Generative Model Watermarking Based on Human Visual System

Li Zhang[1], Yong Liu[1], Shaoteng Liu[2], Tianshu Yang[2], Yexin Wang[3], Xinpeng Zhang[1], and Hanzhou Wu[1(✉)]

[1] School of Communication and Information Engineering, Shanghai University, Shanghai 200444, China
{jmzhangli,xzhang}@shu.edu.cn, h.wu.phd@ieee.org
[2] AI Technology Center, Tencent Inc., Beijing 100080, China
{shaotengliu,gracetsyang}@tencent.com
[3] AI Technology Center, Tencent Inc., Shenzhen 518000, Guangdong, China
yexinwang@tencent.com

Abstract. Intellectual property protection of deep neural networks is receiving attention from more and more researchers, and the latest research applies model watermarking to generative models for image processing. However, the existing watermarking methods designed for generative models do not take into account the effects of different channels of sample images on watermarking. As a result, the watermarking performance is still limited. To tackle this problem, in this paper, we first analyze the effects of embedding watermark information on different channels. Then, based on the characteristics of human visual system (HVS), we introduce two HVS-based generative model watermarking methods, which are realized in RGB color space and YUV color space respectively. In RGB color space, the watermark is embedded into the R and B channels based on the fact that HVS is more sensitive to G channel. In YUV color space, the watermark is embedded into the DCT domain of U and V channels based on the fact that HVS is more sensitive to brightness changes. Experimental results demonstrate the effectiveness of the proposed work, which improves the fidelity of the model to be protected and has good universality compared with previous methods.

Keywords: Model watermarking · Deep learning · Human visual system

1 Introduction

Deep learning (DL) [1] has been applied in many application fields such as computer vision [2] and natural language processing [3]. It is foreseen that DL will continue bringing profound changes to our daily life. However, it is known that training a powerful DL model requires lots of computational resources and

---

L. Zhang and Y. Liu—Equally contributed authors.

well-labelled data. As a result, the trained DL model should be regarded as an important digital asset and should be protected from intellectual property (IP) infringement. Therefore, how to protect the IP of DL models has become a new technical challenge. It motivates researchers and engineers from both academia and industry to exploit *digital watermarking* [4] for IP protection of DL models [5], which is typically referred to as *model watermarking*. However, unlike conventional media watermarking that often treats the host signal as static data, DL models possess the ability of accomplishing a particular task, indicating that the algorithmic design of model watermarking should take into account the functionality of DL models.

An increasing number of model watermarking schemes are proposed in recent years. By reviewing the mainstream algorithms, advanced strategies realizing model watermarking can be roughly divided into three main categories, i.e., network based model watermarking, trigger based model watermarking, and dataset based model watermarking. Network based model watermarking algorithms aim at covertly embedding a watermark into the internal network weights [5–7] or network structures [8]. As a result, the watermark decoder should access the internal details fully or partly of the target DL model. Though trigger based model watermarking algorithms often change the internal parameters of the DL model through training from scratch or fine-tuning, the zero-bit watermark is carried by the mapping relationship between a set of trigger samples and their pre-specified labels. As a result, the watermark is extracted by analyzing the consistency between the labels and the prediction results of the target DL given a set of trigger samples [9,10]. Dataset based model watermarking algorithms exploit the entangled relationship between the DL model and a dataset related to the task of the DL model to construct a fingerprint of the DL model, which may be realized via a lossless way (i.e., without modifying the original model to be protected) to guarantee high security such as [11–13].

Recently, Wu *et al.* [14] propose a general watermarking framework for deep generative models. By optimizing a combined loss function, the trained model not only performs very well on its original task, but also automatically inserts a watermark into any image outputted by the model. In the work, the watermark is embedded into the model through watermark-loss back-propagation. As a result, by extracting the watermark from the outputted image, the ownership of the target model can be identified without interacting directly with the model. Yet another similar work is introduced by Zhang *et al.* [15], which was originally designed to resist surrogate attacks and also allows us to extract a watermark from the outputted image. Though [14] and [15] have demonstrated satisfactory watermark verification performance as reported in the papers, they both do not consider the characteristics of the human visual system (HVS). As a result, the visual quality may not be satisfactory from a broader perspective.

To tackle the aforementioned problem, we utilize the characteristics of HVS for system design of model watermarking. To this end, we propose two HVS based frameworks for generative model watermarking, i.e., HVS-RGB based generative model watermarking and HVS-YUV based generative model watermarking. Since the proposed two frameworks take into account the HVS characteris-

(a) HVS-RGB based generative model watermarking framework

(b) HVS-YUV based generative model watermarking framework
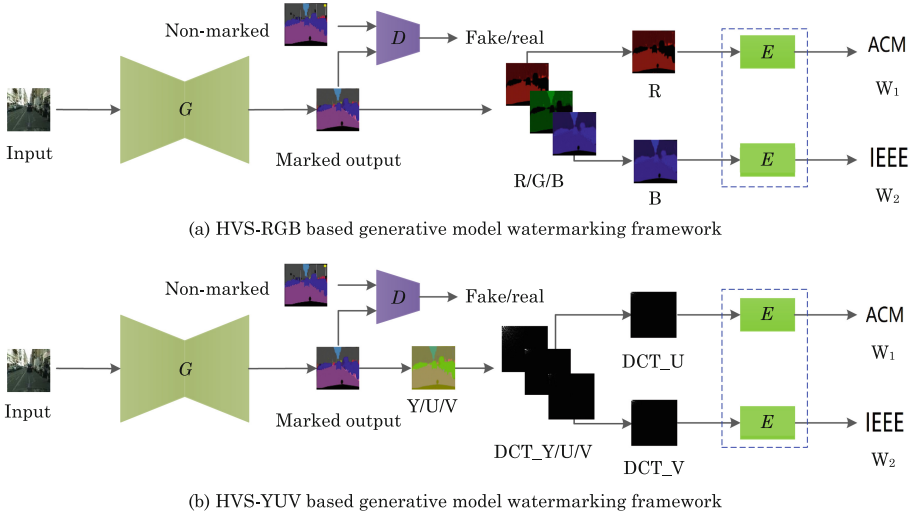
Fig. 1. The proposed two HVS based frameworks for generative model watermarking. For better understanding, the original task of G here is set to semantic segmentation.

tics, the generated *marked* images are more compatible with the HVS, thereby improving the fidelity of the network and the invisibility of the watermark. In summary, the main contributions of this paper include:

– We have analyzed the impact of embedding watermark on different channels of the generated images in the generative model watermarking by combining the HVS characteristics. Based on this analysis, we propose two HVS-based multi-channel generative model watermarking frameworks that improve the fidelity of the network while maintaining the fidelity of the watermark.
– Through convinced experiments on semantic segmentation, the results demonstrate the applicability and superiority of the proposed work.

The rest structure of this paper is organized as follows. In Sect. 2, we give the details of the proposed work. Then, we provide experiments and analysis in Sect. 3. Finally, we conclude this paper and provide discussion in Sect. 4.

2   Proposed Method

2.1   HVS Characteristics

HVS is one of the most complex biological systems [16–18] and typically includes the following characteristics:

– *Spectral characteristics:* The sensitivity of HVS to color is green, red, blue in descending order. Therefore, embedding a watermark in the green channel of

a color image will affect the visual quality of the image more than the others. It is desirable to preferentially embed the watermark into the blue channel and then the red channel. The green component can be changed with the least degree in order to improve the invisibility of the watermark.

– *Brightness characteristics:* The HVS is more sensitive to changes in brightness. Therefore, in YUV color space, embedding the watermark into U or V chromatic aberration channel results in better visual quality than Y channel.

The brightness and spectral characteristics of HVS illustrate the effects of embedding the watermark into different channels of the image. In conventional image watermarking, there are already methods verifying that HVS characteristics are helpful for developing high fidelity watermarking systems such as [18,19]. In this paper, we mainly combine the spectral and brightness characteristics to analyze the effects of watermark embedding in different channels of the generated image in generative model watermarking. In the later experiments, we will analyze the visual quality of the generated images after embedding the watermark in different channels, and analyze the results of the experiments by combining the spectral and brightness characteristics of HVS. It is proved that the HVS characteristics still hold in the generative model watermarking. Therefore, the HVS characteristics can guide us to design model watermarking methods that better match the characteristics of human visual perception system.

2.2   HVS-Based Generative Model Watermarking Frameworks

We propose two HVS-based generative model watermarking (GMW) frameworks, which are shown in Fig. 1. Below, we first provide the overview and then describe the watermark embedding and extraction process.

HVS-RGB Based GMW Framework: As mentioned above, human beings are more sensitive to the green color. Reducing the modifications to the green (G) channel results in better visual quality of the image, which inspires us to preferentially embed the watermark information into the red (R) channel and/or the blue (B) channel. Motivated by this insight, we propose a framework to embed two watermarks into the R channel and the B channel respectively. As shown in Fig. 1, the process can be described as follows. Given the generative model G to be protected, we train G in such a way that the image outputted by G is always marked. The watermark information is carried by the R channel and the B channel of the marked image. Specifically, we determine the R/G/B channels of the image outputted by G. The R channel of the image is used to carry a watermark W_1 and the B channel is used to carry another watermark W_2. To realize this purpose, a neural network E is trained together with G through loss optimization so that W_1 can be extracted from the R channel, and W_2 can be extracted from the B channel. To ensure that G has good performance on its original task, the image outputted by G may be fed to a discriminator D. Though we consider W_1 and W_2 as two different watermarks in this paper, one may consider them as a whole, which will not cause any problem.
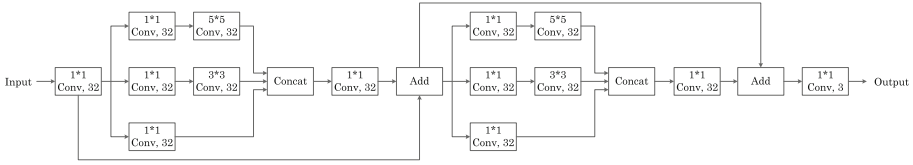
Fig. 2. Structural information for the watermark-extraction network E.

HVS-YUV Based GMW Framework: Similarly, human beings are more sensitive to changes in brightness. Therefore, by representing an image in the YUV color space, it is suggested to embed the watermark information in the U/V channels to not arouse noticeable artifacts. However, directly embedding watermark information into the U/V channels in the spatial domain may lead to significant distortion to the chroma. To address this problem, we propose to embed the watermark information into the U/V channels in the discrete cosine transform (DCT) domain. In this way, the entire process is similar to the above HVS-RGB based framework. The only one difference is that the image outputted by G should be converted to the Y/U/V color space. Then, the U channel and the V channel are processed by DCT. The transformed data can be expressed as DCT_U and DCT_V. Thereafter, DCT_U and DCT_V are fed to E for watermark embedding and watermark extraction.

Both frameworks are similar to each other according to the above analysis. For the two frameworks, we are now ready to describe the watermark embedding and extraction procedure for the host network G as follows:

– *Watermark embedding:* The watermark information is embedded into G by marking the output of G through optimizing a combined loss that will be detailed later. In other words, G is marked during joint training. The networks to be jointly trained include G, E and D, where D is optional.
– *Watermark extraction:* After training, G is deemed marked. By retrieving the watermark information from any image outputted by G with E, we are able to trace the source of the image and identify the ownership of the image and the target model. As multiple watermarks can be embedded, multiple purposes can be achieved.

2.3   Structural Design

G can perform any task outputting color images as results. In this paper, without the loss of generalization, we use the backbone network introduced in [20] as the host network G to be protected. The network structure is inspired by U-net [2]. The original task of G is limited to image semantic segmentation, whose goal is to cluster parts of an image together which belong to the same object class. The input and the output have the same size in this paper.

The objective of E is to extract the corresponding watermark from the corresponding channel (involving R, B, DCT_U, DCT_V). E accepts a single channel as input and then outputs the corresponding watermark (that may be

either a binary/gray-scale image or a color one). It is free to design the network structure of E. In this paper, the structure of E is inspired by [14,21]. ReLU [22] is applied as the activation function for the internal layers, except that the output layer uses TanH. The details of E are shown in Fig. 2. In addition, G can be optionally optimized by a discriminator D. For example, we may use PatchGAN [23] to serve as D shown in Fig. 1.

2.4   Loss Function

Based on the aforementioned analysis, we are now ready to design the loss function for each watermarking framework. We use S_0 to denote the set of images not generated by G and S_1 to denote the set of images generated by G.

Loss Function for HVS-RGB Based GMW Framework: For the watermarking task, we hope that E can extract the corresponding watermark given the corresponding input, which can be expressed as:

$$L_1 = \frac{1}{|S_1|} \sum_{G(x) \in S_1} ||E(G(x)_{\mathrm{R}}) - W_1|| \tag{1}$$

and

$$L_2 = \frac{1}{|S_1|} \sum_{G(x) \in S_1} ||E(G(x)_{\mathrm{B}}) - W_2|| \tag{2}$$

where x is the input image of G, $G(x)_{\mathrm{R}}$ and $G(x)_{\mathrm{B}}$ represent the R component and the B component of the outputted image $G(x)$, respectively. On the other hand, we hope that random noise can be extracted from images not generated by G, which requires us to minimize:

$$L_3 = \frac{1}{|S_0|} \sum_{y \in S_0} ||E(y_{\mathrm{R}}) - W_z|| \tag{3}$$

and

$$L_4 = \frac{1}{|S_0|} \sum_{y \in S_0} ||E(y_{\mathrm{B}}) - W_z|| \tag{4}$$

where y_{R} and y_{B} represent the R component and the B component of the image $y \in S_0$, and, W_z is a noise image randomly generated in advance. For the original task of G, it is expected that the *marked* image generated by G is close to the ground-truth image. Typically, to accomplish the task, we expect to minimize:

$$L_5 = \frac{1}{|S_1|} \sum_{G(x) \in S_1} ||G(x) - T[G(x)]|| \tag{5}$$

where $T[G(x)]$ means the ground-truth image of $G(x)$. Since the R component and the B component of $G(x)$ will be used to carry the additional information, distortion will be introduced. To reduce the impact of watermark embedding, it is desirable to further control the distortion in the R channel and the B channel, i.e., we hope to minimize:

$$L_6 = \frac{1}{|S_1|} \sum_{G(x) \in S_1} ||G(x)_{\text{R}} - T_{\text{R}}[G(x)]|| \tag{6}$$

and

$$L_7 = \frac{1}{|S_1|} \sum_{G(x) \in S_1} ||G(x)_{\text{B}} - T_{\text{B}}[G(x)]|| \tag{7}$$

where $T_{\text{R}}[G(x)]$ and $T_{\text{B}}[G(x)]$ represent the R component and the B component of $T[G(x)]$. Simply minimizing the above losses may not well maintain the structural information of the image, to deal with this problem, we further introduce structural loss for improving the image quality. Namely, we want to minimize:

$$L_8 = \frac{1}{|S_1|} \sum_{G(x) \in S_1} (1 - \frac{\text{SSIM}(G(x), T[G(x)]) + \text{MS-SSIM}(G(x), T[G(x)])}{2}) \tag{8}$$

where $\text{SSIM}(\cdot, \cdot)$ [24] and $\text{MS-SSIM}(\cdot, \cdot)$ [25] are two indicators representing the structural similarity between two images. The entire loss function is then:

$$L_{\text{HVS-RGB}} = \alpha(L_1 + L_2) + \beta(L_3 + L_4) + (L_5 + L_6 + L_7 + L_8) \tag{9}$$

where α and β are pre-determined parameters balancing the losses.

Loss Function for HVS-YUV Based GMW Framework: Similarly, the entire loss function can be divided into two parts, i.e., the watermarking loss and task loss. Referring to the loss function for HVS-RGB based GMW framework, the components of watermarking loss function include:

$$L_9 = \frac{1}{|S_1|} \sum_{G(x) \in S_1} ||E(G(x)_{\text{DCT\_U}}) - W_1|| \tag{10}$$

$$L_{10} = \frac{1}{|S_1|} \sum_{G(x) \in S_1} ||E(G(x)_{\text{DCT\_V}}) - W_2|| \tag{11}$$

$$L_{11} = \frac{1}{|S_0|} \sum_{y \in S_0} ||E(y_{\text{DCT\_U}}) - W_z|| \tag{12}$$

$$L_{12} = \frac{1}{|S_0|} \sum_{y \in S_0} ||E(y_{\text{DCT\_V}}) - W_z|| \tag{13}$$

where $G(x)_{\text{DCT\_U}}$ and $G(x)_{\text{DCT\_V}}$ represent the U channel and the V channel in the DCT domain for $G(x)$, respectively. $y_{\text{DCT\_U}}$ and $y_{\text{DCT\_V}}$ are the U channel and the V channel in the DCT domain for y. On the other hand, the task loss function involves L_5, L_8 and

$$L_{13} = \frac{1}{|S_1|} \sum_{G(x) \in S_1} ||G(x)_{\text{DCT\_U}} - T_{\text{DCT\_U}}[G(x)]|| \tag{14}$$

$$L_{14} = \frac{1}{|S_1|} \sum_{G(x) \in S_1} ||G(x)_{\text{DCT\_V}} - T_{\text{DCT\_V}}[G(x)]|| \tag{15}$$

Table 1. Quality evaluation for the marked images generated by the host network.

Channel	Average PSNR (dB)	Average SSIM	Average MS-SSIM
R	23.74	0.798	0.853
G	22.92	0.782	0.842
B	23.63	0.801	0.858
DCT_Y	22.85	0.781	0.840
DCT_U	23.49	0.794	0.848
DCT_V	23.60	0.793	0.853

where $T_{\text{DCT\_U}}[G(x)]$ and $T_{\text{DCT\_V}}[G(x)]$ represent the U component and the V component of $T[G(x)]$ in the DCT domain. So, the entire loss function is:

$$L_{\text{HVS-YUV}} = \alpha(L_9 + L_{10}) + \beta(L_{11} + L_{12}) + (L_5 + L_8 + L_{13} + L_{14}). \qquad (16)$$

Remark: There are two choices for determining the network parameters of G and E. One is to train G and E together from scratch according to the entire loss function. The other is to train G and E from scratch according to the task loss of G and the watermark-extraction loss corresponding to W_1. Then, G and E can be fine-tuned with the entire loss function. We use the latter strategy due to the relatively lower computational complexity.

3   Experimental Results and Analysis

3.1   Setup

As mentioned previously, we limit the task of G to image semantic segmentation. And, the structure of G is similar to U-net. However, it should be admitted that it is always free to design the network structure of G and specify the task of G. Since our purpose is not to develop G, it is true that the network structure used in this paper is not optimal in terms of the task performance, which, however, does not affect the contribution of this work. We use the dataset mentioned in [26] for experiments, where 4,000 images are used for model training, 500 images are used for validation and another 500 images are used for testing. The size of input and the size of output are set to $256 \times 256 \times 3$ for the host network. For W_1, W_2 and W_z, their sizes are all equal to $256 \times 256 \times 3$ since the watermark-extraction network in Fig. 2 keeps the size of the output same as the size of the input, indicating that the size of the output should be identical to that of the image outputted by G as the latter will be fed into the network in Fig. 2.

During model training, we empirically applied $\alpha = 1$ and $\beta = 0.5$. The Adam optimizer was used to iteratively update the network parameters. The learning rate was set to 2×10^{-4}. Our implementation used TensorFlow trained on a single RTX 3090 GPU. We use three evaluation metrics, i.e., peak signal-to-noise ratio (PSNR), structural similarity (SSIM) [24] and multi-scale structural similarity

Table 2. Quality evaluation for the marked images and the extracted watermarks.

Method	Mean PSNR	Mean SSIM	Mean MS-SSIM	Mean BER$_1$	Mean BER$_2$
HVS-RGB	23.12	0.807	0.857	0.0024	0.0029
HVS-YUV	22.83	0.799	0.852	0.0026	0.0031
[14]	20.83	0.738	0.816	0.0020	0.0028

(MS-SSIM) [25] for evaluation. For all of these widely used indicators, a higher value demonstrates the better quality of the image. In this paper, we limit W_1 to the logo image "ACM" and W_2 to the logo image "IEEE" (refer to Fig. 1). Besides, W_z is a randomly generated noise image. ℓ_1 norm is applied to all the loss functions, which can speed up convergence during training.

3.2   Analysis of Watermark Embedding in Different Channels

Before reporting experimental results, we first analyze the effects of watermark embedding on different channels of the image by taking into account the afore-mentioned spectral and brightness characteristics. To this purpose, we separate three channels from the image generated by the host network and feed each of the channels into the watermark-extraction network. For each channel, the host network together with the watermark-extraction network are trained together so that the image outputted by the host network carries a watermark that can be extracted by the watermark-extraction network. It is pointed that we here only tested the watermark W_1, where W_2 was not used for simplicity. In other words, all the channels independently carry the same watermark for fair comparison.

Table 1 provides the experimental results of image quality evaluation. It can be found that even though different color channels result in different performance in terms of image quality, overall, R/B and DCT_U/DCT_V are superior to G and DCT_Y, respectively. This indicates that from the viewpoint of watermark embedding, it is quite desirable to preferentially embed the watermark data into the R/B and DCT_U/DCT_V channels, rather than the G and DCT_Y channels.

3.3   Main Results

The main contribution of this paper is to improve the visual quality of the marked images generated by the host network by taking into account the characteristics of HVS. Therefore, in this subsection, we have to measure the difference between the generated marked images and the ground-truth images. It is necessary that the embedded watermark information should be extracted with high fidelity for reliable ownership verification. Accordingly, the difference between the extracted watermark and the corresponding ground-truth should be analyzed as well.

We use PSNR, SSIM and MS-SSIM to quantify the visual difference between the images generated by the marked network and the ground-truth images. As shown in Fig. 1, two watermarks are embedded into the generated image. To quantify the watermarking performance, bit error rate (BER) is used for both

(a) Input   (b) Output   (c) Ground-truth   (d) Extracted W_1 (e) Extracted W_2
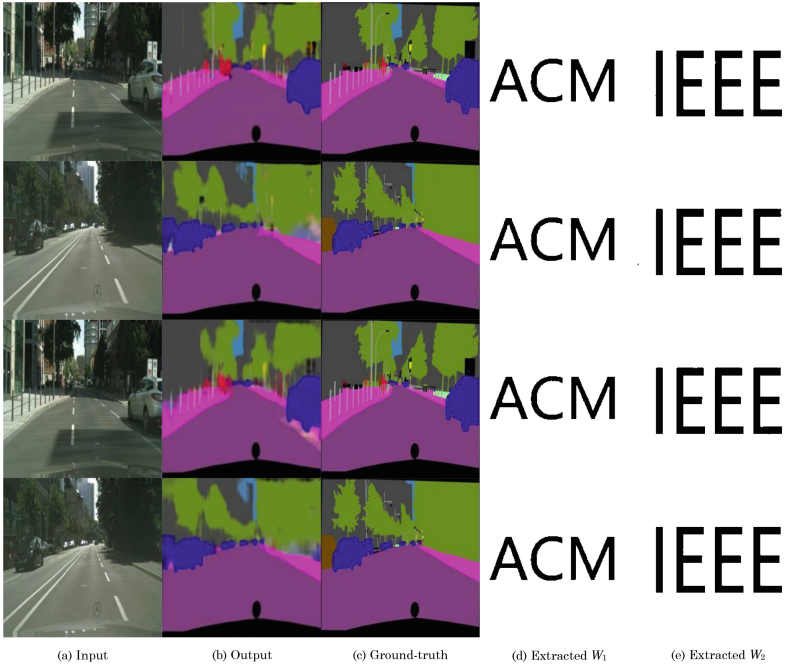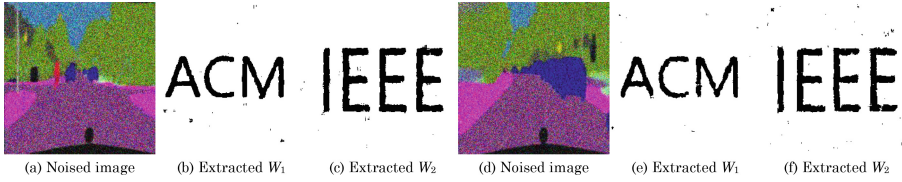
Fig. 3. Visual examples for the marked images and the extracted watermarks. The first two rows correspond to the HVS-RGB framework and the last two rows correspond to the HVS-YUV framework. Notice that, the input in the first row is identical to that in the third row, and the input in the second row is identical to that in the fourth row.

W_1 and W_2. For fair comparison, we also realize a baseline watermarking system based on [14] (without secret key), which feeds the image outputted by G directly into the watermark-extraction network to reconstruct either W_1 or W_2.

Table 2 shows the experimental results, where "mean BER_1" is used for W_1 and "mean BER_2" is used for W_2. The others are used for the images generated by the marked neural network. It is observed from Table 2 that the BER difference between the proposed strategy and the strategy in [14] is very low, which indicates that the proposed strategy does not impair the watermark fidelity. On the other hand, the proposed two frameworks significantly improve the quality of the image generated by the network. It means that the introduction of HVS is indeed helpful for enhancing the performance of generative model watermarking. Figure 3 further provides some examples, from which we can infer that the images are with satisfactory quality, which verifies the applicability.

3.4   Robustness and Ablation Study

In application scenarios, the marked images generated by the host network may be attacked. It is necessary to evaluate the robustness of the proposed two frameworks against common attacks. To deal with this problem, it is suggested to

(a) Noised image   (b) Extracted W_1 (c) Extracted W_2 (d) Noised image   (e) Extracted W_1 (f) Extracted W_2

Fig. 4. Visual examples for the noised images (marked) and the extracted watermarks. (a–c) HVS-RGB based framework and (d–f) HVS-YUV based framework. Here, $\sigma = 0.4$.
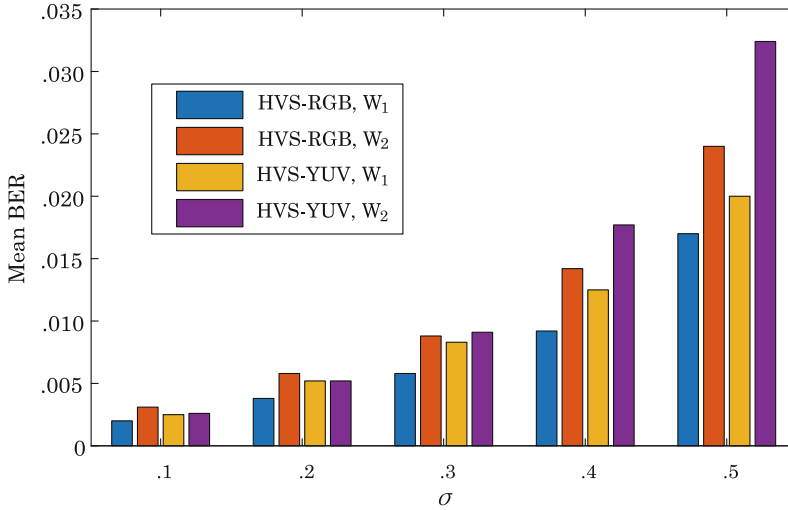
Fig. 5. The mean BERs due to different degrees of noise addition.

mimic the real-world attack to the images generated by the host network during model training, which has been proven to be effective in improving the robustness of DL models. To this end, to evaluate the robustness, during model training, we mimic the real-world attack to the images generated by the host network and feed them into the watermark-extraction network for watermark extraction. We consider one of the most representative attacks, i.e., noise addition, for simplicity.

Specifically, during model training, we add the Gaussian noise to every image generated by the host network by applying $\mu = 0$ and random $\sigma \in (0, 0.5)$. The noised image is then fed into the watermark-extraction network for watermark extraction. As a result, the trained model has the ability to resist noise addition. In this way, we are able to evaluate the robustness of the trained model by using BER. Figure 4 provides an example of noise addition, from which we can infer that the visual quality of the marked image is significantly distorted, but the embedded watermark can be extracted with satisfactory quality. Figure 5 further provides the BERs due to different degrees of noise addition. It can be observed that the BER tends to increase as the degree of noise addition

Table 3. Performance comparison by applying different loss functions. The experimental results in this table are mean values.

Method	SSIM Loss	MS-SSIM Loss	PSNR	SSIM	MS-SSIM	BER$_1$	BER$_2$
HVS-RGB			23.01	0.779	0.839	0.0029	0.0028
HVS-RGB	✓		23.05	0.801	0.846	0.0031	0.0030
HVS-RGB		✓	22.99	0.788	0.852	0.0026	0.0034
HVS-RGB	✓	✓	**23.12**	**0.807**	**0.857**	0.0024	0.0029
HVS-YUV			22.57	0.778	0.835	0.0022	0.0025
HVS-YUV	✓		22.79	0.784	0.838	0.0029	0.0034
HVS-YUV		✓	22.76	0.781	0.844	0.0026	0.0027
HVS-YUV	✓	✓	**22.83**	**0.799**	**0.852**	0.0026	0.0031

increases, which is reasonable because a larger degree of noise addition reduces the watermark information carried by the generated image, thereby resulting in a higher BER. However, overall, the BERs are in a low level. It indicates that by combining robustness enhancement strategies, the proposed two frameworks have satisfactory ability to resist against malicious attacks.

In addition, in order to achieve better visual performance, the proposed work applies structural loss to model training. In order to demonstrate its effectiveness, we analyze the experimental results on both the original task and the watermark task caused by applying different loss functions. In detail, L_8 indicates that both SSIM and MS-SSIM are used for model training. If L_8 is removed from Eq. (9) and Eq. (16), it means to skip the structural loss. By modifying L_8 as

$$\frac{1}{|S_1|} \sum_{G(x) \in S_1} (1 - \text{SSIM}(G(x), T[G(x)])),$$

it means to only use the SSIM loss. By modifying L_8 as

$$\frac{1}{|S_1|} \sum_{G(x) \in S_1} (1 - \text{MS-SSIM}(G(x), T[G(x)])),$$

it means to only use the MS-SSIM loss. Experimental results are shown in Table 3. It can be inferred from Table 3 that using SSIM and MS-SSIM as part of the entire loss function indeed has the ability to further improve the visual quality of the marked images. Meanwhile, the BER differences are all low, meaning that the introduction of structural loss will not significantly impair the watermarking performance. In summary, the proposed two frameworks are suitable for practice.

4   Conclusion and Discussion

In this paper, we propose two watermarking frameworks based on HVS for generative models that output color images as the results. Unlike the previous methods that embed a watermark into the generated image directly, the proposed two

frameworks select the more suitable channel of the image generated by the host network according to the characteristics of HVS for watermark embedding and watermark extraction. As a result, the marked image generated by the trained network has better quality compared with the previous art without impairing the watermark according to the reported experimental results. Though the structure and task of the host network are specified in our experiments, it is open for us to apply the proposed two frameworks to many other networks, indicating that this work has good universality.

On the other hand, as reported in the experimental section, we should admit that the binary watermarks may not be extracted perfectly. This is due to the reason that neural network learns knowledge from given data, which makes the neural network fall into the local optimum point that cannot perfectly model the mapping relationship between the marked image and the watermark. However, despite this, the BER can be kept very low, implying that by introducing error-correcting codes, the original watermark can be actually perfectly reconstructed.

From the viewpoint of robustness, even though augmenting the training data samples through mimicking attacks can improve the robustness of the model, real attacks are actually very complex and unpredictable. It is therefore necessary to further enhance the robustness of the model from the perspective of structural design and loss function design. It may also be very helpful by incorporating the interpretability theory of neural networks, which will be investigated in future.

References

1. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436–444 (2015)
2. Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28
3. Hinton, G., et al.: Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. IEEE Signal Process. Mag. **29**(6), 82–97 (2012)
4. Cox, I., Miller, M., Bloom, J., Fridrich, J., Kalker, T.: Digital Watermarking and Steganography. Morgan Kaufmann, Burlington (2007)
5. Uchida, Y., Nagai, Y., Sakazawa, S., Satoh, S.: Embedding watermarks into deep neural networks. In: Proceedings of the ACM International Conference on Multimedia Retrieval, pp. 269–277 (2017)
6. Li, Y., Wang, H., Barni, M.: Spread-transform dither modulation watermarking of deep neural network. J. Inf. Secur. Appl. **63**, 103004 (2021)
7. Wang, J., Wu, H., Zhang, X., Yao, Y.: Watermarking in deep neural networks via error back-propagation. In: Proceedings of the IS&T Electronic Imaging, Media Watermarking, Security, and Forensics, pp. 22-1-22-9(9) (2020)

8. Zhao, X., Yao, Y., Wu, H., Zhang, X.: Structural watermarking to deep neural networks via network channel pruning. In: Proceedings of the IEEE International Workshop on Information Forensics and Security, pp. 1–6 (2021)
9. Adi, Y., Baum, C., Cisse, M., Pinkas, B., Keshet, J.: Turning your weakness into a strength: watermarking deep neural networks by backdooring. In: Proceedings of the USENIX Conference on Security Symposium, pp. 1615–1631 (2018)
10. Zhao, X., Wu, H., Zhang, X.: Watermarking graph neural networks by random graphs. In: Proceedings of the IEEE International Symposium on Digital Forensics and Security, pp. 1–6 (2021)
11. Liu, G., Xu, T., Ma, X., Wang, C.: Your model trains on my data? Protecting intellectual property of training data via membership fingerprint authentication. IEEE Trans. Inf. Forensics Secur. **17**, 1024–1037 (2022)
12. Lin, L., Wu, H.: Verifying integrity of deep ensemble models by lossless black-box watermarking with sensitive samples. In: Proceedings of the IEEE International Symposium on Digital Forensics and Security, pp. 1–6 (2022)
13. Wu, H.: Robust and lossless fingerprinting of deep neural networks via pooled membership inference. arXiv preprint arXiv:2209.04113 (2022)
14. Wu, H., Liu, G., Yao, Y., Zhang, X.: Watermarking neural networks with watermarked images. IEEE Trans. Circuits Syst. Video Technol. **31**(7), 2591–2601 (2021)
15. Zhang, J., et al.: Model watermarking for image processing networks. In: Proceedings AAAI Conference on Artificial Intelligence, pp. 12805–12812 (2020)
16. Lewis, A.S., Knowles, G.: Image compression using the 2-D wavelet transform. IEEE Trans. Image Process. **1**(2), 244–250 (1992)
17. Levicky, D., Foris, P.: Human visual system models in digital image watermarking. Radio Eng. **13**(4), 1123–1126 (2004)
18. Barni, M., Bartolini, F., Piva, A.: Improved wavelet-based watermarking through pixel-wise masking. IEEE Trans. Image Process. **10**(5), 783–791 (2001)
19. Kutter, M., Winkler, S.: A vision-based masking model for spread-spectrum image watermarking. IEEE Trans. Image Process. **11**(1), 16–25 (2002)
20. Wu, H., Liu, G., Zhang, X.: Hiding data hiding. arXiv:2102.06826 (2021)
21. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.A.: Inception-v4, inception-ResNet and the impact of residual connections on learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 4278–4284 (2017)
22. Nair, V., Hinton, G.E.: Rectified linear units improve restricted Boltzmann machines. In: Proceedings of the International Conference on Machine Learning (2010)
23. Isola, P., Zhu, J.-Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5967–5976 (2017)
24. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. **13**(4), 600–612 (2004)
25. Wang, Z., Simoncelli, E.P., Bovik, A.C.: Multiscale structural similarity for image quality assessment. In: Proceedings of the IEEE Asilomar Conference on Signals, Systems & Computers, pp. 1398–1402 (2003)
26. Cordts, M., et al.: The cityscapes dataset for semantic urban scene understanding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3213–3223 (2016)