SSL [18]	CIN [19]	Ours
JPEG	$QF = 50$	44.61	6.38	27.39	0.96	**0.06**
Gaussian Blur	$\sigma = 2$	26.41	0.08	5.31	**0**	0.01
Median Blur	$k = 5$	29.89	0.46	25.37	30	**0.33**
Resize	$p = 50\%$	23.54	**0**	27.72	**0**	**0**
Gaussian Noise	$\sigma = 0.15$	49.46	0.02	38.99	0.01	**0**
Salt & Pepper	$p = 10\%$	50.04	**0**	44.89	**0**	**0**
Crop	$p = 40\%$	21.69	**0**	34.08	**0**	**0**
Cropout	$p = 50\%$	21.33	**0**	25.58	**0**	**0**
Dropout	$p = 50\%$	22.39	0.0007	19.89	**0**	**0**
Brightness	$f = 1.5$	21.84	0.04	11.97	**0.02**	0.13
Contrast	$f = 1.5$	21.55	**0**	10.91	**0**	0.0007
Saturation	$f = 1.5$	21.01	**0**	1.53	**0**	**0**
Hue	$f = 0.1$	26.26	**0**	4.64	**0**	**0**

Table 3. The FCR values of each method against various attacks. The best and second best result are highlighted in bold and underlined, respectively.

Attack	Factor	Methods				
		HiDDeN [11]	MBRS [16]	SSL [18]	CIN [19]	Ours
JPEG	$QF = 50$	0	0.3158	0.0008	0.8144	**0.9826**
Gaussian Blur	$\sigma = 2$	0	0.9796	0.3602	**1**	0.9976
Median Blur	$k = 5$	0	0.8938	0.0040	0.02	**0.9098**
Resize	$p = 50\%$	0	**1**	0.0010	**1**	**1**
Gaussian Noise	$\sigma = 0.15$	0	0.9944	0	0.9992	**1**
Salt & Pepper	$p = 10\%$	0	**1**	0	**1**	**1**
Crop	$p = 40\%$	0.0002	**1**	0	**1**	**1**
Cropout	$p = 50\%$	0.0002	**1**	0.0010	**1**	**1**
Dropout	$p = 50\%$	0	0.9998	0.0084	**1**	**1**
Brightness	$f = 1.5$	0.0006	0.9958	0.2044	**0.9986**	0.9890
Contrast	$f = 1.5$	0	**1**	0.2068	**1**	0.9998
Saturation	$f = 1.5$	0.0002	**1**	0.7410	**1**	**1**
Hue	$f = 0.1$	0	**1**	0.3622	**1**	**1**

Table 4. The BER (%) values of each method under different attacks and intensities. The best and second best result are highlighted in bold and underlined, respectively.

Attack	Factor	Methods				
		HiDDeN [11]	MBRS [16]	SSL [18]	CIN [19]	Ours
JPEG	$QF = 80$	40.89	1.18	12.81	0.27	**0.0013**
	$QF = 70$	42.33	2.95	18.83	0.47	**0.01**
	$QF = 60$	43.53	4.71	23.74	0.68	**0.04**
Gaussian Blur	$\sigma = 1.0$	21.37	**0**	2.47	**0**	**0**
	$\sigma = 1.5$	23.55	0.0001	3.32	**0**	**0**
	$\sigma = 2.5$	28.42	0.92	8.11	**0.0013**	0.21
Gaussian Noise	$\sigma = 0.05$	31.17	**0**	17.32	**0**	**0**
	$\sigma = 0.10$	47.85	**0**	31.15	**0**	**0**
	$\sigma = 0.20$	49.75	0.36	43.29	0.04	**0.03**
Brightness	$f = 0.50$	20.79	**0**	3.96	**0**	**0**
	$f = 0.75$	20.75	**0**	1.11	**0**	**0**
	$f = 1.25$	21.17	0.01	5.25	**0**	0.03

compared with CIN [19], the BER of the proposed method is at least 9% lower. In addition, it can be seen that when the combined attack strength increases, the BER of our model does not increase significantly, and remains close to 0%. Besides, Table 6 also indicates the superior performance of the proposed method compare with other methods.

Table 5. The BER (%) values of each method under different combined attacks and intensities. JPEG + Crop denotes that image is attacked by Crop after JPEG compression. The two factors are used as the parameters of the two attacks, respectively. The best and second best result are highlighted in bold and underlined, respectively.

Attack	Factor	Methods				
		HiDDeN [11]	MBRS [16]	SSL [18]	CIN [19]	Ours
JPEG + Crop	$QF = 80,\ p = 80\%$	41.38	1.51	30.61	1.17	**0.0013**
	$QF = 70,\ p = 70\%$	43.65	4.37	35.83	3.32	**0.05**
	$QF = 60,\ p = 60\%$	45.24	7.98	40.18	5.88	**0.24**
	$QF = 50,\ p = 50\%$	46.67	12.86	43.62	10	**0.75**
JPEG + Gaussian Noise	$QF = 80,\ \sigma = 0.04$	44.59	7.12	23.55	8.93	**0.0039**
	$QF = 70,\ \sigma = 0.06$	47.53	20.23	30.22	13.73	**0.06**
	$QF = 60,\ \sigma = 0.08$	48.78	29.97	35.11	19.88	**0.26**
	$QF = 50,\ \sigma = 0.10$	49.27	36.54	38.56	26.07	**0.79**
Gaussian Blur + Crop	$\sigma = 1.2,\ p = 80\%$	22.11	0.0021	30.55	**0**	**0**
	$\sigma = 1.3,\ p = 70\%$	22.89	0.0033	34.77	0.01	**0**
	$\sigma = 1.4,\ p = 60\%$	24.23	0.02	39.18	0.06	**0.0013**
	$\sigma = 1.5,\ p = 50\%$	26.22	0.18	43.15	0.33	**0.04**
Gaussian Blur + Brightness	$\sigma = 1.2,\ f = 1.2$	22.28	**0.05**	7.55	0.09	0.06
	$\sigma = 1.3,\ f = 1.3$	23.15	0.17	11.21	0.27	**0.12**
	$\sigma = 1.4,\ f = 1.4$	24.29	0.42	14.92	0.72	**0.28**
	$\sigma = 1.5,\ f = 1.5$	25.38	0.77	18.65	1.35	**0.49**

Table 6. The FCR values of each method under different combined attacks and intensities. The best and second best result are highlighted in bold and underlined, respectively.

Attack	Factor	Methods				
		HiDDeN [11]	MBRS [16]	SSL [18]	CIN [19]	Ours
JPEG + Crop	$QF = 80,\ p = 80\%$	0	0.7342	0	0.8618	**0.9996**
	$QF = 70,\ p = 70\%$	0	0.4614	0	0.6952	**0.9862**
	$QF = 60,\ p = 60\%$	0	0.2528	0	0.4828	**0.9386**
	$QF = 50,\ p = 50\%$	0	0.0982	0	0.2370	**0.8290**
JPEG + Gaussian Noise	$QF = 80,\ \sigma = 0.04$	0	0.1862	0.0076	0.6846	**0.9988**
	$QF = 70,\ \sigma = 0.06$	0	0.0054	0.0004	0.4268	**0.9846**
	$QF = 60,\ \sigma = 0.08$	0	0.0002	0	0.1474	**0.9344**
	$QF = 50,\ \sigma = 0.10$	0	0	0	0.0322	**0.8178**
Gaussian Blur + Crop	$\sigma = 1.2,\ p = 80\%$	0.0004	0.9996	0	0.9998	**1**
	$\sigma = 1.3,\ p = 70\%$	0	0.9990	0	0.9984	**1**
	$\sigma = 1.4,\ p = 60\%$	0	0.9946	0	0.9824	**0.9996**
	$\sigma = 1.5,\ p = 50\%$	0	0.9560	0	0.9004	**0.9890**
Gaussian Blur + Brightness	$\sigma = 1.2,\ f = 1.2$	0.0004	**0.9968**	0.3276	0.9884	0.9952
	$\sigma = 1.3,\ f = 1.3$	0	0.9854	0.2012	0.9622	**0.9890**
	$\sigma = 1.4,\ f = 1.4$	0	0.9646	0.1266	0.9150	**0.9744**
	$\sigma = 1.5,\ f = 1.5$	0	0.9350	0.0754	0.8524	**0.9542**

4.3.3. *Robustness against Social Platform Processing*

Image is one of the most shared content in social platform. During the transmission of various social media, images are often subject to unknown compression. Thus, in this section, we mainly discuss the effectiveness of the proposed scheme against social platform processing. We conduct experiments on two different types of social platforms, Twitter and WeChat. Specifically, we use the tweet function of Twitter and the non-original image transmission of WeChat to obtain distorted images. We randomly select 100 images from the validation set for testing and use a fixed watermark message for these images.

Tables 7 and 8 report the comparison results on the social platform processing. Even under the process of unknown compression, the BER and FCR of our model are still close to 0% and 100%, respectively, which shows that our method achieves strong robustness against social media transmission evidently. In addition, we can see that the image compression intensity of WeChat is stronger than Twitter, and the BER of other methods has a large increase, but our model can still maintain a low BER.

The performance mainly benefits from the combined noise layer and the flow-based encoder. The combined noise layer which contains real JPEG and JPEG-Mask makes the model achieve strong robustness to JPEG compression and thus resist social platform processing. Besides, multiple coupling operations in the flow-based encoder not only embed the watermark into the deep feature of the cover image, but also greatly increase the redundancy of the watermark, resulting in the improvement of the robustness against compression. It is proved that improving the robustness against JPEG compression is of practical significance to image watermarking.

Table 7. The BER (%) values of each method against social platform processing. The best and second best result are highlighted in bold and underlined, respectively.

Social platforms	Methods				
	HiDDeN [11]	MBRS [16]	SSL [18]	CIN [19]	Ours
Twitter	25.44	<u>0.17</u>	15.37	10.27	**0**
WeChat	28.27	<u>7.03</u>	27.03	20.31	**0.1**

Table 8. The FCR values of each method against social platform processing. The best and second best result are highlighted in bold and underlined, respectively.

Social platforms	Methods				
	HiDDeN [11]	MBRS [16]	SSL [18]	CIN [19]	Ours
Twitter	0	<u>0.95</u>	0.01	0.13	**1**
WeChat	0	<u>0.17</u>	0	0.04	**0.97**

Table 9. Model ablation study. Robustness and imperceptibility are measured by BER (%) and PSNR (dB), respectively. The best and second best result are highlighted in bold and underlined, respectively.

Modules			Robustness			Imperceptibility
Flow-based Encoder	Conv Decoder	FC Decoder	JPEG ($QF = 50$)	Gaussian Blur ($\sigma = 2$)	Crop ($p = 40\%$)	
	✓	✓	<u>0.07</u>	0.16	0.45	<u>35.68</u>
✓	✓		0.09	**0**	<u>0.003</u>	34.99
✓		✓	5.09	5.03	7.68	34.34
✓	✓	✓	**0.06**	<u>0.01</u>	**0**	**35.79**

4.4. Ablation Study

Since we propose flow-based encoder and dual-decoder for better performance, in this section, we conduct ablation experiments to verify their effectiveness.

Table 9 represents the results of model ablation experiments. It can be observed that absence of any part in SEDD results in the decreasing of robustness and imperceptibility. Comparing the top and bottom row, we can see that flow-based encoder model outperforms the common encoder model. Especially for crop, the BER is 0.45% lower. It shows that with flow-based encoder, the watermark information in the watermarked image is more redundant. Besides, training with dual-decoder gets better performance than training with any single decoder. Although the second row in Table 9 shows that the convolution decoder-based model also has strong robustness, its imperceptibility is unsatisfactory. Figure 8 compares the visual quality of watermarked images generated by the convolution decoder model and the dual decoder model. From the locally enlarged image block, it can be noted that training with convolution decoder will lead to the corner artifacts, that is, the watermark information will gather at the corners of the watermark block to form prominent color or dot traces. With dual-decoder, the watermark information is more evenly distributed in the image, which mitigates the corner artifacts in the watermarked images.

5. CONCLUSIONS

In this paper, we propose a novel robust image watermarking framework named SEDD, which effectively enhances the robustness while ensuring high imperceptibility. To better fuse the watermark and cover image, we utilize the flow-based model to couple them and realize watermark embedding. In addition, we propose a parallel dual-decoder to mitigate the corner artifacts and improve the visual quality of the watermarked image. Extensive experiment shows that our method achieves stronger robustness in not only most individual attacks but also combined attacks and social platform processing. However, the corner artifacts on watermarked images have not been completely eliminated. In the future, our main research will focus on minimizing the influence of corner artifacts on visual quality while ensuring robustness.

DATA AVAILABILITY

The data underlying this article will be shared on reasonable request to the first and corresponding author.

ACKNOWLEDGEMENT

This work was supported by the National Natural Science Foundation of China (NSFC) under Grants 62272331 and 61972269, Sichuan Science and Technology Program under Grant 2022YFG0320, the Key Laboratory of Data Protection and Intelligent Management, Ministry of Education, Sichuan University and the Fundamental Research Funds for the Central Universities under Grant SCU2023D008.

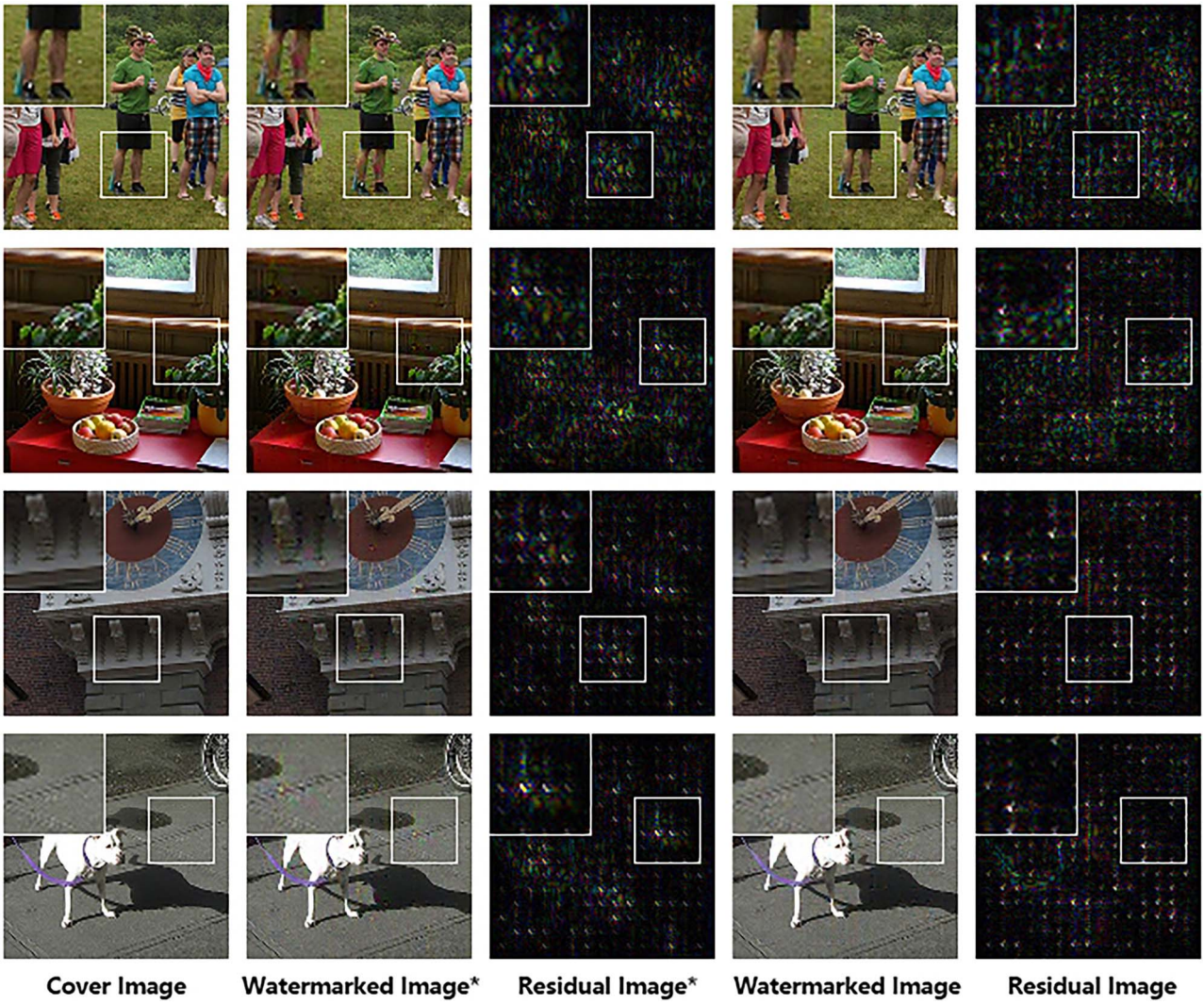

Figure 8. Visual comparison of the watermarked image between convolution decoder model and dual-decoder model. The residual image is the magnified difference between cover image and watermarked image. The symbol '$*$' denotes images generated by the convolution decoder-based model.

REFERENCES

1. Hsu, C.-T. and Wu, J.-L. (1999) Hidden digital watermarks in images. *IEEE Trans. Image Process.*, **8**, 58–68.
2. Bi, N., Sun, Q., Huang, D., Yang, Z. and Huang, J. (2007) Robust image watermarking based on multiband wavelets and empirical mode decomposition. *IEEE Trans. Image Process.*, **16**, 1956–1966.
3. Basso, A., Cavagnino, D., Pomponiu, V. and Vernone, A. (2011) Blind watermarking of color images using Karhunen Loève transform keying. *The Computer Journal*, **54**, 1076–1090.
4. Urvoy, M., Goudia, D. and Autrusseau, F. (2014) Perceptual DFT watermarking with improved detection and robustness to geometrical distortions. *IEEE Trans. Inf. Forensics Secur.*, **9**, 1108–1119.
5. Ko, H.-J., Huang, C.-T., Horng, G. and Wang, S.-J. (2020) Robust and blind image watermarking in DCT domain using inter-block coefficient correlation. *Inform. Sci.*, **517**, 128–147.
6. Huan, W., Li, S., Qian, Z. and Zhang, X. (2022) Exploring stable coefficients on joint sub-bands for robust video watermarking in DT CWT domain. *IEEE Trans. Circuits Syst. Video Technol.*, **32**, 1955–1965.
7. Huang, Y., Guan, H., Liu, J., Zhang, S., Niu, B. and Zhang, G. (2023) Robust texture-aware local adaptive image watermarking with perceptual guarantee. *IEEE Trans. Circuits Syst. Video Technol.*, **33**, 4660–4674.
8. Hosny, K. M. and Darwish, M. M. (2019) Resilient color image watermarking using accurate quaternion radial substituted Chebyshev moments. *ACM Transactions on Multimedia Computing, Communications, and Applications*, **15**, 46:1–46:1, 25.
9. Wang, C., Wang, X., Xia, Z., Ma, B. and Shi, Y.-Q. (2020) Image description with polar harmonic Fourier moments. *IEEE Trans. Circuits Syst. Video Technol.*, **30**, 4440–4452.
10. Hu, R. and Xiang, S. (2021) Cover-lossless robust image watermarking against geometric deformations. *IEEE Trans. Image Process.*, **30**, 318–331.
11. Zhu, J., Kaplan, R., Johnson, J., and Fei-Fei, L. (2018) HiDDeN: hiding data with deep networks. *Proceedings of the European conference on computer vision (ECCV), Munich, GERMANY, 08-14 September*, pp. 657–672. Springer, Cham.
12. Ahmadi, M., Norouzi, A., Karimi, N., Samavi, S. and Emami, A. (2020) ReDMark: framework for residual diffusion watermarking based on deep networks. *Expert Systems with Applications*, **146**, 113157.

13. Zhang, C., Benz, P., Karjauv, A., Sun, G., and Kweon, I. S. (2020) UDH: universal deep hiding for steganography, watermarking, and light field messaging. *Advances in neural information processing systems, virtual, 06-12 December*, pp. 10223–10234. Curran associates Inc., New York, NY.
14. Zhong, X., Huang, P.-C., Mastorakis, S. and Shih, F.Y. (2021) An automated and robust image watermarking scheme based on deep neural networks. *IEEE Trans. Multimed.*, **23**, 1951–1961.
15. Liu, Y., Guo, M., Zhang, J., Zhu, Y., and Xie, X. (2019) A novel two-stage separable deep learning framework for practical blind watermarking. *Proceedings of the 27th ACM international conference on multimedia, Nice, FRANCE, 21-25 October*, pp. 1509–1517. Association for Computing Machinery, New York, NY, USA.
16. Jia, Z., Fang, H., and Zhang, W. (2021) MBRS: enhancing robustness of DNN-based watermarking by Mini-batch of real and simulated JPEG compression. *Proceedings of the 29th ACM international conference on multimedia, virtual event, China, 20-24 October*, pp. 41–49. Association for Computing Machinery, New York, NY, USA.
17. Luo, X., Zhan, R., Chang, H., Yang, F., and Milanfar, P. (2020) Distortion agnostic deep watermarking. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, Seattle, WA, USA, 13-19 June*, pp. 13548–13557. IEEE computer society, los Alamitos, CA, USA.
18. Fernandez, P., Sablayrolles, A., Furon, T., Jégou, H., and Douze, M. (2022) Watermarking images in self-supervised latent spaces. *IEEE international conference on acoustics, speech and signal processing (ICASSP), Singapore, Singapore, 23-27 may*, pp. 3054–3058. IEEE, Piscataway, NJ.
19. Ma, R., Guo, M., Hou, Y., Yang, F., Li, Y., Jia, H., and Xie, X. (2022) Towards blind watermarking: combining invertible and non-invertible mechanisms. *Proceedings of the 30th ACM international conference on multimedia, Lisbon, Portugal, 10-14 October*, pp. 1532–1542. Association for Computing Machinery, New York, NY, USA.
20. Fang, H., Jia, Z., Qiu, Y., Zhang, J., Zhang, W. and Chang, E.-C. (2023) De-END: decoder-driven watermarking Network. *IEEE Trans. Multimed.*, **25**, 7571–7581.
21. Fang, H., Jia, Z., Ma, Z., Chang, E.-C., and Zhang, W. (2022) PIMoG: an effective screen-shooting noise-layer simulation for deep-learning-based watermarking Network. *Proceedings of the 30th ACM international conference on multimedia, Lisbon, Portugal, 10-14 October*, pp. 2267–2275. Association for Computing Machinery, New York, NY, USA.
22. Wengrowski, E. and Dana, K. (2019) Light field messaging with deep photographic steganography. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, Long Beach, CA, 16-20 June*, pp. 1515–1524. IEEE computer society, los Alamitos, CA, USA.
23. Tancik, M., Mildenhall, B., and Ng, R. (2020) StegaStamp: invisible hyperlinks in physical photographs. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, ELECTR NETWORK, 14-19 June*, pp. 2117–2126. IEEE computer society, los Alamitos, CA, USA.
24. Mallika, U. and J. S., and Aggarwal, A. K. (2022) Neural style transfer for image within images and conditional GANs for destylization. *Journal of Visual Communication and Image Representation*, **85**, 103483.
25. Garg, M., Ubhi, J.S. and Aggarwal, A.K. (2023) Neural style transfer for image steganography and destylization with supervised image to image translation. *Multimed. Tools Appl.*, **82**, 6271–6288.
26. Liang, J., Lugmayr, A., Zhang, K., Danelljan, M., Van Gool, L., and Timofte, R. (2021) Hierarchical conditional flow: A unified framework for image super-resolution and image rescaling. *Proceedings of the IEEE/CVF international conference on computer vision, ELECTR NETWORK, 11-17 October*, pp. 4076–4085. IEEE computer society, los Alamitos, CA, USA.
27. Xiao, M., Zheng, S., Liu, C., Wang, Y., He, D., Ke, G., Bian, J., Lin, Z., and Liu, T.-Y. (2020) Invertible image rescaling. *Proceedings of the European conference on computer vision (ECCV), Glasgow, UK, 23-28 august*, pp. 126–144. Springer, Cham.
28. Xie, Y., Cheng, K. L., and Chen, Q. (2021) Enhanced invertible encoding for learned image compression. *Proceedings of the 29th ACM international conference on multimedia, virtual event, China, 20-24 October*, pp. 162–170. Association for Computing Machinery, New York, NY, USA.
29. Dinh, L., Krueger, D., and Bengio, Y. (2015) NICE: non-linear independent components estimation. *Proceedings of the 3rd international conference on learning representations (ICLR workshop)*, San Diego, CA, USA, 7-9 may, pp. 1–13. OpenReview.net.
30. Rezende, D. and Mohamed, S. (2015) Variational inference with normalizing flows. *Proceedings of the 32nd international conference on machine learning, Lille, France, 07-09 July*, pp. 1530–1538. PMLR, New York.
31. Dinh, L., Sohl-Dickstein, J., and Bengio, S. (2017) Density estimation using real NVP. *5th international conference on learning representations, conference track proceedings, Toulon, France, 24-26 April*, pp. 1–32. OpenReview.net.
32. Kingma, D. P. and Dhariwal, P. (2018) Glow: generative flow with invertible 1x1 convolutions. *Advances in neural information processing systems, Montréal, Canada, 03-08 December*, pp. 10236–10245. Curran associates, Inc., New York, NY.
33. Lu, S.-P., Wang, R., Zhong, T., and Rosin, P. L. (2021) Large-capacity image steganography based on invertible neural networks. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, ELECTR NETWORK, 19-25 June*, pp. 10816–10825. IEEE computer society, los Alamitos, CA, USA.
34. Jing, J., Deng, X., Xu, M., Wang, J., and Guan, Z. (2021) HiNet: deep image hiding by invertible Network. *Proceedings of the IEEE/CVF international conference on computer vision, ELECTR NETWORK, 11-17 October*, pp. 4733–4742. IEEE computer society, los Alamitos, CA, USA.
35. Xu, Y., Mou, C., Hu, Y., Xie, J., and Zhang, J. (2022) Robust invertible image steganography. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, New Orleans, LA, 18-24 June*, pp. 7875–7884. IEEE computer society, los Alamitos, CA, USA.
36. Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C., Qiao, Y., and Change Loy, C. (2018) ESRGAN: enhanced super-resolution generative adversarial networks. *Proceedings of the European conference on computer vision (ECCV) workshops, Munich, GERMANY, 08-14 September*, pp. 63–79. Springer, Cham.
37. Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014) Microsoft COCO: common objects in context. *Proceedings of the European conference on computer vision (ECCV), Zurich, SWITZERLAND, 06-12 September*, pp. 740–755. Springer, Cham.
38. Collobert, R., Kavukcuoglu, K., and Farabet, C. (2011) Torch7: A Matlab-like environment for machine learning. *BigLearn, NIPS workshop, Granada, Spain, January*, pp. 1–6. Curran associates Inc., New York, NY.
39. Kingma, D. P. and Ba, J. (2015) Adam: A method for stochastic optimization. *Proceedings of the 3rd international conference on learn-*

ing representations (ICLR), San Diego, CA, USA, 7-9 may, pp. 1–15. OpenReview.net.

40. Almohammad, A. and Ghinea, G. (2010) Stego image quality and the reliability of PSNR. *2010 2nd international conference on image processing theory, tools and applications, Paris, France, 07-10 July*, pp. 215–220. IEEE, Piscataway, NJ.
41. Wang, Z., Bovik, A., Sheikh, H. and Simoncelli, E. (2004) Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.*, **13**, 600–612.