IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 20 October 2023, accepted 23 November 2023, date of publication 30 November 2023,
date of current version 8 December 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3337812

RESEARCH ARTICLE

Enhanced Attention Mechanism-Based Image Watermarking With Simulated JPEG Compression

JIA-YI ZHONG[1], DUO-WEN PAN[2], JING-JIE WANG[3], AND XUAN-BO JIA[4]
[1]School of Information Science and Technology, Northwest University, Xi'an 710127, China
[2]International School, Beijing University of Posts and Telecommunications, Beijing 100876, China
[3]School of Mathematics and Statistics, Southwest University, Chongqing 400700, China
[4]College of Excellence, Hangzhou Dianzi University, Hangzhou 310018, China

Corresponding author: Xuan-Bo Jia (littleangelet@outlook.com)

ABSTRACT Deep learning-based image watermarking algorithms have been widely studied as an important technology for copyright protection. These methods utilize an end-to-end architecture with an encoder, a noise layer and a decoder to make the watermark robust to various distortions. However, recent algorithms present unsatisfactory visual quality and robustness against JPEG compression, which is the most common image processing operation but is non-differential thus cannot be directly included in the noise layer. To address this limitation, this study proposes a novel enhanced attention-based image watermarking algorithm with simulated JPEG compression, which leverages the channel and spatial attention mechanism to facilitate watermark embedding and simulates JPEG compression with a suitably designed function. Precisely, we design a differentiable rounding function based on the Fourier series to replace the quantization process in JPEG compression, which overcomes the non-differentiability of JPEG compression and can be incorporated in the training process. In addition, we propose an enhanced dual attention module in the encoder, which combines channel and spatial attention to improve the performance of our model. The channel attention guides the encoder to fuse the watermark into more important channels and the spatial attention further helps to embed the watermark into regions with more complex textures. The experimental results show that our method generates high quality watermarked images, with PSNR over 50 when no noise is applied. Compared with current methods, our model achieves stronger robustness to JPEG compression, with bit accuracy over 99% under the JPEG compression with quality factor of 50. Besides, the proposed framework also exhibits excellent robustness for a variety of common distortions, including cropout and dropout.

INDEX TERMS Robust watermarking, JPEG compression, attention mechanism, simulated rounding function, deep learning.

I. INTRODUCTION

With the development of communication technology, large amounts of images are spread over the networks, which makes the copyright protection become an increasingly important problem [1], [2], [3]. Digital watermarking [4] is a technique that embeds a marked message into multimedia files while maintaining the visual quality of the original file and providing robustness against various noise attacks, including random cropping, blurring, and JPEG (Joint Photographic Experts Group) compression. This feature has led to the widespread use of digital watermarking in the field of multimedia copyright protection and authentication by inserting author logotypes as evidence in situations of copyright disputes. The technique has been extensively applied in protecting the property rights of images [5], [6], videos [7], [8], and audio

In 1994, Schyndel et al. [11] first proposed the concept of digital watermarking and discussed the research on undetectable digital watermarking based on the Least Significant Bit algorithm (LSB) standard image coding. Classical digital image watermarking methods can be summarized as spatial watermarking and frequency watermarking. The spatial

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

watermarking algorithms mentioned above manipulate the least significant bits of some selected pixels, yet can be easily detected by statistics methods. Thus Jiansheng et al. turned to make an effort on embedding the watermarking in discrete cosine transform (DCT) coefficients, based on the principle that this technique can embed large amounts of bit data without causing perceptible defects, hence their method has better robustness and image visual quality. However, the method relies heavily on artificial shallow feature extraction, which limits the robustness of the algorithm.

Traditional methods typically use heuristics to decide how much to modify each pixel to embed the watermark into images. These heuristics are only effective in the domains for which they are designed, but they are static and are difficult to be robust to new distortions. For example, some algorithms are only designed to resist JPEG compression, but they might be vulnerable to cropping operations. Deep learning-based methods can learn a more robust algorithm without heuristics by introducing diverse noise layers in the model and setting a suitably designed loss function. Thus, this paper focuses on studying the deep learning based algorithm for image watermarking.

During recent decades, deep neural networks (DNN), especially convolutional neural networks (CNN) have been extensively used to provide an end-to-end model for the watermarking problem. Zhu et al. [12] proposed HiDDeN, a CNN-based framework for image watermarking. Currently, Jia et al. [13] proposed a method that uses a Mini-Batch of Simulated and Real JPEG compression (MBRS). One of the simulated JPEG, real JPEG compression, and a noise-free layer will be chosen randomly to be the distortion, thus the model can be trained for different scenarios. Tan et al. [14] proposed to introduce the frequency channel attention into digital watermarking, and built a two-branch structure to concentrate information from different frequency channels. While the researches above show great results, a significant issue cannot be ignored: those approaches fail to achieve satisfactory performance in the robustness against JPEG compression.

To address this limitation, this study proposes a new image watermarking method based on dual attention mechanism and simulated JPEG compression. Based on the previous work [15], [16] about channel attention combined with frequency analysis, we present a mechanism to add a dual attention block into the architecture, which combines channel attention [15] with spatial attention [17]. The dual attention block can be divided into two branches in a serial manner. One branch extracts the feature map from the image and explores the relationship between different channel feature maps. Each channel extracts the importance which should be truly care about and feeds them back to the model. The other branch generates a spatial attention map and uses the spatial feature to highlight the feature space location that should be focused on. The dual attention mechanism can extract more useful details and ignore the unimportant to withstand a variety of common noise attacks.

What's more, the standard quantization operation in JPGE compression rounds each DCT coefficient of floating-pint type into its nearest integer number, which is a non-differentiable stair-case function, as shown in Fig. 3 (orange line). We propose a differentiable function based on Fourier series (Equation (6)) to replace the rounding function, which acts as a smooth approximation of the stair-case function, as shown in Figure 3 (blue line).

In summary, the contributions of this paper are as follows:

1) This study proposes a novel enhanced attention-based image watermarking algorithm, which leverages the dual attention mechanism to facilitate watermark embedding. The dual attention mechanism consists of channel attention and spatial attention, which extracts more useful details and ignore the unimportant to withstand a variety of common noise attacks.
2) We design a differentiable rounding function based on the Fourier series to replace the quantization process in JPEG compression, which overcomes the non-differentiability of JPEG compression and can be incorporated in the training process, improving the robustness against JPEG compression under a variety of quality factors.
3) Experimental results show that the proposed method is superior to current watermarking schemes in image quality and robustness with various distortions, including JPEG compression, Cropout, and Dropout.

The remainder of the paper is organized as follows. In Section II, we review some related works about the DNN-based watermarking frameworks and methods to approximate JPEG. Section III provides the detailed description of our proposed method. The experimental results and the analysis are presented in Section IV. Section V summarizes the paper.

II. RELATED WORK

In this section, we introduce related work about deep learning-based digital watermarking, adversarial networks, and typical JPEG simulation methods used for JPEG-resistant image watermarking.

A. DEEP LEARNING FOR DIGITAL WATERMARKING

Due to the advantages of feature extraction ability of deep neural networks, many watermarking framework based on deep learning have been proposed. Zhu et al. [12] proposed an end-to-end DNN-based model for watermarking, which is mainly composed by an encoder, a decoder, and a discriminator used for more realistic visual effect. Ahmad et al. [18] proposed a framework that can support operations on DCT domain, which uses residual structure for encoder to control the strength of watermark patterns. Tancik et al. [19] paid close attention to print-shooting robustness. They achieved the robustness through stimulating the differential operation and applying them in the noise layer. Although these deep learning based watermarking frameworks facilitated both

encoder and decoder, their models are still inapplicable due to the differential limitation of the noise layer. Then a two-stage separable deep learning framework [20] is proposed, where the encoder and decoder are initialized without noise layer in stage one, and the decoder are enhanced alone in non-differential distortions in the stage two. However, it can not well deal with JPEG compression distortion.

B. ADVERSARY NETWORKS

The adversarial training was proposed by Goodfellow [21] in order to evaluate generative models. And many progresses are proposed to generate lots of variants of GAN. For example, DCGAN [22] and WGAN [23] are proposed to enhance the stability of training and quality of generated images and CycleGAN [24] and pix2pix [25] models are proposed for image to image translation. And CGAN [26] is proposed to add more conditions for image generating. Many models for watermarking use an adversary for the encoder to obtain higher image quality, and all of them obtain good results.

C. JPEG SIMULATION

Many methods which use differential operations to simulate the JPEG compression have been proposed in order to satisfy the need of one-stage end-to-end training. Zhu et al. [12] proposed JPEG-Mask, a method that zeros a fixed set of high frequency coefficient. The method only keeps the 3×3 low frequency region of U and V channels, the 5×5 low frequency region of Y channel. Through the method, the network can obtain the robustness against JPEG compression. What's more, other researches pay attention to the simulation of the quantization function in JPEG compression. Shin and Song [27] approximate the quantization step near zero.

III. PROPOSED METHOD

In this section, we first provide an overview of our proposed model. Then we introduce the detailed network architectures. Next, we describe the proposed dual attention module and the novel JPEG differentiable approximation method. Finally, we illustrate how to balance the image quality and imperceptibility with the strength factor.

A. MODEL OVERVIEW

As shown in Fig. 1, the architecture of our watermarking framework mainly includes five parts: message processor, encoder, decoder, noise layer, and adversary discriminator. The noise layer and adversary discriminator are only used in the training procedure to help the model gain robustness against specific noise and improve the image quality by adversarial training. In the inference stage, we only need message processor, encoder, and decoder. The message processor processes the message from a bit vector to a high dimensional feature map. The encoder embeds the message feature map into the original image, outputting the watermarked images. The noise layer includes noises that image might experience in transmission process. The decoder decodes the message from the noised image.

TABLE 1. The structure and the parameters of the proposed model.

Component	Structure
Message Process	Reshape [3 × 3 Conv+BN+ReLU] × 3 [Dual Attention Module] × 3
Encoder	3 × 3 Conv+BN+ReLU [Dual Attention Module] × 4 3 × 3 Conv+BN+ReLU Add with message feature map 3 × 3 Conv+BN+ReLU 1 × 1 Conv
Decoder	3 × 3 Conv+BN+ReLU [Dual Attention Module] × 4 3 × 3 Conv+BN+ReLU Reshape
Adversary	[3 × 3 Conv+BN+ReLU] × 4 Global Average Pooling

The discriminator discriminates whether an image contains watermarks, acting as an adversary. The detailed architecture of each component of the proposed model is shown in Table 1.

B. DETAILED NETWORK ARCHITECTURE

In the encoding process, messages should be handled in an appropriate way. To this end, we add a message processor (MP) to process the message and provide the processed feature map to the encoder. Firstly, MP receives a randomly generated binary key message M, which is composed of bits of L length, and reshapes the secret information M to $\{0, 1\}^{1\times h\times w}$, where $L = h \times w$. Then, it is amplified by 3×3 ConvBNReLU layers (convolution layer, batch normalization and normalization function), and then extended to $C \times H \times W$ through some transposed convolution layers with a stride $= 2$ (C is the number of feature channels, H and W are the length and width of the cover image respectively). Finally, in the step of expanding the message, the features of the message map are extracted by several shape-preserving convolutional block attention modules. After passing through each transposed convolutional layer, the width and height of the input tensor are twice that of the original tensor. Therefore, the length L of the secret message and the shape $H \times W$ of the cover image usually conform to the following relationship:

$$L = h \times w = \left(H/2^n\right) \times \left(W/2^n\right) \tag{1}$$

$n \in Z^*$ is an integer determined by L, H and W.

1) ENCODER

The encoder with parameters θ_E aims to encode the watermark into the host image with lower visual distortion effect, so as to minimize the distance between I_{en} and I_{co} and make them more visually similar. In order to make it easier for us to select channel features, we choose a mixed frequency channel attention block (which contains several convolutional block attention modules. [28]). First, we use a 3×3 ConvBNReLU layers to enlarge the cover image I_{co}, and then extract the image features of the same shape with the dual attention block, and then map it through

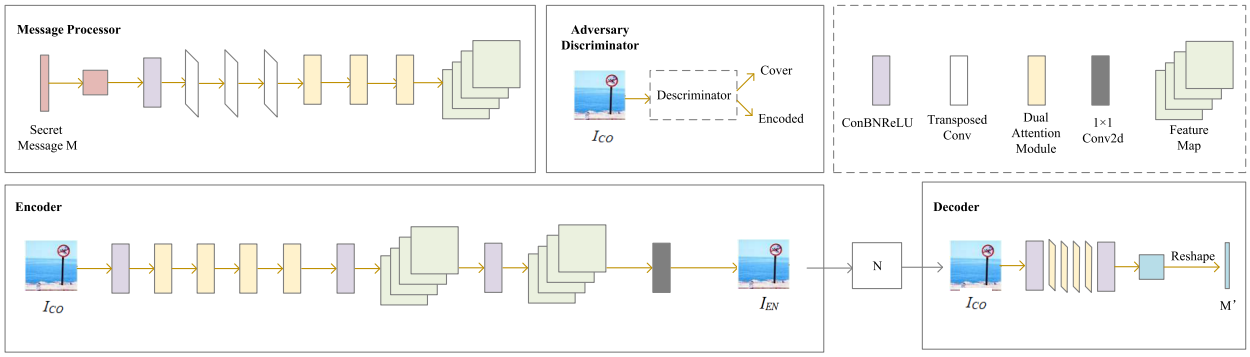

FIGURE 1. Model overview. The encoder embeds secret messages into the entire cover image. The message processor implements message expansion and redundancy by transposing the convolutional layer. The noise layer provides robustness to distortions such as JPEG, Cropout, and Dropout. Because the real JPEG compression is not differentiable, we propose a differentiable JPEG simulator. The decoder extracts the secret information from the encoded image, and another adversary discriminator distinguishes the cover image from the encoded image.

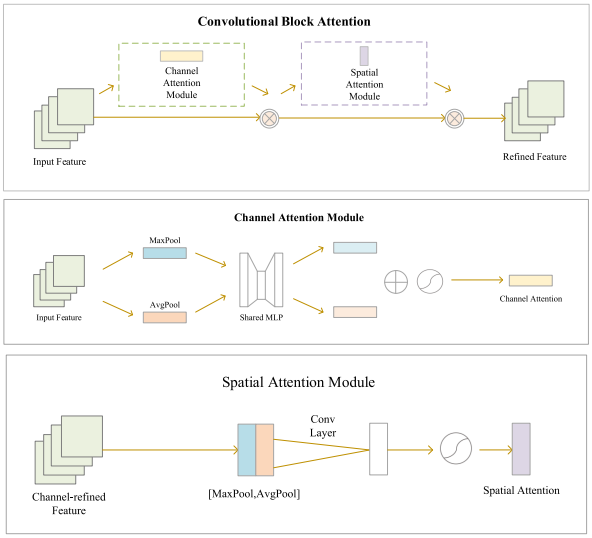

FIGURE 2. The overview of dual attention module. The module has two sequential sub-modules: the channel attention module and the spatial attention module. The intermediate feature map is adaptively refined through our module at every convolutional block of deep networks.

the 3 × 3 ConvBNReLU layer again. The tensor is then concatenated with the message feature map, and fed to a 3 × 3 ConvBNReLU layer. The tensor obtained by this step and the cover image are concentrated in a new tensor, which is fed into an 1 × 1 convolution layer. Finally, the encoded image I_{en} is obtained. The object of encoder training is to minimize:

$$L_{E1} = \text{MSE}\left(I_{co}, I_{en}\right) = \text{MSE}\left(I_{co}, E\left(\theta_E, I_{co}, M\right)\right) \quad (2)$$

2) NOISE LAYER

The noise layer provides robustness for the entire model. We select different noises from the specified noise pool as the noise layer, including the Identity, Dropout, JPEG and Cropout layers. The size of the input and output noise images is the same, and all types of noise need to participate in the end-to-end model training process [29]. Identity layer is the most simple, keeping I_{en} unchanged. The Dropout and Cropout layers cancel some changes made by the encoder, and generate a noisy image by combining the pixels in the cover image I_{co} and the encoded image I_{en}. Both types of noise preserve the pixel percentage p in I_{en} and use the remaining pixels in I_{co}, but Dropout selects each pixel independently, while Cropout preserves the random square clipping in I_{en}. The JPEG layer uses a quality factor of $Q \in (0, 100)$ to I_{en}. We also we propose a differentiable JPEG simulator which simulates color space transformation, DCT, and quantization steps in JPEG compression. And this is one of our contributions to this article.

Please note that all non-identity noise layers have a scalar hyperparameter to control the intensity of distortion : Dropout and Cropout retain a small part of pixels p; JPEG has a quality factor Q [12].

3) DECODER

The decoder with parameters θ_D determines the ability of the entire model to extract watermarks. In the decoding process, we input the noise image I_{no} to the 3 × 3 ConvBNReLU layer to enlarge it, and then reduce the sample through several convolutional block attention modules (CBAM) [14] and convert it to $C \times h \times w$. Finally, we transform the previously obtained multi-channel tensor into a single-channel tensor through a 3 × 3 convolution layer, and reshape it to obtain the decoding information M_D. The object of decoder training is to minimize:

$$L_D = \text{MSE}\left(M, M_D\right) = \text{MSE}\left(M, D\left(\theta_D, I_{no}\right)\right) \quad (3)$$

4) ADVERSARY

The adversary discriminator with parameters θ_A consists of several 3 × 3 convolution layers and a global average pooling layer. During the training process, under the influence of the adversarial network, the encoder will deceive the discriminator as much as possible to prevent it from distinguishing the encoded image and making the correct judgment. The encoding visual quality is updated by θ_A and minimize the update parameters L_{E2} to minimize the loss of classification

L_A. The loss of the adversary discriminator is:

$$L_A = \log(1 - A(\theta_A, E(\theta_E, I_{co}, M))) + \log(A(\theta_A, I_{co})) \quad (4)$$

The encoder parameters θ_E is updated by minimizing:

$$L_{E2} = \log(A(\theta_A, I_{en})) = \log(A(\theta_A, E(\theta_E, I_{co}, M))) \quad (5)$$

The total loss function is $L_A = \lambda_E L_{E1} + \lambda_D L_D + \lambda_A L_{E2}$ for the encoder and decoder, and loss L_A for the discriminator. $\lambda_E, \lambda_D, \lambda_A$ are hyper-parameters to balance the losses.

C. DUAL ATTENTION MODULE

Inspired by [30], we introduce a dual attention module in our architecture. As shown in Fig. 3, the dual attention module consists of two main components: the channel attention module and the spatial attention module. The channel attention module is responsible for capturing channel-wise attention, while the spatial attention module is responsible for capturing spatial attention.

In the channel attention module, we first use both global average pooling and max pooling to get different spatial context description. Then the features are sent to two fully connected layers to generate a set of channel attention maps, which weight the feature maps of each channel based on their importance for watermark embedding. In the spatial attention module, we use a convolutional layer followed by a sigmoid activation function to capture spatial attention. The output of the spatial attention module is a set of spatial attention maps, which weight the feature maps of each spatial location based on their importance. The dual attention module combines the outputs of the two components using element-wise multiplication, resulting in a set of attention maps that capture both spatial and channel-wise attention. These attention maps are then used to weight the feature maps of each channel and spatial location, producing a more informative representation of the input image. By combining these two attentions, our model will learn more suitable embedding space to facilitate watermark embedding.

D. JPEG DIFFERENTIABLE APPROXIMATION

There are four steps in JPEG compression, which contains color space transformation, discrete cosine transform (DCT), quantization, and encoding. During the process, since the conversion of colour space, each 8×8 image block is represented as three 8×8 matrices representing Y, U, and V components. Then DCT makes the data clearly divided into two parts, DC component and AC component, which means each 8×8 image block becomes three 8×8 floating point number matrices. Then the DCT map is divided by quantization coefficient matrix and rounding to the integer type. Since all the floating point data remain to be integer type approximately, the JPEG compression algorithm is not differentiable. Therefore, it cannot be directly incorporated into the noise layer, which will make the gradient disappear. In order to train a module that can tolerate JPEG compression distortion and can be directly integrated into our end-to-end framework, we propose a differentiable JPEG simulator which simulates color space transformation, DCT, and quantization steps in JPEG compression. Specially, we design a differentiable rounding function base on the Fourier series, which can be defined as:

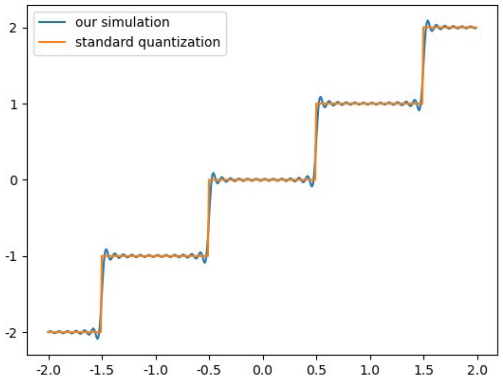

FIGURE 3. The quantization process of JPEG compression is simulated by the rounding function approximately.

$$Q(I) = I - \frac{1}{\pi} \sum_{k=1}^{K} \frac{(-1)^{k+1}}{k} \sin(2\pi kI) \quad (6)$$

where I is the input map after divided by quantization tables in JPEG compression, and K is used for maintaining a balance between approximation accuracy and computation efficiency. As K adds, the simulation function will be closer to the real round function, with the running time increase, too. We managed to make the value of K larger to get a better simulation, and empirically set K to 12. The rounding function is illustrated in Fig. 2. The orange curve refers to the original non-differentiable function, while the blue one is our simulated function.

E. STRENGTH FACTOR

Let $I_{diff} = I_{en} - I_{co}$ represents the residual image between the encoded image and the cover image. Strength factor is a factor multiplied to I_{diff} to obtain an adjusted encoded image. The role of strength factor is to balance the robustness and the imperceptibility during the *inference* stage. The adjusted encoded image is calculated as follows:

$$I_{en,S} = I_{co} + S \cdot I_{diff} \quad (7)$$

where I_{co} is the cover image, I_{diff} is the residual between the encoded image I_{en} and the cover image I_{co}, $I_{en,S}$ is the adjusted encoded with strength S.

Please note that we only use this adjustment operation in the inference stage. In training state, we do not adopt this operation, *i.e.*, we fix $S = 1$ to train our models. In other words, during the inference process, we multiply the residual between the encoder output image and the cover image by a factor, and add this scaled residual to the cover image to obtain the final watermark image. This factor is strength factor. We change S in the range from 0.2 to 2 to get

the best performance in PSNR (Peak Signal-to-Noise Ratio) and SSIM (Structural Similarity Index), with the highest bit accuracy in the inference process. Since our method is a blind watermarking algorithm, the trick is used only in the encoder not for decoder.

IV. EXPERIMENTS

In this section, we first describe the experimental setup, metrics used for evaluating our algorithm, and the baseline methods. Then we evaluate the image quality of the watermarked image produced by our model. Next, we make comparisons with previous methods to show the advantage of our algorithm. Then we present ablation study to show how each component in our method impacts the overall watermark performance.

A. EXPERIMENT SETUP, METRICS, AND BASELINE

1) EXPERIMENT SETUP

We use the MS COCO [31] dataset in the experiments. The COCO (Common Objects in Context) dataset is a widely used benchmark dataset for object detection, segmentation, and captioning tasks in computer vision. It was created to provide a large-scale and diverse dataset for training and evaluating models on various visual recognition tasks. The dataset consists of 328K images. We follow the same dataset setup as HiDDeN [12]. The original dataset is very large scale, so we randomly choose 11000 images from the dataset, among which 10000 images are used for training the model and 1000 images are used for testing. Unless otherwise specified, all images are resized to 128×128. Specifically, we download the 2017 training images from the official COCO website.[1] The we randomly choose 10000 images from it as the training set, 1000 images as the test set.

The strength factor is set as 1 during training. We choose $\lambda_E = 1$, $\lambda_D = 10$, and $\lambda_A = 0.0001$ for the weight factors in the loss function. Each model is trained for 100 epochs with a batchsize 12. The proposed algorithm takes about 5 hours for training and the inference time is 0.01 second for each image.

2) METRICS

- PSNR (Peak Signal-to-Noise Ratio). PSNR measures the similarity between I_{en} and I_{co}. Let the images I_{en} and I_{co} have two dimensions i and j, composed of c number of channels. The mean squad error (MSE) of two images is:

$$MSE = \frac{1}{c * i * j} \sum (I_{co} - I_{en})^2 \tag{8}$$

Then the PSNR is expressed as:

$$PSNR = 10 \cdot \log_{10} \left(\frac{MAX_I^2}{MSE} \right) \tag{9}$$

Here the MAX_I is the maximum valid value for a pixel. In case of the simple single byte image per pixel per channel this is 255. When two images are the same the MSE will give zero, resulting in an invalid divide by zero operation in the PSNR formula. In this case the PSNR is undefined and as we'll need to handle this case separately. The transition to a logarithmic scale is made because the pixel values have a very wide dynamic range. Typically result values are anywhere between 30 and 50 for compression, where higher is better. If the images significantly differ we'll get much lower ones like 15 and so. This similarity check is easy and fast to calculate, however in practice it may turn out somewhat inconsistent with human eye perception. The structural similarity algorithm aims to correct this.

- SSIM (Structural Similarity Index). SSIM is a perception-based model that considers image degradation as perceived change in structural information, while also incorporating important perceptual phenomena, including both luminance masking and contrast masking terms. The SSIM index is calculated on various windows of an image. The measure between two windows x and y of common size $N \times N$.

$$\text{SSIM}(x, y) = \frac{\left(2\mu_x\mu_y + c_1\right)\left(2\sigma_{xy} + c_2\right)}{\left(\mu_x^2 + \mu_y^2 + c_1\right)\left(\sigma_x^2 + \sigma_y^2 + c_2\right)} \tag{10}$$

where μ_x, μ_y are the pixel sample mean, σ_x^2, σ_y^2 are the pixel sample variance, σ_{xy} is the cross correlation, $c1, c2$ are two variables to stabilize the division with weak denominator. Higher SSIM represents higher image quality.

- Bit Accuracy. Bit accuracy is equal to the number of correctly decoded bits in the message divided by the total number of bits. The robustness is measured by the bit accuracy. Higher bit accuracy represents higher robustness.

3) BASELINE

Our baselines for comparison are HiDDeN [12] and MBRS [13]. In pursuit of the real results, we realize the baselines based on their open source of both codes and models.

- HiDDeN [12] is the first end-to-end trainable framework for data hiding. It can force the model to learn encodings that can survive noisy transmission by inserting noise layers between the encoder and decoder which apply different image transformations. It uses JPEG-Mask and JPEG-Drop to approximate non-differentiable JPEG compression.
- MBRS [13] utilizes mini-batch of real and simulated JPEG compression to enhance the JPEG robustness. Precisely, for different mini-batches, it randomly chooses one of real JPEG, simulated JPEG and noise-free layer as the noise layer.
- TS [20] adopts a two-stage separable deep learning framework, where the encoder and decoder are initialized without noise layer in stage one, and the decoder

[1] https://cocodataset.org/#download

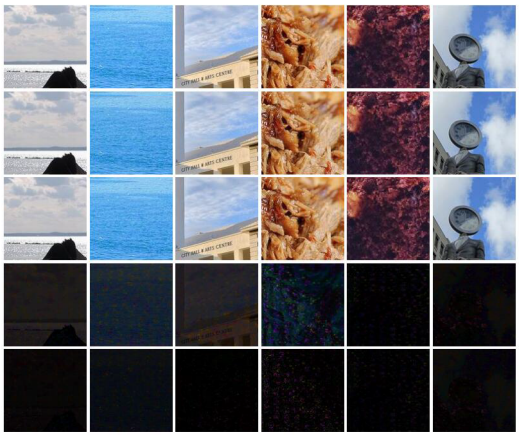

FIGURE 4. Six images randomly chosen from the test set to illustrate the image quality. We test them with the dropout noise layer. From top to bottom is: cover image, encoded image, noised image, R_1 and R_2.

TABLE 2. Comparison with previous methods against JPEG compression ($Q = 50$).

Model	HiDDeN [12]	TS [20]	MBRS [13]	Ours
Image size	128 × 128	128 × 128	128 × 128	128 × 128
Message length	30	30	64	30
Noise Layer	JPEG-Mask	JPEG	Mixed	Simulated JPEG
PSNR	31.6963	33.51	36.49	32.4048
SSIM	0.9544	-	0.9173	**0.9566**
Bit Accuracy	0.6076	0.7770	-	**0.9974**

are enhanced alone in non-differential distortions in the stage two.

We design a novel function based on the Fourier series to replace the quantization process in JPEG compression, which is incorporated into the noise layer to make our model more robust to JPEG compression. Besides, we propose an enhanced attention mechanism in the encoder, which combines channel and spatial attention to improve the performance of our model.

B. VISUAL QUALITY

To demonstrate the visual quality of the images generated by our model, we randomly selected some images from the test set and used the model trained under the dropout noise layer to generate encoded images. Some examples are shown in Fig. 4. In the figure, R_1 means the residual signal between the cover and encoded images, which can be represented as $R_1 = |I_{en} - I_{co}|$. R_2 is the residual signal between encoded and noised images, which can be represented as $R_2 = |I_{en} - I_{no}|$. Hence, We can observe that the proposed encoder can embed the message into the cover image while maintaining the high visual quality of the cover image.

C. COMPARISON WITH PREVIOUS METHODS

In this section, we compare our model with three state-of-the-art models [12], [13], [20]. Since the size of the input image and the length of the message are different for each method, we use the same image size and similar message length for a more fair comparison. More specifically, we use message length of $L = 30$ for $3 \times 128 \times 128$ images.

TABLE 3. Comparison with previous methods against other distortion.

Model	HiDDeN [12]	TS [20]	MBRS [13]	Ours
Image size	128 × 128	128 × 128	128 × 128	128 × 128
Message length	30	30	64	30
Noise Layer		Cropout (p=0.06)		
PSNR	32.8921	33.5	38.1182	**40.1767**
SSIM	0.967	-	0.9825	**0.9878**
Bit Accuracy	0.738	0.9730	0.968	**0.9874**

Model	HiDDeN [12]	TS [20]	MBRS [13]	Ours
Image size	128×128	128×128	128×128	128×128
Message length	30	30	64	30
Noise Layer		Dropout (p=0.8)		
PSNR	30.9728	33.5	38.8582	**41.9346**
SSIM	0.9606	-	0.9882	**0.9941**
Bit Accuracy	0.9674	0.9740	0.9999	**1**

Model	HiDDeN [12]	TS [20]	MBRS [13]	Ours
Image size	128×128	128×128	128×128	128×128
Message length	30	30	64	30
Noise Layer		Identity		
PSNR	35.9145	33.5	49.3006	**51.5677**
SSIM	0.9818	-	0.9985	**0.9991**
Bit Accuracy	1	1	1	**1**

1) JPEG COMPRESSION

In this part, we mainly show and discuss the robustness of our method against JPEG compression. In order to train a module that can resist JPEG compression distortion, we propose a differentiable JPEG simulator to simulate the color space transformation, DCT, and quantization steps in JPEG compression. The test model trains with the identity and JPEG (quality factor $Q = 50$) noise layer. As shown in Table 2, our method improves bit accuracy while maintaining higher image quality. In particular, our model achieves a bit accuracy higher than 99.74%, indicating that our model structure significantly improves image quality and bit accuracy.

Table 2 shows the different results of JPEG compression in different methods. We use the combination of simulated JPEG ($Q = 50$) and Identity noise layer to train our model. The bit accuracy is not reported in [13]. SSIM is not reported in [20].

The bit accuracy of HiDDeN is worst, which suggests JPEG-Mask and JPEG-Drop cannot well approximate JPEG compression. TS improves the bit accuracy by introducing a two-stage strategy, which only learns the robustness in the second stage. The bit accuracy of our model is higher than the other three baseline models, indicating better robustness against JPEG compression. It shows that the end-to-end one stage training is more effective than the two-stage method by designing a suitable differential function to approximate JPEG compression. We design such a function based on

TABLE 4. The image quality and robustness to Cropout with different strength factors.

Strength Factor	0.2	0.4	0.6	0.8	1	1.2	1.4	1.6	1.8	2
Bit Accuracy	0.9954	0.9945	0.9921	0.9928	0.9874	0.9957	0.9938	0.9901	0.9924	0.9934
PSNR	54.1025	48.1031	44.5956	42.1063	40.1767	38.6016	37.2712	36.1213	35.1117	34.2067
SSIM	0.9995	0.9979	0.9954	0.992	0.9878	0.9829	0.9775	0.9714	0.9649	0.9579

TABLE 5. The image quality and robustness to Dropout with different strength factors.

Strength Factor	0.2	0.4	0.6	0.8	1	1.2	1.4	1.6	1.8	2
Bit Accuracy	1	1	1	1	1	0.9999	1	1	1	1
PSNR	55.8589	49.8596	46.3535	43.8641	41.9346	40.3572	39.0232	37.8727	36.8573	35.9439
SSIM	0.9997	0.999	0.9978	0.9962	0.9941	0.9917	0.9889	0.9859	0.9827	0.9791

Fourier series, which is more mathematically explainable than heuristic methods like JPEG-Mask and JPEG-Drop. In general, the performance of our model is better than that of previous studies, which improves the accuracy of watermark decoding and maximizing the similarity with the original image. Because our model learns to embed more redundant information into the images, decreasing the PSNR. The redundant information is benefit to recover the original message bits, so the bit accuracy is higher.

2) OTHER DISTORTION

In addition to JPEG compression distortion, our proposed model can also be applied to other image processing distortions such as dropout and cropout. We train a model to embed a 30-bit message into a 128×128 image. We use Dropout ($p = 0.8$), Cropout ($p = 0.06$) layers to test the model, and test a distortion each time. As shown in Table 3, we found that our model outperforms other methods [12], [13], [20] under Identity, Dropout, Cropout distortions. This demonstrates the advantage of our network. In Identity, all methods have bit accuracy of 1 because no noise is added on the watermarked images. But our model has the highest PSNR and SSIM, i.e., the best image quality. In Dropout and Cropout, our approach shows the best results on all metrics. In general, the improvement of our method is related to the application of dual attention module in the model. The spatial attention module helps to find the most suitable embedding areas which can well withstand various distortions and the channel attention module makes the model focus on important channels in training.

Previous experiments were conducted on a fixed size of 128×128 for fair comparison. Due to the fact that the output image size is the same as the input image size, our model can also embed watermarks to images of any size, which meets the needs of the real world. To verify this property, we test the robustness of the model against Cropout and Dropout using images of the original size of the test set. We also vary the strength factor to observe its impact on the model performance. The results are show in Table 4 and Table 5. It can be seen that our method still has strong robustness with different image sizes under different strength factors. Besides, lower strength factor tends to produce images of higher quality, this is because less embedding changes are added on to the original image with lower strength factors.

D. ABLATION STUDY

In this section, we explore the reason why our method can outperform previous methods via ablation study. Thus we have a deep insight into all components and ultimately find the contribution of dual attention and the differentiable rounding function of JPEG compression in our method make a great contribution to the final data hiding effect. We conduct ablation study on the JPEG simulator, dual attention module, and the strength factor respectively, and the results are summarized as follows.

1) DIFFERENTIABLE JPEG SIMULATOR

It is easy to figure out that the differentiable rounding function contributes to a large portion of the model bit accuracy in the robustness against JPEG compression. We design the test experiment with the JPEG noise layer on two different trained models. One is a model trained with identity noise, the other is trained with our JPEG simulator. As shown in Table 6, the bit accuracy of the model trained on identity is only 0.5070. After adding the JPEG simulator, the bit accuracy reaches 0.9946. As the bit accuracy can be considered a criterion for evaluating robustness, we can claim that it has a giant performance boost after adding the JPEG simulator to the model.

The JPEG simulator has a parameter K in Equation (6). It maintains the balance between the approximation accuracy and computation efficiency. Setting K to be higher will make the JPEG simulator more similar to the real JPEG compression. We change the parameter K to conduct experiments. The results are shown in Table 7. When K increases, the robustness against JPEG compression gets better, but the image quality decreases. To balance the image quality and the robustness, we set $K = 12$ in the main experiments.

2) DUAL ATTENTION MECHANISM

To illustrate the importance of our new structure of dual attention mechanism, we train four different models under different structures, which will be in more detail about

TABLE 6. The ablation study on the JPEG simulator. Two models are trained: the model trained with identity noise, and with JPEG simulator.

	PSNR	SSIM	Bit accuracy
Identity	51.5590	0.9991	0.5070
JPEG simulator	32.1045	0.9597	0.9946

TABLE 7. The ablation study on parameter K of the JPEG simulator.

K	1	5	8	10	12	15
PSNR	42.4496	40.3462	36.2347	34.8527	32.4048	32.1023
SSIM	0.9928	0.9873	0.9714	0.9679	0.9566	0.9512
Bit Accuracy	0.6899	0.8137	0.9123	0.9634	0.9974	0.9989

TABLE 8. The ablation experiment of dual attention. Baseline:without attention networks; with channel attention only; with spatial attention only; with both channel and spatial attention. All these results are tested under "dropout noise".

ID	None	Spatial attention	Channel attention	Mix
PSNR	30.9143	38.458	38.8582	41.9346
SSIM	0.9768	0.9893	0.9882	0.9941
Bit Accuracy	0.5774	0.9998	0.9999	1

TABLE 9. bit accuracy, PSNR, and SSIM values are logged under strength factors varying from 0.6 to 1.4. bit accuracy is tested under the different quality factors of JPEG compression.

Strength Factor		0.6	0.8	1	1.2	1.4
Bit Accuracy	Q=10	0.7622	0.7682	0.7626	0.7588	0.7692
	Q=30	0.9772	0.9751	0.9769	0.9729	0.9745
	Q=50	0.9942	0.9935	0.9946	0.9933	0.9939
	Q=70	0.9971	0.9970	0.9976	0.9965	0.9976
	Q=90	0.9984	0.9990	0.9989	0.9980	0.9988
PSNR		36.5277	34.0444	32.1045	30.5381	29.1994
SSIM		0.9844	0.9733	0.9597	0.9440	0.9272

them in Table 8. Table 8 shows the outcomes of the four models. In the table, the models from left to the right are: (1) None block: no attention model; (2) Spatial attention: add spatial attention network; (3) Channel attention: add channel attention network; (4) Dual attention: add both channel and spatial attention block. The model shows the best performance by combining both channel and spatial attention blocks, which is improved highly compared to the baseline under the dropout noise.

3) STRENGTH FACTOR

The strength factor is an adjustable parameter used to balance robustness and imperceptibility. We set the value of the strength factor S, from 0.6 to 1.4, with an interval of 0.2, and test the model under different quality factors for JPEG compression. The quality factor is a parameter applied for balancing the degree of compression and image quality. As the quality factor grows, the degree of compression will be lower, meanwhile, the quality of the image will be better and the bit accuracy increases. The results are shown in Table 9. As S increases, PSNR and SSIM values decrease, and the quality of the encoded image deteriorates while the extraction accuracy grows better.

We combine analyses of results from Table 9, it appears that our algorithm performs well at most compression factors. Moreover, changing the value of the strength factor will result in no significant disturbance of the performance. Considering the application in different scenarios, we should make a trade-off between bit accuracy and the quality of the image. For example, we can regulate the strength factor to obtain the 0.9990 bit accuracy, with PSNR = 34.04; or a much higher image quality with PSNR = 36.53 and bit accuracy = 0.9984.

V. DISCUSSION

This section mainly discusses the potential impact of our study and the limitation of our method.

A. IMPACT

The proposed algorithm improves the image quality and robustness of the existing DNN-based image watermarking. It will facilitate the copyright protection of images in real word applications, especially the social networks where the JPEG compression is typically used to process the images. The image own can use our algorithm to add a watermark into his images. When the images are illegally distributed or used by third-parties, he can extract the watermark to claim his ownership.

B. LIMITATION

The robustness of the proposed image watermarking algorithm relies on the noise layer in the model. This study assumes that the owner of the image has knowledge of the distortions the images will experience and introduces the distortion in the noise layer to resist it. JPEG compression if the most common distortion so it is our main concern. However, the robustness might be compromised if the images experience unknown distortions, i.e., the distortions not involved in the model training process.

VI. CONCLUSION

This study proposes a new architecture with dual attention mechanism and simulated JPEG compression for blind image watermarking. We achieve a better balance between robustness and imperceptibility with channel attention and spatial attention integrated into the network architecture. By introducing a differentiable JPEG simulator with an approximate rounding function, we realize stronger robustness against JPEG compression. We evaluate our method through extensive experiments, and the results demonstrate that the superiority of our method over previous approaches in image quality and robustness.

In the future work, we hope to improve the robustness of our watermarking algorithm to unknown distortions and more non-differentiable image manipulations. We are also interested in investigating how to extend our algorithm to embed user fingerprints into the images to trace unauthorized usage. Moreover, we would like to explore the feasibility to apply our method to hide a full image in another image as watermarks.

REFERENCES

[1] M. Li, Y. Liu, Z. Tian, and C. Shan, "Privacy protection method based on multidimensional feature fusion under 6G networks," *IEEE Trans. Netw. Sci. Eng.*, vol. 10, no. 3, pp. 1462–1471, May 2023.

[2] J. Qiu, Z. Tian, C. Du, Q. Zuo, S. Su, and B. Fang, "A survey on access control in the age of Internet of Things," *IEEE Internet Things J.*, vol. 7, no. 6, pp. 4682–4696, Jun. 2020.

[3] M. Li, C. Shan, Z. Tian, X. Du, and M. Guizani, "Adaptive information hiding method based on feature extraction for visible light communication," *IEEE Commun. Mag.*, vol. 61, no. 4, pp. 102–106, Apr. 2023.

[4] I. Cox, M. Miller, J. Bloom, and C. Honsinger, "Digital watermarking," *J. Electron. Imag.*, vol. 11, no. 3, p. 414, 2002.

[5] C.-T. Hsu and J.-L. Wu, "Hidden digital watermarks in images," *IEEE Trans. Image Process.*, vol. 8, no. 1, pp. 58–68, Jan. 1999.

[6] J. R. Hernandez, M. Amado, and F. Perez-Gonzalez, "DCT-domain watermarking techniques for still images: Detector performance analysis and a new structure," *IEEE Trans. Image Process.*, vol. 9, no. 1, pp. 55–68, Jan. 2000.

[7] I. J. Cox, J. Kilian, F. T. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," *IEEE Trans. Image Process.*, vol. 6, no. 12, pp. 1673–1687, Dec. 1997.

[8] G. C. Langelaar, I. Setyawan, and R. L. Lagendijk, "Watermarking digital image and video data. A state-of-the-art overview," *IEEE Signal Process. Mag.*, vol. 17, no. 5, pp. 20–46, 2000.

[9] P. Bassia, I. Pitas, and N. Nikolaidis, "Robust audio watermarking in the time domain," *IEEE Trans. Multimedia*, vol. 3, no. 2, pp. 232–241, Jun. 2001.

[10] M. D. Swanson, B. Zhu, A. H. Tewfik, and L. Boney, "Robust audio watermarking using perceptual masking," *Signal Process.*, vol. 66, no. 3, pp. 337–355, May 1998.

[11] R. G. van Schyndel, A. Z. Tirkel, and C. F. Osborne, "A digital watermark," in *Proc. 1st Int. Conf. Image Process.*, 1994, pp. 86–90.

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

[13] 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.

[14] J. Tan, Y. Hu, Z. Shi, and B. Wang, "Deep image watermarking to JPEG compression based on mixed-frequency channel attention," *Comput. Math. Methods Med.*, vol. 2022, Jul. 2022, Art. no. 9880038.. [Online]. Available: https://europepmc.org/articles/PMC9303108

[15] Z. Qin, P. Zhang, F. Wu, and X. Li, "FcaNet: Frequency channel attention networks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 763–772.

[16] H. Zhang, K. Zu, J. Lu, Y. Zou, and D. Meng, "EPSANet: An efficient pyramid squeeze attention block on convolutional neural network," in *Proc. Asian Conf. Comput. Vis.*, 2022, pp. 1161–1177.

[17] H. Wang, Y. Fan, Z. Wang, L. Jiao, and B. Schiele, "Parameter-free spatial attention network for person re-identification," 2018, *arXiv:1811.12150*.

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

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

[20] 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.

[21] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. NIPS*, 2014, pp. 1–9.

[22] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2015, *arXiv:1511.06434*.

[23] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," in *Proc. 34th Int. Conf. Mach. Learn.*, 2017, pp. 214–223.

[24] J. Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, Jun. 2017, pp. 2223–2232.

[25] P. Isola, J. Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2017, pp. 1125–1134.

[26] M. Mirza and S. Osindero, "Conditional generative adversarial nets," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 2672–2680.

[27] R. Shin and D. Song, "Jpeg-resistant adversarial images," in *Proc. NIPS*, vol. 1, 2017, p. 8.

[28] S. Woo, J. Park, J.-Y. Lee, and I. So Kweon, "CBAM: Convolutional block attention module," 2018, *arXiv:1807.06521*.

[29] Y. Xing, Z. Qian, and Q. Chen, "Invertible image signal processing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 6283–6292.

[30] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 3–19.

[31] 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 *Computer Vision—ECCV*. Zurich, Switzerland: Springer, 2014, pp. 740–755.

JIA-YI ZHONG is currently pursuing the bachelor's degree with the School of Information Science and Technology, Northwest University, Xi'an, China. Her research interests include machine learning and image processing.

DUO-WEN PAN is currently pursuing the bachelor's degree with the International School, Beijing University of Posts and Telecommunications, Beijing, China. His research interests include machine learning and computer vision.

JING-JIE WANG is currently pursuing the bachelor's degree in mathematics and in applied mathematics and statistics.

XUAN-BO JIA is currently pursuing the bachelor's degree in computer engineering and in computer science with Hangzhou Dianzi University, China. His research interest includes machine learning.