Deep Boosting Robustness of DNN-based Image Watermarking via DBMARK

Guanhui Ye  
Department of Computer Science and Engineering  
Southern University of Science and Technology  
Shenzhen, China  
12132370@mail.sustech.edu.cn

Wei Xie  
School of Journalism and Communication  
Hunan University  
Changsha, China  
tntxie@hnu.edu.cn

Xuetao Wei  
Department of Computer Science and Engineering  
Southern University of Science and Technology  
Shenzhen, China  
weixt@sustech.edu.cn

Jiashi Gao  
Department of Computer Science and Engineering  
Southern University of Science and Technology  
Shenzhen, China  
12131101@mail.sustech.edu.cn

Abstract—Image watermarking is a technique for hiding information into images that can withstand distortions while requiring the encoded image to be perceptually identical to the original image. Recent work based on deep neural networks (DNN) has achieved impressive progression in digital watermarking. Higher robustness under various distortions is the eternal pursuit of these DNN-based image watermarking approaches. In this paper, we propose DBMARK, a novel end-to-end digital image watermarking framework to deep boost the robustness of DNN-based image watermarking. The key novelty is the synergy of invertible neural networks (INN) and effective watermark features generation. The framework generates watermark features with redundancy and error correction ability through the effective neural network based message processor, synergized with the powerful information embedding and extraction abilities of INN to achieve higher robustness and invisibility. The powerful learning ability of neural networks enables the message processor to adapt to various distortions. In addition, we propose to embed the watermark information in the discrete wavelet transform (DWT) domain and design low-low (LL) sub-band loss to enhance invisibility. Extensive experiment results demonstrate the superiority of the proposed framework compared with the state-of-the-art ones under various distortions such as dropout, cropout, crop, Gaussian filter, and JPEG compression.

Keywords—Image Watermarking, Video Watermarking, Convolutional Neural Networks, Robustness, Invisibility

I. INTRODUCTION

Digital watermarking has been widely used in the copyright protection of multimedia products since its inception [1]. Digital watermarking hides the watermark information with specific meanings in digital content, such as images, videos, audio, documents, etc., through digital embedding. The extraction and recovery of watermark information can be used to prove the ownership and as evidence for identifying illegal infringement. The goal of digital watermarking is to embed the secret message into the cover image in an invisible way and to extract the accurate secret message in the case of various distortions. In other words, digital watermarking requires high robustness and high invisibility.

In recent years, with the upsurge of deep learning, many researchers have applied deep neural networks (DNN) to digital image watermarking, which significantly facilitates its development. These DNN-based methods [2]–[4] have shown advantages in robustness under various distortions compared with traditional methods. Zhu et al. [2] proposed the first DNN-based method named Hidden and demonstrated superior performance than most traditional methods. Meanwhile, various subsequent DNN-based methods have adopted a similar framework. Such a framework uses a separate encoder and decoder, which treats the watermark encoding and decoding processes independently. Xu et al. [5] simply applied invertible neural networks (INN) in image watermarking, which did not achieve satisfactory performance. Since higher robustness is the eternal pursuit of these DNN-based image watermarking approaches, our research question is: how to deep boost the robustness of DNN-based image watermarking under various distortions?

In this paper, we propose a novel end-to-end digital image watermarking framework DBMARK to deep boost the robustness of DNN-based image watermarking. The key novelty is the synergy of invertible neural networks (INN) and effective watermark features generation. The framework generates watermark features with redundancy and error correction ability through the effective neural network based message processor, synergized with the powerful information embedding and extraction abilities of INN to achieve higher robustness and invisibility. The powerful learning ability of neural networks enables the message processor to adapt to various distortions. We embed and extract watermark features by INN, in which the forward and inverse processes of INN share the same parameters and modules. Instead of hiding information directly in the spatial domain, we embed watermark information in...
the frequency domain of the cover image through the DWT. Moreover, we embed most of the watermark information into the high-frequency component through the LL sub-band loss to improve invisibility from the perspective of the human visual system. Experiment results show that our DBMARK framework outperforms existing SOTA (State Of The Art) methods on robustness and invisibility evidently.

In summary, the main contributions of this paper are:

- We present a novel end-to-end digital image watermarking framework DBMARK to deep boost the robustness of DNN-based image watermarking, which is the synergy of the invertible neural networks (INN) and effective watermark features generation.
- We propose the neural network based message processor to generate watermark features with redundancy and error correction ability, significantly improving robustness against various distortions simultaneously.
- We propose low-low (LL) sub-band loss to embed more watermark information in the high-frequency component of the discrete wavelet transform (DWT) domain, which enhances the invisibility of our DBMARK evidently.

II. RELATED WORK

In recent years, DNN-based methods have shown more advantages than traditional methods in both invisibility and robustness against various distortions, which is due to the powerful feature extraction ability of deep neural networks. Zhu et al. [2] and Shin et al. [6] proposed differentiable JPEG-Mask and JPEG-SS respectively to approximate the real JPEG. However, the gap between the JPEG simulation and the real JPEG results in poor robustness against real JPEG. Liu et al. [3] proposed a two-stage separable framework to solve the non-differentiable distortions problem. The method jointly trains the encoder and the decoder without a noise layer in stage one to obtain a powerful encoder. In stage two, the encoder is frozen and the decoder gains robustness enhancement from noise layer. Jia et al. [4] focused on JPEG compression and proposed a mixed training method that randomly selects one from Identity, JPEG-Mask and real JPEG as noise layer during training. Furthermore, Zhang et al. [7] proposed the forward attack simulation layer method to improve the robustness against non-differentiable distortions in an end-to-end framework. The previous work has demonstrated that DNN have great potential in digital image watermarking. Since the Invertible Neural Network was first proposed by Nice [8], INN has been applied for many computer vision tasks due to its outstanding performance. INN can be seen as a bijective function, which contains a forward mapping \( f_\theta(\cdot) \) and a corresponding inverse mapping \( f_\theta^{-1}(\cdot) \). Both mappings have a tractable Jacobian matrix, allowing explicit computation of posterior probabilities. Therefore, for the computational result that \( y = f_\theta(x) \), we can recover \( x \) directly through \( x = f_\theta^{-1}(y) \). Although INN have great potential in information embedding and extraction, it shows weakness in robustness against lossy data compression and other distortions, which are the key issues of digital watermarking. Xu et al. [5] applied INN in image watermarking, but the simple usage of INN can not achieve high robustness. CIN [9] combined non-invertible mechanisms with INN and achieved excellent performance. However, it has shown weakness against JPEG compression and required an additional module to handle JPEG compression distortion.

III. THE FRAMEWORK DBMARK

In this section, we describe the proposed framework DBMARK in detail. As shown in Figure 1, the proposed model is an end-to-end framework including DWT modules, message processors, INN blocks, and a noise layer.

A. Network Architecture

Figure 1 shows the framework architecture of our DBMARK. In the watermark encoding process, the cover image and secret message are received as pair-wise input. The cover image \( I_{co} \in \mathbb{R}^{C \times H \times W} \) is first transformed into frequency domain features \( F_{co} \in \mathbb{R}^{4C \times H/2 \times W/2} \) through DWT, in which \( C \) is channel number, \( H \) is height, and \( W \) is width. The secret message \( M \in \{0, 1\}^L \) needs to enter the message processor to generate the watermark features \( F_M \in \mathbb{R}^{C_M \times H/2 \times W/2} \) with the same dimension as the frequency domain features \( F_{co} \). \( C_M \) is the channel number of watermark features, which is set to equal \( L \) by default. Then, these two kinds of features go through a series of invertible neural network blocks. The output of the last block contains the frequency domain feature of encoded image \( F_{en} \) and the lost information \( r \). The encoded image features \( F_{en} \) produce the encoded image \( I_{en} \) after inverse discrete wavelet transform (IDWT). The noise layer adds random noise to the encoded image and outputs the noised image \( I_{no} \). In the watermark decoding process, the noise image features \( F_{no} \) and an auxiliary matrix \( z \) pass through a series of invertible neural network blocks to extract the watermark features \( F_M' \), which are then fed into the inverse message processor to recover the secret message.

Discrete Wavelet Transform. DWT is a transform that decomposes the original image into different sub-bands, namely low-low (LL), low-high (LH), high-low (HL), and high-high (HH). The LL sub-band contains no edge information since it is the low-frequency component of the image. According to the masking effect of the human sensory system, the digital watermark information can be embedded in the area where the original carrier is not easily perceptible, especially the high-frequency component in the DWT domain, so that the digital watermark has strong invisibility. Here, we use the Haar wavelet to implement our DWT algorithm, which is simple and effective. After DWT, the feature map of the cover images with size \((C, H, W)\) is transformed into a frequency domain with size \((4C, H/2, W/2)\).

Message Processor. In the watermark encoding process, the one-dimensional secret message \( M \) needs to be processed to two-dimensional watermark features \( F_M \) with the same spatial dimension as the frequency domain features of cover images \( F_{co} \). Here, we design a neural network to generate watermark features instead of simply duplicating the message.
or using a manual coding algorithm. \( F_M \) contains much more bits than the original secret message, making it redundant and error-correcting. Here, we design a message processor to generate watermark features. First, the secret message \( M \in \{0, 1\}^l \) passes through a fully connected layer to increase the length of the message to \( L' \). The motivation for this fully connected layer is from [4], which uses the full connection layer to diffuse watermark information to the whole image to resist distortions. Here, we utilize the fully connected layer to diffuse the watermark features to the frequency domain of the cover image. Then, the message is reshaped to a 3-dimensional tensor \( [0, 1]^{1 \times h \times w} \) where \( L' = h \times w \). It is fed into a single convolution layer to increase the channel number to \( C_M \). And then, the shape of \( F_M \) expanded to \( C_M \times H / 2 \times W / 2 \) by several \( 2 \times \) upsampling layers. After that, the watermark features are sent to several Squeeze-and-Excitation Network (SENet) [10] blocks and then pass through INN concatenated with the cover image features.

In the watermark decoding process, the inverse message processor receives the watermark features \( F_{M'} \) from the first block of INN and outputs the recovered message \( M' \). The inverse message processor has a similar network structure to the message processor but with a reverse information flow direction. During the upsampling and downsampling step, the number of upsampling or downsampling layers \( n \) is decided by \( L', H \), and \( W \). The formula is defined as follows,

\[
L' = h \times w = (H/2^{n-1}) \times (W/2^{n-1}).
\]

**Forward and Inverse Process in INN.** The INN has shown outstanding performance in image steganography [11], [12]. In our proposed framework, we build the INN blocks to embed and extract the watermark features to the image. The INN consists of \( N \) INN blocks with the same structure. As shown in Figure 1, the forward and inverse processes in the INN share the same INN blocks but with reverse data flow directions. In the forward process, the INN receives the frequency domain features of the cover image \( F_{en} \) and watermark features \( F_{M} \) as inputs. For the \( i \)-th INN block, we denote the inputs as \( b_1 \) and \( b_2 \). The corresponding outputs \( b_1^{i+1} \) and \( b_2^{i+1} \) are calculated as follows,

\[
b_1^{i+1} = b_1 + \phi (b_2^i),
\]

\[
b_2^{i+1} = b_2^i \odot \exp (\alpha (\rho (b_1^{i+1}))) + \eta (b_1^{i+1}),
\]

where \( \phi (\cdot) \) is a sigmoid function multiplied by a constant, \( \exp (\cdot) \) is an exponential function, and \( \odot \) is the Hadamard product. Here, \( \phi (\cdot), \rho (\cdot) \) and \( \eta (\cdot) \) are arbitrary functions and we apply the dense block for its simplicity and effectiveness. Finally, the INN outputs \( b_2^N +1 \) and \( b_2^{N+1} \), where \( b_2^N +1 \) are the frequency domain features of encoded image \( F_{en} \) and \( b_2^{N+1} \) is lost information \( r \). \( F_{en} \) is then processed by IWT to generate the encoded image \( I_{en} \).

Accordingly, the inverse process in the \( i \)-th block of INN is formulated as follows,

\[
b_1^i = b_1^{i+1} - \phi (b_2^i),
\]

\[
b_2^i = (b_2^i - \eta (b_1^{i+1})) \odot \exp (-\alpha (\rho (b_1^{i+1}))).
\]

In order to keep the dimensions of the input and output of the INN consistent, the input contains an auxiliary matrix \( z \) in addition to the frequency domain features \( F_{en} \) generated by the noised image passing through the DWT module. The auxiliary matrix \( z \) is randomly sampled from a Gaussian distribution.
Specifically, for the $N$-th block, the inputs $b_1^{N+1}$ and $b_2^{N+1}$ are the frequency domain features of the noised image $F_{no}$ and the auxiliary matrix $z$ correspondingly. Finally, the output of the first block $b_2^{1}$ is fed into the inverse message processor as the watermark feature $F_M$.

**Noise Layer.** To improve the robustness against various distortions, inserting a noise layer after encoded images is the most common method in deep learning. However, the standard noise layer cannot gain enhanced robustness from non-differentiable noise like JPEG. Zhang et al. [7] proposed a forward attack simulation layer method to overcome such limitations. The method calculates the difference between noised and encoded images and indicates this difference as the pseudo-noise. Then, the pseudo-noise is added to the encoded image to generate the pseudo-noised image. During the backward propagation, the pseudo-noise does not participate in the gradient propagation. Therefore, the gradient from the inverse blocks is directly back-propagated to the forward process in INN without passing through the noise layer. In our DBMARK, we propose to apply the forward attack simulation layer method for non-differentiable noise and standard noise layer for differentiable noise.

**B. Loss Functions**

Our DBMARK has two main goals. The goal of the watermark encoding process is to embed the secret message $M$ into the cover image $I_{co}$ to generate an encoded image $I_{en}$. The encoded image is required to be as similar as possible to the cover image. Therefore, the encoding loss function $\mathcal{L}_{en}$ is calculated by mean square error (MSE),

$$\mathcal{L}_{en} = MSE(I_{co}, I_{en}). \quad (6)$$

The watermark decoding process aims to extract and recover the secret message from the encoded image and minimize the difference between the original secret message $M$ and the recovered message $M'$. According to this goal, we define the decoding loss function as follows,

$$\mathcal{L}_{de} = MSE(M, M'). \quad (7)$$

Since the human eye is less sensitive to noise in high-frequency sub-bands of the image, Zebbech et al. [13] proposed a perceptual masking model to embed the hidden information into the high-frequency components in the DWT domain, which enhances the performance significantly in terms of invisibility and robustness. Inspired by this work, we propose a $LL$ sub-band loss $\mathcal{L}_{LL}$ to enhance our DBMARK's invisibility. Suppose that $\mathcal{F}(\cdot)_{LL}$ is a filter to extract the $LL$ sub-band of the image. Here, we minimize the difference between the cover image and encoded image in $LL$ sub-band to allow more watermark information to be embedded in the high-frequency components of the encoded image. Therefore, the $LL$ sub-band loss is defined as follows,

$$\mathcal{L}_{LL} = MSE(\mathcal{F}(I_{co})_{LL}, \mathcal{F}(I_{en})_{LL}). \quad (8)$$

In summary, the total loss function $\mathcal{L}_{total}$ is formulated as:

$$\mathcal{L}_{total} = \lambda_{en}\mathcal{L}_{en} + \lambda_{de}\mathcal{L}_{de} + \lambda_{LL}\mathcal{L}_{LL}, \quad (9)$$

where $\lambda_{en}$, $\lambda_{de}$ and $\lambda_{LL}$ are the weights of the corresponding losses for balancing different loss functions.

**IV. Experiments**

In this section, we conduct experiments to validate the efficacy and robustness of our proposed framework DBMARK. Performance against the baseline and parameter sensitivity on performance are considered for effectiveness verification. The experiments cover a wide range of complex image distortions in terms of robustness.

**A. Experimental Settings**

**Implementation Details.** Our DBMARK is trained under 10000 images of COCO training data set, and the neural network is evaluated through 5000 images of COCO test data set to ensure the generalization of the model. Our model is implemented through PyTorch and runs on NVIDIA A100 GPUs. In order to compare with previous work, we adjust the image’s resolution to 128 × 128. The secret message lengths are 30 bits and 64 bits for combined training and specified training, where each bit is generated by random sampling. Our DBMARK contains 16 invertible neural network blocks. The parameters $\lambda_{en}$, $\lambda_{de}$ and $\lambda_{LL}$ are set to 0.1, 100.0 and 0.1, respectively. The mini-batch size is 16, and the Adam optimizer is adopted with standard parameters and an initial learning rate of $1 \times 10^{-5}$.0.

**Metrics.** There are two objectives in digital image watermarking: robustness, measured by the Bit Error Rate (BER) of the recovered message; and invisibility, measured by both peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) between the cover and encoded image.

**Benchmarks.** To verify the performance of our proposed DBMARK, we conduct the comparison with four DNN-based methods [2], [3], [5] and [4], especially with the state-of-the-art (SOTA) digital watermark method proposed in [4]. The architecture of [5] is based on INN, but the author does not open-source their code. Although [2] and [3] open-sourced their code, we cannot reproduce the results as they reported. In order to respect the results they have reported, we directly compare them with the results published in [2], [3] and [5], [4] open-sourced both their codes and their model, so we use the pre-trained model for comparison.

**B. Experimental Results**

In this subsection, we show the comparison results between our DBMARK and previous methods. For JPEG compression, since the message length of each method is different, we use a larger message length to train our model for a fair comparison. Specifically, we use secret message length $L = 64$, which is larger than [2], [3] and [5]. For combined compression, we use secret message length $L = 30$. In terms of visual quality, our watermark mask is distributed across the image like vertical stripes.
TABLE I
Comparison with [2], [3], [5] and [4] trained by specified noise layer. SSIM is not reported in [2], [3] and [5]. PSNR is measured for RGB channels, except in [2], they use Y channel of YUV channels. Our DBMARK achieves the highest PSNR/SSIM and the lowest BER.

| Method       | Hidden [2] | TSDL [3] | IWN [5] | MBRS [4] | DBMARK |
|--------------|------------|----------|---------|----------|--------|
| Message size | 30         | 30       | 30      | 64       | 64     |
| Noise Layer  | JPEG-Mask  | JPEG     | JPEG-SS | Mixed(MBRS) | Forward ASL |
| PSNR (for Y) | 30.09      | 33.51    | 36.16   | 36.49    | 39.75  |
| SSIM         | 6.26%      | 6%       | 7%      | 11%      | 14%    |
| BER          | 0.005%     | 0.0092%  | 0.0092% | 0.9137   | 0.9486 |

TABLE II
Comparison with [2], [3], [5] and [4] trained by combined noise layer. Our DBMARK achieves the lowest BER under six types of distortions.

| Method       | PSNR  | Identity | Cropout \((p = 0.3)\) | Dropout \((p = 0.3)\) | Crop \((p = 0.035)\) | GF \((\sigma = 2)\) | JPEG \((Q = 50)\) |
|--------------|-------|----------|------------------------|------------------------|----------------------|-------------------|------------------|
| Hidden [2]   | 33.5  | 0%       | 6%                     | 7%                     | 12%                  | 4%                | 37%              |
| TSDL [3]     | 33.5  | 0%       | 2.7%                   | 2.6%                   | 11%                  | 1.4%              | 23.8%            |
| IWN [5]      | 33.0  | 0.06%    | 5.29%                  | 24.71%                 | 16.69%               | 13.89%            | 23.13%           |
| MBRS [4]     | 33.5  | 0%       | 0.0027%                | 0.0087%                | 4.15%                | 0.011%            | 4.48%            |
| DBMARK       | 34.8  | 0%       | 0.002%                 | 0.0013%                | 3.74%                | 0%                | 0%               |

TABLE III
Effectiveness of message processor and \(\mathcal{L}_{LL}\) loss.

| Message processor | \(\mathcal{L}_{LL}\) loss | PSNR | SSIM | Identity | Cropout | Dropout | Crop | GF | JPEG |
|-------------------|---------------------------|------|------|----------|---------|---------|------|----|------|
| ✓                 | 36.36                     | 0.9060 | 0%   | 0%       | 0.0013% | 0.002% | 8.26% | 0.008% | 0.012% |
| ×                 | 36.03                     | 0.8970 | 0%   | 0%       | 0.004%  | 8.11%  | 0.009% | 0.011% |
| ✓                 | 38.32                     | 0.9321 | 0%   | 0.0007%  | 0.0047% | 5.68%  | 0%   | 0.0034% |

1) JPEG Comparison: In this experiment, we mainly compare the JPEG robustness between the proposed method and four other methods. [2], [3], [5] and [4]. Our model is trained by the forward ASL method proposed by [7] with real JPEG. All the testing processes are performed under real JPEG compression with quality factor \(Q = 50\). As shown in Table I, our DBMARK achieves higher PSNR and SSIM and lower BER, which indicates higher invisibility and robustness. Significantly, the BER is only 0.0018%. Experimental results demonstrate that our DBMARK is more effective against JPEG compression than the previous methods.

2) Combined Comparison: To demonstrate that our DBMARK is resistant to various distortions simultaneously, we train a noise-resistant combined model by using a random noise layer for each mini-batch during the training. The noise layers include Identity, Cropout \((p = 0.3)\), Dropout \((p = 0.3)\), Crop \((p = 0.035)\), Gaussian Filter \((GF, \sigma = 2)\) and JPEG \((Q = 50)\). We use forward ASL for JPEG since it is a non-differentiable distortion. We adjust the strength factor to achieve the best performance of our DBMARK. When the strength factor is large, with the increment of the strength factor, the robustness and invisibility of the encoded image decrease simultaneously. Therefore, we do not use a large strength factor to obtain similar PSNR value with different methods. As we can see in Table II, our DBMARK can maintain higher PSNR and lower BER than previous methods under each kind of distortion. In particular, our method can achieve 0% bit error rate for Gaussian filter and JPEG compression distortions. Furthermore, our DBMARK has a dramatic improvement in robustness compared to [5], which is also implemented based on INN.

C. Ablation Study

1) Message Processor Analysis: Since the INN is not robust against distortions, we design a neural network as the message processor to enhance the robustness of the proposed framework. Specifically, as shown in the second and third row of Table III, the message processor evidently decreases the BER values under various distortions. Meanwhile, the PSNR value increases 2.29 dB and the SSIM value increases 0.0351. Although the distortion modifies the hidden information in the encoded image, the message processor still has enough information to recover the secret message from the redundant watermark features through powerful error correction ability.

2) Effectiveness of \(LL\) sub-band loss: The \(\mathcal{L}_{LL}\) loss is designed to hide more watermark information in the high-frequency component of the cover image so that the human eye can hardly detect the presence of a watermark. The first and third rows in Table III show the quantitative results of \(\mathcal{L}_{LL}\) loss. At a similar BER, the average PSNR of our DBMARK with \(\mathcal{L}_{LL}\) loss increases 1.76 dB, and the average SSIM increases 0.0261. Visual comparisons of encoded images with and without \(\mathcal{L}_{LL}\) loss are shown in Figure 2. As we can see, the encoded images with \(\mathcal{L}_{LL}\) Loss maintain a higher image quality rather than the one without \(\mathcal{L}_{LL}\) loss, which...
Fig. 2. Samples of encoded and cover images for our DBMARK with/without $L_{LL}$ loss. From left to right are: the cover image $I_{co}$, the encoded image with $L_{LL}$ loss $I_{en}$, the difference between $I_{en}$ and $I_{co}$, the encoded image without $L_{LL}$ loss $I'_{en}$, the difference between $I'_{en}$ and $I_{co}$. The differences are five times magnified for visualization.

demonstrates that the $L_{LL}$ loss can enhance the invisibility of our method. In addition, the differences between $I_{co}$ and $I'_{en}$ are like the afterimage of the cover image. In contrast, we can only see some sparse vertical artifacts in the differences between $I_{co}$ and $I_{en}$.

V. CONCLUSION

In this paper, we have proposed a novel end-to-end digital image watermarking framework DBMARK to deep boost the robustness of DNN-based image watermarking. We have proposed to generate robust watermark features through message processing against distortions. In model training, the $LL$ subband loss has been proposed to improve the visual quality of the encoded image. Our DBMARK has synergized the strengths of INN, discrete wavelet transform and the forward attack simulation layer method to offer better robustness and invisibility. Extensive experiments have demonstrated that our DBMARK can achieve digital image watermarking with higher robustness and invisibility, outperforming other SOTA methods evidently.

ACKNOWLEDGMENT

This work was supported in part by National Key R&D Program of China under Grant 2021YFF0900300 and in part by Key Talent Programs of Guangdong Province under Grant 2021QN02X166. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the funding parties.

REFERENCES

[1] R. G. Van Schyndel, A. Z. Tirkel, and C. F. Osborne, “A digital watermark,” in Proceedings of 1st international conference on image processing, vol. 2, pp. 86–90, IEEE, 1994.

[2] J. Zhu, R. Kaplan, J. Johnson, and L. Fei-Fei, “Hidden: Hiding data with deep networks,” in Proceedings of the European conference on computer vision (ECCV), pp. 657–672, 2018.

[3] Y. Liu, M. Guo, J. Zhang, Y. Zhu, and X. Xie, “A novel two-stage separable deep learning framework for practical blind watermarking,” in Proceedings of the 27th ACM International Conference on Multimedia, pp. 1509–1517, 2019.

[4] Z. Jia, H. Fang, and W. Zhang, “Mbrs: Enhancing robustness of dnn-based watermarking by mini-batch of real and simulated jpeg compression,” in Proceedings of the 29th ACM International Conference on Multimedia, pp. 41–49, 2021.

[5] H.-B. Xu, R. Wang, J. Wei, and S.-P. Lu, “A compact neural network-based algorithm for robust image watermarking,” arXiv preprint arXiv:2112.13491, 2021.

[6] R. Shin and D. Song, “Jpeg-resistant adversarial images,” in NIPS 2017 Workshop on Machine Learning and Computer Security, vol. 1, p. 8, 2017.

[7] C. Zhang, A. Karjauv, P. Benz, and I. S. Kweon, “Towards robust deep hiding under non-differentiable distortions for practical blind watermarking,” in Proceedings of the 29th ACM International Conference on Multimedia, pp. 5158–5166, 2021.

[8] L. Dinh, D. Krueger, and Y. Bengio, “Nice: Non-linear independent components estimation,” arXiv preprint arXiv:1410.8514, 2014.

[9] R. Ma, M. Guo, Y. Hou, F. Yang, Y. Li, H. Jia, and X. Xie, “Towards blind watermarking: Combining invertible and non-invertible mechanisms,” in Proceedings of the 30th ACM International Conference on Multimedia, pp. 1532–1542, 2022.

[10] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 7132–7141, 2018.

[11] J. Jing, X. Deng, M. Xu, J. Wang, and Z. Guan, “Hinet: deep image hiding by invertible network,” in Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 4733–4742, 2021.

[12] S.-P. Lu, R. Wang, T. Zhong, and P. L. Rosin, “Large-capacity image steganography based on invertible neural networks,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10816–10825, 2021.

[13] K. Zebbiche and F. Khelifi, “Efficient wavelet-based perceptual watermarking for robust fingerprint image watermarking,” IET Image Processing, vol. 8, no. 1, pp. 23–32, 2014.