J. Vis. Commun. Image R. 79 (2021) 103244

Contents lists available at ScienceDirect

J. Vis. Commun. Image R.

journal homepage: www.elsevier.com/locate/jvci

Full Length Article

# GIFMarking: The robust watermarking for animated GIF based deep learning☆

Xin Liao [a,b,*], Jing Peng [a], Yun Cao [b]

[a] College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China
[b] State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China

ARTICLE INFO

*Keywords:*
Animated GIF images
Robust watermarking
3D convolutional neural networks
Adversarial network

ABSTRACT

Animated GIF has become a key communication tool in contemporary social platforms thanks to highly compatible with affective performance, and it is gradually adopted in commercial applications. Therefore, the copyright protection of the animated GIF requires more attention. Digital watermarking is an effective method to embed invisible data into a digital medium that can identify the creator or authorized users. However, few works have been devoted to robust watermarking for the animated GIF. One of the main challenges is that the animated image also contains time frame dimension information compare with still images. This paper proposes a robust blind watermarking framework based 3D convolutional neural networks for the animated GIF image, which achieves watermark image embedding and extraction for the animated GIF. Also, noise simulation is developed in frame-level to ensure robustness for the attack of the temporal dimension in this framework. Furthermore, the invisibility of the watermarked animated image is optimized by adversarial learning. Experimental results provide the effectiveness of the proposed framework and show advantages over existing works.

## 1. Introduction

Graphics interchange format (GIF) has grown to be a vital communication tool on various social networks. It is widely pursued on social media for users' online communication due to its high platform portability and vivid emotional expression. The animated GIF is a perfect combination between image and video, which is lightly weighted compared to videos and more expressive than still images. The animated GIF is created and employed by millions of Internet users every day. Under this background, many famous GIF emotions and creative cinemagraphs [1] are generated even become commercialization. At present, some websites provide users with the online creation of animated images. These operations are elementary and easy to use, which undoubtedly brings convenience. Everyone can create their unique animated image, especially artists engaged in the image art industry. Since the copyright of animated images may rely on the mainstream platform, corresponding watermarking technology also brings limitations to individual creators. It is necessary that provide a practical and suitable choice for individual animated image creators and helps to realize the real freedom of animated GIF creation. The animated GIF creators can obtain watermarking protection independent of the platform and belong to the work itself without any prior knowledge of watermarking technology. Therefore, there is an urgent need to address the copyright protection of the animated GIF image.

Watermarking is a technique to embed information into host data in a subtle manner [2]. Digital watermarking is always used for copyright protection to claim legal ownership. It plays a pivotal role in multimedia security in modern society. In general, a watermarking method should have the ability to embed the watermark into the cover image while only producing minimal distortion. Besides, the watermarked image can withstand noise attacks in the communication channel and reach copyright authentication. Robustness and invisibility are two fundamental issues in digital watermarking [3]. Robustness is the ability of the hidden message to withstand if watermarking experiences image distortion. Invisibility is the degree of perceptual transparency between the cover image and the watermarked image. Our goal aims at contributing to this growing demand for protecting ownership of the animated GIF by robust watermarking. Animated images often attack by noises in communication channels, so several typical image processing operations including Gaussian blur, Salt-and-Pepper noise, Median filtering, and JPEG compression considered in this method to gain robustness during transmission in communication channels. It is widespread to delete or replace frames of the animated

☆ This paper has been recommended for acceptance by Zicheng Liu.

* Corresponding author at: College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China.
*E-mail address:* xinliao@hnu.edu.cn (X. Liao).

https://doi.org/10.1016/j.jvcir.2021.103244
Received 22 August 2020; Received in revised form 24 June 2021; Accepted 25 July 2021
Available online 30 July 2021

image artificially while the animated image is widely used in social lives. Therefore, we specially design the frame-level noise type to withstand the image distortion in this scene.

Traditional GIF image watermarking is based on the palette image format [4–9]. Moreover, few traditional robust watermarking studies have focused on the animated GIF image. Compared with fragile watermarking, the issue of robust watermarking for the animated GIF has not received considerable attention. Due to its spatiotemporal dimension, the animated image also faces noise attacks at the frame-level, demanding higher requirements for its robust watermarking. Recent information hiding research has turned to use deep learning technology and shown a promising future. Deep learning is a strongly effective way to complete the image information hiding task. In recent years, convolutional neural networks (CNN) have made significant achievements in the field of image task [10–12], and becomes a practical choice to solve animated GIF watermarking. Hiding messages in an image by CNN as a long-term research task has yielded some remarkable research results [13–16]. This paper aims at hiding a watermark image into an animated GIF clip. Considering the increasing popularity of GIF data online, the research of GIF image robust watermarking draws a vital research topic of critical practical implication.

In this paper, we pay attention to the animated GIF, which is a valuable practical application. Blindly applying past embedding strategies to the animated image is not optimal. We propose a novel method for robust watermarking of the animated GIF image, a solution used for copyright in a real scenario. This work design a framework for embedding and extracting the watermark information, which comprises a Pre-network, an Encoder, a Decoder, and an Adv-network. The main contributions of this paper are as follows.

(1) The proposed Pre-network is effective for preprocessing the watermark. This preprocessing method automatically learns the feature of the watermark image at the frame-level by upsampling, which is fit for embedding into the animated image in the next stage.

(2) We propose a watermarking Encoder network that consists of 3D convolutional blocks inspired by U-Net [17] for hiding. To ensure the invisibility of watermarking, we add the Adv-network to minimize the watermarked image distortion.

(3) We specially design several noise attack types for animated GIF images at the frame-level, such as Frame-deletion and Frame-replacement. Training with noises in the Noise layer can gain robustness against mixed noise attacks simultaneously.

The remaining part of the paper proceeds as follows: Section 2 briefly introduces relevant research related to the GIF image watermarking methods and embedding strategies based on deep learning. Section 3 indicates our proposed framework in detail. The network architectures and the cost function are described sequentially. Section 4 demonstrates experimental settings, evaluations, comparative experiments with previous works, and ablation experiments on preprocessing methods. Finally, a conclusion is given for this work in Section 5.

## 2. Related work

Watermarking is closely related to steganography in that they both aim to hide information into an image but with a different emphasis [18]. Under the environment of the multimedia carrier of big data, the task of information hiding is also more tend diversification. Grayscale image steganography cannot meet the demand, and color image steganography research is more suitable for the development of the multimedia image. For example, the related research of multiple image steganography and colorful image steganography can satisfy the requirement of in the large data environment steganography [19–21]. The GIF animated image watermarking should also take advantage of the rapid development of multimedia carriers.

Watermarking focuses on maintaining a balance between robustness and invisibility. This technology is widely used to protect copyright, mark products, confirm the ownership, identify the authenticity, and so on [22]. Watermarking can be categorized as robust, semifragile, and fragile. Robust watermarking is a useful tool that is widely employed for the copyright protection of multimedia carriers. For instance, in [23], Hsu and Hu devised a blind watermarking scheme, which applies partly sign-altered mean and mixed modulation in the DCT-based inter-block and improves robustness by substituting a set of coefficients for a single coefficient. Wang and Du presented a DWT-based watermark embedding scheme, which considers the multi-scale 2-D discrete wavelet algorithm and combined Haar wavelet function as the wavelet function and the embedded matrix [24]. Al-Otum designed a robust color image watermarking method, which employs the interrelationship between the subbands of the primary R, G, and B color components and a two-level security procedure to achieve high color watermarking imperceptibility [25]. Generally, the robust digital watermarking must be robust enough to resist noise attacks from image processing operations in the communication channel.

GIF is the palette image format that maps between the index value and the palette reference. Thus, the watermarking algorithm for the palette image format is also applicable to the GIF image. Fridrich et al. introduced a watermarking method based optimal parity assignment for the palette. This method embeds the byte message into the image using color quantization and dithering [4]. Yang and Bao designed a watermarking scheme for the clinical brain atlas, which is the color palette image. This scheme compares the watermark bit with the parity bit for embedding [5]. Chi and Wen proposed a fragile watermarking and digital signature approach for the palette image, which selects pixels with embeddability property and uses the digital signature to compensate the embedding capacity [6]. Yang et al. devised a robust watermarking for palette image, which embeds and extracts message by adjusting the color frequencies of the nearest pixels in the image and increase capacity by grad modulation scheme [7]. Chen and Tsai demonstrated one GIF watermarking method which uses two-level color randomization in the watermark area based user-specified key [8]. Chang and Lin proposed an approach that transforms the color image from RGB to YST color and divided the T channel into non-overlapping blocks for embedding message [9]. In conclusion, most of these algorithms are fragile watermarking methods, which is less practical for the animated GIF. Some steganography algorithms can resist a few attack types as robust watermarking, such as Ezstego [26]. As far as we know, there are few robust watermarking algorithms for the animated GIF image.

Recently, researches on information hiding based CNN has emerged, and then a series of researches have been achieved, among which the most relevant one to ours is as follows. Baluja [13] presented an embedding method hiding a full-sized color image into a same-sized color image that includes preprocessing, hiding, and revealing. Lately, the adversarial network has been applied in the field of digital watermarking. Goodfellow et al. first proposed generative adversarial nets (GAN) [27], which aims at image generation task by adversarial training. The adversarial network and generator in GAN compete with each other to facilitate network training. GAN has performed excellently in computer vision and has yielded some remarkable research results [28–31]. Likewise, GAN is gradually implemented in digital watermarking. Zhu et al. first proposed a scheme based on the adversarial network to implement digital watermarking and steganography of the still image [14]. This model chooses the byte string message as watermark information, but the embedding method would lead to the limitation of the embedding capacity. Liu et al. devised a two-stage separable framework based on the adversarial network for digital watermarking, which can resist some black-box noises attack operations [15]. Hamamoto and Kawamura developed a watermarking method based on CNN, which can against rotation and JPEG compression noise attack. The attack simulator combined a rotation layer, and an additive noise layer can simulate the rotation attack and the JPEG compression attack [32]. Ahmadi et al. proposed an end-to-end diffusion watermarking model composed of two fully convolutional neural networks with residual structure, which

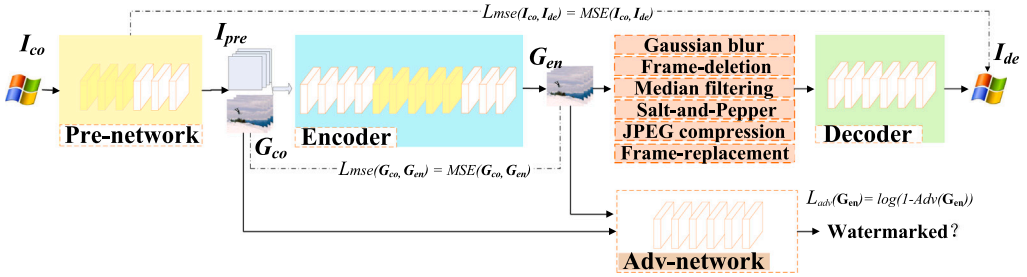


**Fig. 1.** The overall architecture of GIFMarking.

simulates various attacks as a differentiable network layer to facilitate training [33]. Luo et al. presented a watermarking framework that uses the attack network to generalize on robustness and design channel encoder to enhance the watermark redundancy for extraction accuracy. But, the proposed attack network has limitations different from noise simulation directly [34]. Yu devised a hiding framework that consists of two adversarial networks, based on the attention mask to remove perturbation of the spotlight and inconsistent loss to improve the visual quality of the stego image [35]. Nevertheless, the above methods pay attention to embed byte string message into the still image, which is limited in capacity and not proper to apply in the animated image. In summary, these researches focus on the still image and lack robustness in frame-level for the animated image.

GIF image has not received much attention compared to other formats. Existing GIF image watermarking studies are few and mainly focus on fragile watermarking. We have learned there is no prior work based deep learning that explores robust watermarking for the animated GIF image. As a result, our work aims at exploring a practical approach to this growing copyright protection requirement for the animated GIF.

## 3. The proposed framework

In this part, the description of the proposed framework is given as follows. We introduce the process of watermark embedding and extraction in Section 3.1, analyze the detailed design of this model in Section 3.2, and explain each sub-module of this framework and cost function in Sections 3.3–3.8.

### 3.1. Embedding and extraction process

We propose a robust watermarking approach for the animated GIF image in this paper. The proposed watermarking framework is mainly composed of an Encoder and a Decoder for watermark embedding and watermark extraction. Besides, the Pre-network is designed for preprocessing the watermark image, and a Noise layer to simulate the noise attacks in the real scene to gain robustness. Specifically, the framework training with mixed noise generates watermarking, which can resist existing noises in the Noise layer. In addition, the Adv-network can optimize the invisibility of generated watermarking by adversarial training. The watermarking framework is denoted as GIFMarking. Fig. 1 depicts the overall architecture of GIFMarking.

In this paper, taking the original animated image ($G_{co}$) as an example, we represent the sequential frames in the animated image as $g_{co}^{t}$, where $t$ is the ordinal number of the frame. Correspondingly, taking the original watermark image ($I_{co}$) as an example, the still single frame image is denoted as $i_{co}$.

In the preprocessing stage, the Pre-network is designed for processing the watermark image, inputs the watermark image $I_{co} = (i_{co})^{W\times H}$, and generates the preprocessed watermark image $I_{pre} = \left(i_{pre}^{1}, i_{pre}^{2}, \ldots, i_{pre}^{T}\right)^{W\times H\times T}$, where $W$, $H$ and $T$ are the width, height, and frame of the image respectively.

In the encoding stage, our goal is to hide the preprocessed watermark image $I_{pre}$ into a cover image $G_{co} = \left(g_{co}^{1}, g_{co}^{2}, \ldots, g_{co}^{T}\right)^{W\times H\times T}$ to generate watermarked image $G_{en} = \left(g_{en}^{1}, g_{en}^{2}, \ldots, g_{en}^{T}\right)^{W\times H\times T}$.

$G_{no} = \left(g_{no}^{1}, g_{no}^{2}, \ldots, g_{no}^{T}\right)^{W\times H\times T}$ represents as the watermarked image that has passed through the noise attacks in the Noise layer. In each iteration of the training process, the noise type is selected with the same probability. In this way, we gain robustness against combined noise attacks. The Noise layer is only used to simulate the image processing operations rather than a network. We denote function $f$ as an image operation in the Noise layer. The noised watermarked image $G_{no}$ attacked by the Noise layer can be defined as follows:

$$G_{no} = f\left(G_{en}\right), f \in \left\{f_1, f_2, \ldots, f_n\right\}, \tag{1}$$

we specify that $f$ is one of the noise set in the Noise layer and $n$ is the number of noise types.

In the revealing stage, our goal is to reveal the watermark image $I_{co}$ from the noised image $G_{no}$, and denote the revealed image as $I_{de} = (i_{de})^{W\times H}$. We hope that the revealed image $I_{de}$ is visually similar to the origin watermark image $I_{co}$.

The Adv-network performs adversarial training with the Encoder by judging the original image $G_{co}$ and the watermarked image $G_{en}$.

### 3.2. Model architecture

The GIFMarking framework comprises of the following modules: (1) A Pre-network, which preprocesses the watermark image for embedding in the next stage; (2) An Encoder, which hides the preprocessed watermark into an animated GIF image; (3) A Noise layer, which simulates noises to hides images robustly against a variety of image distortion; (4) A Decoder, which extracts the hidden watermark from the watermarked image. (5) An Adv-network, which is helpful to optimize the watermarked image by adversarial training.

The watermark image obtains high-capacity information and is more intuitive than the byte string message. The watermark image is a still image that only has a spatial dimension, while GIF contains spatiotemporal information. However, unlike the hiding task that encodes a still image into another still image [13]. Our work aims to hide a still image within an animated GIF in an end-to-end manner efficiently. Therefore, the previous strategy designed for the still image is not sufficient for the animated GIF image. Compared with the animated GIF, the still image contains only spatial information and lacks temporal information at the frame-level. We make the following design for the temporal dimension of the animated GIF image. In the prior computer vision tasks, feature extraction was mostly performed using two-dimensional CNN (2DCNN). 2DCNN is limited by not considering the temporal dimension, mainly used for the still image. The three-dimensional convolution kernel is a cube by stacking multiple consecutive frames. Three-dimensional CNN (3DCNN) is competent in modeling temporal information better owing to the 3D convolution kernel. For this reason, 3DCNN is more suitable for the animated GIF watermarking tasks in this paper. Hence, all models in our framework use 3DCNN. The dimension of the still image and the animated image

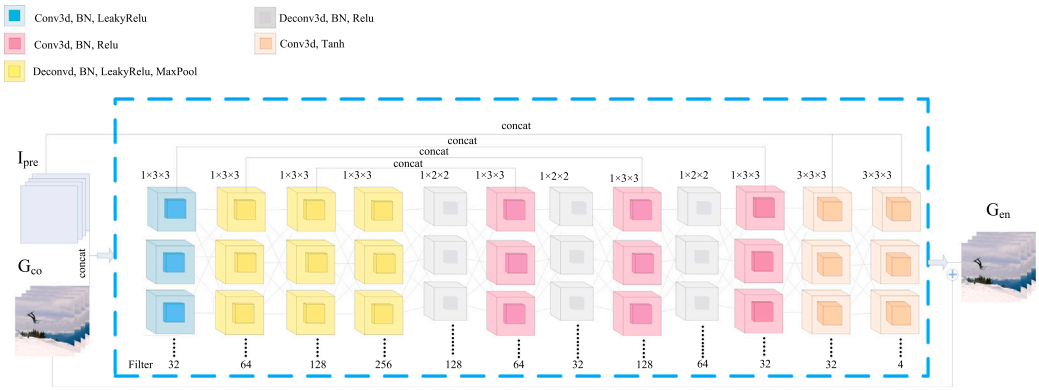


**Fig. 2.** The Encoder architecture.

**Table 1**
The Pre-network architecture.

| Index | Type | Kernel | Stride | Padding | Input,Out | Next |
|---|---|---|---|---|---|---|
| 1 | Deconv3d | (3,1,1) | (1,1,1) | (0,0,0) | (3,32) | BN,Relu |
| 2 | Deconv3d | (2,1,1) | (2,1,1) | (0,0,0) | (32,32) | BN,Relu |
| 3 | Deconv3d | (3,1,1) | (1,1,1) | (0,0,0) | (32,32) | BN,Relu |
| 4 | Conv3d | (1,3,3) | (1,1,1) | (0,1,1) | (32,64) | BN,Relu |
| 5 | Conv3d | (1,3,3) | (1,1,1) | (0,1,1) | (64,32) | BN,Relu |
| 6 | Conv3d | (1,3,3) | (1,1,1) | (0,1,1) | (32,7) | Tanh |

is different, and we implement a Pre-network to learn the watermark image pattern at different level and higher dimension that is beneficial to the hiding stage.

Deconvolution is one of the most popular methods for image upscaling in deep learning. Batch normalization (BN) [36] has been applied to faster convergence after every layer network. Our network structure is generally composed of blocks called Conv-BN-ReLU or Deconv-BN-ReLU, which stands for the combination of convolution or deconvolution, BN, and ReLU activation function [37].

### 3.3. Pre-network

In the Pre-network, we construct a 3D convolutional space to extract features of the watermark image at different levels. This model looks for learning the watermarking feature pattern that is more suitable for hiding. The watermark image is precessed by upsampling on the time frame dimension to learn features at frame-level. The Pre-network takes the watermark image $I_{co}$ as input and outputs the preprocessed image $I_{pre}$. This Pre-network is composed of three Deconv-BN-ReLU blocks, two Conv-BN-ReLU blocks, and one convolution layer followed by Tanh activation function. The architecture of the Pre-network can be seen in Table 1. The proposed Pre-network performs the required temporal upsampling and spatial downsampling simultaneously to learn features and expand the dimension of still watermark image at frame-level. Here, deconvolutional blocks that two deconvolutional layers with a kernel size $3 \times 1 \times 1$ and a deconvolutional layer with a kernel size of $2 \times 1 \times 1$ deliver upsampling in time. The spatial kernel size of all deconvolutional layers is set 1. We employ three convolutional layers with a kernel size of $1 \times 3 \times 3$ to downsample in space. The feature learning of the 3D convolutional blocks is used to learn the frame-level of the still image. Compared with other preprocessing methods for the still image, the advantages of our proposed Pre-network will be discussed in Section 4.4.

### 3.4. Encoder

The Encoder network accepts the preprocessed watermark image $I_{pre}$ and an animated GIF image $G_{co}$ as input then outputs the watermarked GIF $G_{en}$. Before inputting into the Encoder, we concatenate the same dimension in color channel when combining the preprocessed watermark and the original animated image. The Encoder network architecture is illustrated in Fig. 2 and detailed in Table 2. The U-Net [17] is an outstanding architecture for segmenting 2D microscopy images. Our Encoder structure is inspired by the shape of U-Net, which designs a contracting path formed by three convolution blocks, and an expansive path formed by three Deconv-BN-ReLU blocks to upsample feature maps. To embed watermark information redundant at any location of the GIF image, we repeatedly concatenate the preprocessed watermark and convolve to distribute watermark information in the intermediary representation. Sequentially, feature map from earlier blocks is concatenated with the preprocessed watermark image and fed to later Conv-BN-ReLU blocks. Finally, the feature map from the Encoder is concatenated with the duplicated cover image to minimize the watermarked image visible degradation.

### 3.5. Noise layer

Digital watermarking is concerned with robustness and invisibility. We insert a Noise layer between the Encoder and the Decoder to simulate image transformations in many practical scenarios. When training with the Noise layer, the model can learn to generate the watermarked GIF $G_{en}$ has robustness against combined image distortions. Considering that the watermark image could be subject to noise attack in the temporal dimension, we specially design the image processing operations at the frame-level. We consider simulating these noises as follows:

(1) Gaussian blur: Gaussian blur is a data smoothing technique in image processing. For the temporal dimension of the GIF, we use the three-dimensional Gaussian kernel and the kernel's width is set to 3. This operation can be expressed as:

$$G_{no} = kernel_{Gaussian} * G_{en}, \tag{2}$$

where $*$ represents convolution operation and $kernel_{Gaussian}$ denotes Gaussian kernel, which can be calculated as follows:

$$kernel_{Gaussian} = Gaussian(x) = \frac{e^{-(i^2+j^2+k^2)/2\sigma^2}}{2\pi\sigma^2}, \tag{3}$$

where $x$ is the pixel in row $i$ column $j$ and layer $k$ of the watermarked image $G_{en}$ $(1 < i < W; 1 < j < H; 1 < k < T)$. $W, H, T$ denote width, height, and frame of the image respectively. $e$ is the base of the natural logarithm. $\sigma$ represents standard deviation and is set to 1.

(2) Salt-and-Pepper noise: Salt-and-Pepper noise is a common noise in image operations, which is a kind of random white or black spots [38] and can be expressed as follows:

$$G_{no} : G_{en}(x) \rightarrow \left\{0, 255, G_{en}(x)\right\}, \tag{4}$$

**Table 2**
The Encoder architecture.

| Index | Type | Kernel | Stride | Padding | Input,Out | Next | Concat with |
|---|---|---|---|---|---|---|---|
| 1 | Conv3d | (1,3,3) | (1,1,1) | (0,1,1) | (4+7,32) | BN,LeakyRelu | N/A |
| 2 | Conv3d | (1,3,3) | (1,1,1) | (0,1,1) | (32,64) | BN,LeakyRelu,MaxPool | N/A |
| 3 | Conv3d | (1,3,3) | (1,1,1) | (0,1,1) | (64,128) | BN,LeakyRelu,MaxPool | N/A |
| 4 | Conv3d | (1,3,3) | (1,1,1) | (0,1,1) | (128,256) | BN,LeakyRelu,MaxPool | N/A |
| 5 | Deconv3d | (1,2,2) | (1,2,2) | (1,1,1) | (256,128) | BN,Relu | N/A |
| 6 | Deconv3d | (1,2,2) | (1,2,2) | (1,1,1) | (128,64) | BN,Relu | N/A |
| 7 | Deconv3d | (1,2,2) | (1,2,2) | (1,1,1) | (64,32) | BN,Relu | N/A |
| 8 | Deconv3d | (1,3,3) | (1,1,1) | (0,1,1) | (256,128) | BN,Relu | layer3 and layer5 |
| 9 | Deconv3d | (1,3,3) | (1,1,1) | (0,1,1) | (128,64) | BN,Relu | layer2 and layer6 |
| 10 | Conv3d | (1,3,3) | (1,1,1) | (0,1,1) | (64+7,32) | BN,Relu | N/A |
| 11 | Conv3d | (3,3,3) | (1,1,1) | (1,1,1) | (32+7,32) | BN,Relu | N/A |
| 12 | Conv3d | (3,3,3) | (1,1,1) | (1,1,1) | (32,4) | N/A | N/A |

which base on the following probability distribution,

$$P_r[G_{no}(x)]\begin{cases}\frac{2}{p}, & G_{en}(x)=0\\ \frac{2}{p}, & G_{en}(x)=255\\ 1-p, & G_{en}(x)\end{cases} \tag{5}$$

where $p$ is the proportion of the image processed by pixels, and $p$ is set to 0.01. $x$ is the pixel of the watermarked image $G_{en}$. We simulate Salt-and-Pepper noise with mean 0, and the standard deviation is 0.2.

(3) Median filtering: Median filtering is a non-linear digital filter technique, which tends to remove noise in images. Median filtering sorts the pixel values in a window area, then finds the middle value and replaces the other pixels in the window. The middle-value image can be calculated as follows:

$$G_{no}(x) = media(x(i,j), x(i+1,j), \ldots, x(i+m, j+m)), \tag{6}$$

where $1 <= i, j <= m$, $x$ is the pixel of the watermarked image $G_{en}$, m is the window size, and $media$ is used to calculate the median of the sequence of pixel values. We implement the Median filtering with a window size of 3 and used with zero paddings.

(4) JPEG compression: JPEG compression image step is composed of color channel transformation, discrete cosine transform (DCT), quantization, zigzag scanning, and coding. JPEG compression is represented as follows:

$$G_{no} = JPEG_{compression}(G_{en}), \tag{7}$$

It should be noted that the process of quantization is lossy, which is unfit for gradient-based optimization. However, Tancik [39] proposed a piecewise function with nonzero derivative almost everywhere to approximate quantization and fit the JPEG compression is differentiable. The piecewise function is described as follow:

$$q(z) = \begin{cases} z^3, & z < 0.5 \\ z, & z \geq 0.5 \end{cases}, \tag{8}$$

where $z$ is the DCT factor. When $z$ is less than 0.5, it can turn into a decimal close to 0, which simulates the process of abandoning the high-frequency coefficient in the DCT transform. And the JPEG quality is set as 80.

(5) Frame-replacement: The animated GIF image is coherent in movement, and it is often spliced to achieve new visual effects such as replacements with frames in the GIF image. To simulate Frame-replacement, we randomly select some of the original frames from the cover image to replace encoded frames in the watermarked image. The origin GIF image is expressed as $G_{co} = \{g_{co}^1, g_{co}^2, \ldots, g_{co}^T\}$, the watermarked GIF is denoted as $G_{en} = \{g_{en}^1, g_{en}^2, \ldots, g_{en}^T\}$, and $T$ is frame of the animated GIF image. Frame-replacement can be defined as follow:

$$G_{no} = \left\{g_{en}^a, g_{co}^x, g_{en}^b, g_{co}^y, \ldots, g_{en}^c\right\}, \tag{9}$$

**Table 3**
The Decoder architecture.

| Index | Type | Kernel | Stride | Padding | Input,Out | Next |
|---|---|---|---|---|---|---|
| 1 | Conv3d | (3,3,3) | (1,1,1) | (0,1,1) | (4,16) | BN,Relu |
| 2 | Conv3d | (2,3,3) | (2,1,1) | (0,1,1) | (16,32) | BN,Relu |
| 3 | Conv3d | (3,3,3) | (1,1,1) | (0,1,1) | (32,32) | BN,Relu |
| 4 | Conv3d | (3,3,3) | (1,1,1) | (1,1,1) | (32,32) | BN,Relu |
| 5 | Conv3d | (3,3,3) | (1,1,1) | (1,1,1) | (32,16) | BN,Relu |
| 6 | Conv3d | (1,1,1) | (1,1,1) | (0,0,0) | (16,3) | Tanh |

where $g_{co}^x$, $g_{co}^y$ denote the original frames from the cover image ($1 <= x, y <= T$), and $g_{en}^a, g_{en}^b, g_{en}^c$ denote encoded frames in the watermarked image ($a < b < c$, $1 <= a, b, c <= T$).

(6) Frame-deletion: Generally, the animated GIF is always clipped from videos. It is an ordinary operation to splice the temporal frame of GIF. Specific frames will be deleted from the GIF image during splicing, which may cause loss of watermark information. As a result, we consider removing some animated GIF frames to simulate frame loss and use zero vector for padding the deleted frame of the animated GIF. Frame-deletion can be represented as follow:

$$G_{no} = \left\{g_{en}^a, C_{zero}, g_{en}^b, C_{zero}, \ldots, g_{en}^c\right\}, \tag{10}$$

where $g_{en}^a, g_{en}^b, g_{en}^c$ denote encoded frames in the watermarked image ($a < b < c$, $1 <= a, b, c <= T$), and $C_{zero} = 0^{W \times H}$ represents the zero vector with the same spatial size as the animated image.

### 3.6. Decoder

The Decoder is responsible for revealing the watermark image from the watermarked image. This Decoder network takes the watermarked GIF $G_{en}$ as input and produces the revealed image $I_{de}$, which recovers watermark information as much as possible. The Decoder network structure is provided in Table 3. This model designs spatiotemporal feature extraction for revealing the watermark. The Decoder consists of five Conv-BN-ReLU blocks and one convolution layer. The output layer of the Decoder, a final convolution layer with a $1 \times 1 \times 1$ kernel, is activated using Tanh function and applies no BN.

### 3.7. Adv-network

In this paper, the Adv-network can minimize the distortion of the watermarked image. There is a competing trend between the adversarial network and the Encoder. The Adv-network judge the watermarked GIF $G_{en}$ and the original GIF $G_{co}$ to guide the Encoder generating $G_{en}$.

The visual effect of the watermarked image can be optimized by adversarial training. The structure of the Adv-network is similar to the Decoder, and the architecture of this adversarial network can be found in Table 4. Additionally, spectrum normalization (SN) [40] is employed after the final convolution layer to stabilize the training of

(a): the original GIF image

(b): the watermarked GIF image

(c): the original watermark image, and the revealed image under six noises respectivly.

(d): the original GIF image

(e): the watermarked GIF image

(f): the original watermark image, and the revealed image under six noises respectivly.

**Fig. 3.** Sample images from Dataset1. The animated GIF image is rendered as still images in frame order. The revealed watermark images from left to right are extracted under Gaussian blur, Salt-and-Pepper noise, Median filtering, JPEG compression, Frame-replacement, and Frame-deletion, respectively.

**Table 4**
The Adv-network architecture.

| Index | Type | Kernel | Stride | Padding | Input,Out | Next |
|---|---|---|---|---|---|---|
| 1 | Conv3d | (3,3,3) | (1,1,1) | (0,1,1) | (4,16) | BN,Relu |
| 2 | Conv3d | (2,3,3) | (2,1,1) | (0,1,1) | (16,32) | BN,Relu |
| 3 | Conv3d | (3,3,3) | (1,1,1) | (0,1,1) | (32,32) | BN,Relu |
| 4 | Conv3d | (3,3,3) | (1,1,1) | (1,1,1) | (32,32) | BN,Relu |
| 5 | Conv3d | (1,3,3) | (1,1,1) | (0,1,1) | (32,32) | SN |
| 6 | Linear | N/A | N/A | N/A | (32,1) | N/A |

this adversarial network. The Adv-network is trained by minimizing the cost as:

$$L(G_{co}, G_{en}) = log(1 - Adv(G_{co})) + log(Adv(G_{en})), \tag{11}$$

where $Adv$ denotes the Adv-network, $Adv(G_{co})$ and $Adv(G_{en})$ indicate inputting the original animated image and the watermarked image to the Adv-network.

### 3.8. Cost function

The watermarking task has two goals: maximizing the recovery of watermark information and minimizing cover distortion. The mean squared error (MSE) is used to measure distortion of the cover image and the watermark image. The MSE is computed using:

$$L_{mse}\left(Q, Q'\right) = \frac{1}{W \times H \times T} \sum_{i=1}^{W} \sum_{j=1}^{H} \sum_{k=1}^{T} \left\| q^{i,j,k} - q'^{i,j,k} \right\|_2^2, \tag{12}$$

where $Q = \left(q^{i,j,k}\right)^{W \times H \times T}$, $Q' = \left(q'^{i,j,k}\right)^{W \times H \times T}$ denote the sized-$W \times H \times T$ image, and $\|\cdot\|$ is the Frobenius norm.

This framework should be able to retrieve the watermark message from the watermarked image accurately despite image distortion. We train the GIFMarking by minimizing the cost function:

$$L(G_{co}, G_{en}, I_{co}, I_{de}) = L_{mse}(G_{co}, G_{en}) + \lambda L_{mse}(I_{co}, I_{de}) + \gamma L_{adv}(G_{en}), \tag{13}$$

where $L_{mse}$ denote MSE, $L_{adv}(G_{en}) = log(1 - Adv(G_{en}))$, and $\lambda$ and $\gamma$ are weight factors.

## 4. Experiments

In this section, the proposed GIFMarking is experimented on two sets of datasets and measure the invisibility performances of this model. Then, we employ the relevant state-of-the-art scheme for comparative experiments and analysis the advantage of the proposed method based on the experimental results. Finally, an ablation experiment about preprocessing the watermark image is conducted.

**Table 5**
Visual evaluation of GIFMarking testing under six noises respectively based on Dataset1.

| Noise | $G_{en}$ | | | $I_{de}$ | | | |
|---|---|---|---|---|---|---|---|
| | PSNR | SSIM | RMSE | PSNR | SSIM | RMSE | VIF |
| Gaussian blur | 46.37 | 0.9738 | 1.2247 | 34.30 | 0.9693 | 4.9152 | 0.7861 |
| Salt-and-Pepper | 46.37 | 0.9738 | 1.2247 | 33.90 | 0.9539 | 5.1468 | 0.7556 |
| Median filtering | 46.37 | 0.9738 | 1.2247 | 32.81 | 0.9415 | 5.8350 | 0.7161 |
| JPEG compression | 46.37 | 0.9738 | 1.2247 | 34.63 | 0.9741 | 4.7320 | 0.7864 |
| Frame-replacement | 46.37 | 0.9738 | 1.2247 | 34.15 | 0.9684 | 5.0008 | 0.7816 |
| Frame-deletion | 46.37 | 0.9738 | 1.2247 | 33.48 | 0.9634 | 5.4018 | 0.7671 |

### 4.1. Training details

The proposed GIFMarking experiment on two sets of different datasets while evaluating this approach's performance. The animated GIF datasets are *TGIF* [41], and cinemagraphy images that are crawled from the web site *https://giphy.com/expl-ore/cinemagraph*. The available cinemagraphy images are limited. In order to expand this cinemagraph dataset, two spatiotemporal sizes of 256 × 256 × 8 patches are randomly cut out from each animated image after flipping vertically and horizontally.

For each GIF image, we randomly clip eight adjacent frames as a cover image. Considering the practical situation which confirms the animated image product owner of a brand, we adopt color loge image as the watermark image. Therefore, a set of watermark images are picked from *Logo-2K+* [42]. Also, another watermark dataset randomly chooses from *COCO* [43]. We use *TGIF* and *Logo-2K+* as a group of datasets, denote this group of the dataset as Dataset1. Cinematography and *COCO* as another group of datasets is denoted as Dataset2. We randomly split training/testing subsets, with 10,000, and 2500 cover-watermark image pairs. All watermark images are resized to a spatial size of 256 × 256, and all cover images are reprocessed into animated GIF images with a size of 256 × 256 × 8. Taking the eight-frame GIF image as an example, the frames replaced or deleted in the Frame-replacement and Frame-deletion noise operation is set to 2. The proposed GIFMarking is trained iteratively using adam algorithm [44] ($\beta 1 = 0.5$, $\beta 2 = 0.999$) with an initial learning rate of $10^{-4}$. The training objective is to minimize the cost function, and we set weight factors $\lambda = 0.8$, and $\gamma = 0.01$. Due to the high memory overload, using batch size = 1 in the process of training. All experiments in this paper are implemented in Python with Pytorch [45], on an NVIDIA GeForce RTX 2080 Ti GPU.

We evaluate the proposed framework on visual similarity and quality loss of the watermark image and the watermarked GIF by peak signal to noise ratio (PSNR), structural similarity (SSIM) [46], visual information fidelity (VIF) [47] and root mean square error (RMSE).

MSE is the common basis measure in image quality, which is described in Eq. (12). PSNR is an MSE-based image quality measure, which approximates the estimation for reconstruction's human perception. The higher the PSNR, the greater the similarity between the two images. Generally, the PSNR is more useful in image quality measures than MSE, and PSNR can be defined as:

$$PSNR = 10\log_{10}\left(\frac{L^2}{MSE}\right), \tag{14}$$

where $L$ is the maximum value in the image data, which usually is 255.

SSIM is motivated by the human visual system, as a highly effective image fidelity measure. It is designed by modeling any image distortion as a combination of three factors that are loss of correlation, luminance distortion, and contrast distortion. Given two same size image $X$ and $Y$, SSIM can be described as:

$$SSIM(X,Y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2\mu_y^2 + C_1)(\sigma_x^2\sigma_y^2 + C_2)}, \tag{15}$$

where $\mu_x$, $\mu_y$, are the means, $\sigma_x$, $\sigma_y$ are the standard deviations , and $\sigma_{xy}$ is the cross-covariance of X and Y respectively. $C_1$ and $C_2$, are constants used to avoid a null denominator.

VIF is based on natural scene statistics model and uses Gaussian scale mixtures (GSM) to simulate natural images in the wavelet domain, which via capture three types of distortion blur, additive noise, and global or local contrast changes. VIF ranges from 0 to 1, which a higher value means greater image quality.

$$VIF = \frac{\sum_{j\in subband\,s} I(\vec{C}^{N,j}; \vec{F}^{N,j}|s^{N,j})}{\sum_{j\in subband\,s} I(\vec{C}^{N,j}; \vec{E}^{N,j}|s^{N,j})}, \tag{16}$$

where $I(\vec{C}^{N,j}; \vec{F}^{N,j}|s^{N,j})$ and $I(\vec{C}^{N,j}; \vec{E}^{N,j}|s^{N,j})$ denote the reference and distorted image information respectively, $\vec{C}^{N,j}$ represents $N$ elements of the GSM models of the subband $j$ of wavelet decomposition, $E$ and $F$ are the visual output of the reference and the test images after the HVS model, $s$ represents the scale number of the image.

RMSE as deformation of MSE is also a commonly used evaluation. RMSE is the square root of the MSE, a type of error measuring between two images based MSE, which can be defined as:

$$RMSE = \sqrt{MSE}, \tag{17}$$

### 4.2. Experimental results

We train the proposed framework in which six types of noises are combined in the Noise layer and then test under six noise types, respectively. Specifically, as shown in Figs. 3–4, the example images display two batches of the original GIF image, the watermarked GIF image, the original watermark image, and the revealed image under six noises respectively. Among them, the animated images are divided into frames, which displays as single-frame images. The row of sequential revealed images extracted under Gaussian blur, Salt-and-Pepper noise, Median filtering, JPEG compression, Frame-replacement, and Frame-deletion, respectively.

The experimental results of GIFMarking on Dataset1, as reported in Table 5. The PSNR and SSIM between the original GIF image and the watermarked GIF image can reach (46.37 dB, 0.9738) when tested under different noise types. The PSNR and SSIM between the original watermark and the revealed watermark image achieve superior performance (34.63 dB, 0.9741) when tested on the JPEG compression. The visual effect of the revealed watermark image attacked by Median filtering can achieve 32.81 dB of PSNR and 0.9415 of SSIM, which is slightly worse than other noise types, but still achieves less visual loss than the original watermark image. The sample images from Dataset1 are displayed in Fig. 3. Table 6 demonstrates the results of GIFMarking on Dataset2. This model PSNR and SSIM between the original GIF image and the watermarked GIF image can reach (47.15 dB, 0.9823). When tested on the JPEG compression, the visual performance between the original watermark and the revealed watermark image achieve superior (34.92 dB, 0.9601). The example images from Dataset2 are shown in Fig. 4. We have to admit that the extracted watermark image is slightly inferior in visual performance to the watermarked image because the high-quality watermarked image is still our primary mission. The revealed watermark images can easily identify the LOGO image's annotation.

(a): the original GIF image

(b): the watermarked GIF image

(c): the original watermark image, and the revealed image under six noises respectivly

(d): the original GIF image

(e): the watermarked GIF image

(f): the original watermark image, and the revealed image under six noises respectivly

**Fig. 4.** Sample images from Dataset2. The animated GIF image is rendered as still images in frame order. The revealed watermark images from left to right are extracted under Gaussian blur, Salt-and-Pepper noise, Median filtering, JPEG compression, Frame-replacement, and Frame-deletion, respectively.

**Table 6**
Visual evaluation of GIFMarking testing under six noises respectively based on Dataset2.

| Noise | $G_{en}$ | | | $I_{de}$ | | | |
|---|---|---|---|---|---|---|---|
| | PSNR | SSIM | RMSE | PSNR | SSIM | RMSE | VIF |
| Gaussian blur | 47.15 | 0.9823 | 1.1195 | 34.80 | 0.9674 | 4.6402 | 0.7659 |
| Salt-and-Pepper | 47.15 | 0.9823 | 1.1195 | 33.95 | 0.9550 | 5.1173 | 0.7423 |
| Median filtering | 47.15 | 0.9823 | 1.1195 | 31.48 | 0.9182 | 6.8005 | 0.7089 |
| JPEG compression | 47.15 | 0.9823 | 1.1195 | 34.92 | 0.9601 | 4.5766 | 0.7683 |
| Frame-replacement | 47.15 | 0.9823 | 1.1195 | 34.85 | 0.9554 | 4.6136 | 0.7502 |
| Frame-deletion | 47.15 | 0.9823 | 1.1195 | 32.95 | 0.9511 | 5.7417 | 0.7464 |

**Table 7**
Comparison of GIFMarking, Baluja [13] and Hidden [14] without the Noise layer.

| Method | $G_{en}$ | | | $I_{de}$ | | | |
|---|---|---|---|---|---|---|---|
| | PSNR | SSIM | RMSE | PSNR | SSIM | RMSE | VIF |
| Baluja [13] | 42.07 | 0.9831 | 2.0093 | 32.54 | 0.8769 | 6.0192 | 0.6316 |
| Hidden [14] | 42.13 | 0.9885 | 1.9954 | 31.75 | 0.8643 | 6.5923 | 0.6138 |
| GIFMarking | 46.29 | 0.9891 | 1.2361 | 33.89 | 0.9558 | 5.1528 | 0.7652 |

### *4.3. Comparison experiments*

There is no previous method exploring hiding information in the animated GIF by deep learning. The available state-of-the-art method Baluja [13] becomes a potential object for comparison, which is the same as ours to choose an image embedding into another image by CNN technique. Also, the most representative previous study in watermarking based on CNN Hidden [14] embeds the byte string into

**Table 8**
Comparison of GIFMarking, Baluja [13] and Hidden [14] with the Noise layer.

| Noise | Method | $G_{en}$ | | | $I_{de}$ | | | |
|---|---|---|---|---|---|---|---|---|
| | | PSNR | SSIM | RMSE | PSNR | SSIM | RMSE | VIF |
| Gaussian blur | Baluja [13] | 42.56 | 0.9669 | 1.8991 | 31.16 | 0.8786 | 7.0557 | 0.6289 |
| | Hidden [14] | 42.24 | 0.9808 | 1.9703 | 30.38 | 0.8258 | 7.7186 | 0.5665 |
| | GIFMarking | 47.10 | 0.9813 | 1.1260 | 33.18 | 0.9406 | 5.5917 | 0.7284 |
| Salt-and-Pepper | Baluja [13] | 42.56 | 0.9669 | 1.8991 | 32.18 | 0.8689 | 6.2739 | 0.6211 |
| | Hidden [14] | 42.24 | 0.9808 | 1.9703 | 30.62 | 0.8331 | 7.5083 | 0.5675 |
| | GIFMarking | 47.10 | 0.9813 | 1.1260 | 33.50 | 0.9263 | 5.3894 | 0.7068 |
| Median filtering | Baluja [13] | 42.56 | 0.9669 | 1.8991 | 30.02 | 0.8367 | 8.0453 | 0.5951 |
| | Hidden [14] | 42.24 | 0.9808 | 1.9703 | 29.78 | 0.8113 | 8.2707 | 0.5542 |
| | GIFMarking | 47.10 | 0.9813 | 1.1260 | 30.65 | 0.8978 | 7.4824 | 0.6482 |
| JPEG compression | Baluja [13] | 42.56 | 0.9669 | 1.8991 | 32.70 | 0.8859 | 5.9094 | 0.6426 |
| | Hidden [14] | 42.24 | 0.9808 | 1.9703 | 31.21 | 0.8563 | 7.0152 | 0.5956 |
| | GIFMarking | 47.10 | 0.9813 | 1.1260 | 35.31 | 0.9620 | 4.3756 | 0.7742 |
| Frame-replacement | Baluja [13] | 42.56 | 0.9669 | 1.8991 | — | — | — | — |
| | Hidden [14] | 42.24 | 0.9808 | 1.9703 | — | — | — | — |
| | GIFMarking | 47.10 | 0.9813 | 1.1260 | 34.40 | 0.9627 | 4.8589 | 0.7283 |
| Frame-deletion | Baluja [13] | 42.56 | 0.9669 | 1.8991 | — | — | — | — |
| | Hidden [14] | 42.24 | 0.9708 | 1.9703 | — | — | — | — |
| | GIFMarking | 47.10 | 0.9813 | 1.1260 | 33.77 | 0.9684 | 5.2244 | 0.7221 |

— denotes extraction failed.

**Table 9**
Comparison of GIFMarking, Baluja [13] and Hidden [14] of embedding eight watermark images into an animated GIF.

| Method | $G_{en}$ | | | $I_{de}$ | | | |
|---|---|---|---|---|---|---|---|
| | PSNR | SSIM | RMSE | PSNR | SSIM | RMSE | VIF |
| Baluja [13] | 32.81 | 0.8606 | 5.8350 | 32.00 | 0.8607 | 6.4053 | 0.6210 |
| Hidden [14] | 35.81 | 0.8895 | 4.1309 | 33.33 | 0.9150 | 5.4959 | 0.6812 |
| GIFMarking | 40.03 | 0.9329 | 2.5412 | 35.40 | 0.9599 | 4.3305 | 0.7880 |

**Table 10**
Visual evaluation of the preprocessing method Zero-padding.

| Noise | $G_{en}$ | | | $I_{de}$ | | | |
|---|---|---|---|---|---|---|---|
| | PSNR | SSIM | RMSE | PSNR | SSIM | RMSE | VIF |
| Gaussian blur | 39.81 | 0.9723 | 2.6064 | 30.75 | 0.8912 | 7.3967 | 0.6404 |
| Salt-and-Pepper | 39.81 | 0.9723 | 2.6064 | 31.93 | 0.8788 | 6.4571 | 0.6372 |
| Median filtering | 39.81 | 0.9723 | 2.6064 | 28.38 | 0.8321 | 9.7172 | 0.5733 |
| JPEG compression | 39.81 | 0.9723 | 2.6064 | 30.92 | 0.8815 | 7.2534 | 0.6327 |
| Frame-replacement | 39.81 | 0.9723 | 2.6064 | 30.82 | 0.8883 | 7.3373 | 0.6336 |
| Frame-deletion | 39.81 | 0.9723 | 2.6064 | 30.09 | 0.8776 | 7.9807 | 0.6174 |

**Table 11**
Visual evaluation of the preprocessing method Image-reproduction.

| Noise | $G_{en}$ | | | $I_{de}$ | | | |
|---|---|---|---|---|---|---|---|
| | PSNR | SSIM | RMSE | PSNR | SSIM | RMSE | VIF |
| Gaussian blur | 39.67 | 0.9731 | 2.6487 | 30.55 | 0.8741 | 7.5690 | 0.6086 |
| Salt-and-Pepper | 39.67 | 0.9731 | 2.6487 | 30.69 | 0.8414 | 7.4480 | 0.5874 |
| Median filtering | 39.67 | 0.9731 | 2.6487 | 28.36 | 0.8211 | 9.7396 | 0.5257 |
| JPEG compression | 39.67 | 0.9731 | 2.6487 | 31.16 | 0.8686 | 7.0557 | 0.6189 |
| Frame-replacement | 39.67 | 0.9731 | 2.6487 | 30.07 | 0.8619 | 7.9991 | 0.5801 |
| Frame-deletion | 39.67 | 0.9731 | 2.6487 | 29.67 | 0.8589 | 8.3761 | 0.5687 |

a still image, which is different from the embedding and extracting way with the color images. We changed the embedding and revealing mode to adapt the watermark images and retained the model without other changes. To measure the experimental effect with the previous works, we train the GIFMarking with and without the Noise layer respectively for comparison experiments. It is noted that the previous works are embedding into a still image that is not specially designed for the animated GIF. To ensure the same embedding capacity as ours, we reimplement a fixed frame of GIF as cover during training Hidden and Baluja. Besides that, we apply the Noise layer behind the hiding network in Baluja to compare with ours. We tried to implement these schemes using the same experimental setup as in the Baluja. Hence, the watermark image dataset used ImageNet [48] here, which is consistent with the original of Baluja. These hyperparameters are used: learning rate: $10^{-3}$, optimizer: adam, batch size: 25, and epochs: 50.

In Table 7, we report the performance comparison of the GIFMarking, Hidden and Baluja without the Noise Layer. It can be observed that our method offers a clear superiority of the watermarked image. In particular, we designed frame-level noise attacks for the animated image, a challenge for Baluja applying with the Noise layer. The corresponding frame containing watermark information may be deleted if Frame-deletion and Frame-replacement are set in the Noise layer. Consequently, the watermark information will be at risk of being lost. When such a situation occurs in the training process, it will not be easy to reveal the watermark image correctly. While training nearly 30 epochs, Hidden and Baluja with the Noise layer are still unable to

extract the watermark image. Therefore, we only set Gaussian blur, Salt-and-Pepper noise, Median filtering, and JPEG compression, which are not frame-level noise apply in Hidden and Baluja with the Noise layer. The results of the models with the Noise layer test under six noise attacks are shown in Table 8. The result shows that our model performs few distortions for both the watermarked GIF and the revealed watermark. It can be seen that, compared with the existing embedding method Hidden and Baluja, the proposed GIFMarking significantly outperforms even attacked by the frame-level noises.

The payload of the above comparative experiment is embedding a still watermark image into an animated image. The comparative experiments are aimed at the still image, which leads to application in the animated image that can only focus on embedded in a single frame. Furthermore, we add an experiment of embedding eight still watermark images in an animated image, which makes the comparative methods embed instead of being limited to a single frame. For the comparative methods, each still watermark image is embedded into each frame of the animated image by using the trained model. The watermark image data set is expanded for embedding different watermark images into each frame. To make the proposed method suitable for this embedding rate, we need to remove the Pre-network and set the padding of the Decoder to avoid downsampling. Table 9 demonstrates the performance comparison of the GIFMarking, Hidden, and Baluja embedding eight watermark images into an animated GIF. The result shows that the proposed method has better visual performance on the watermarked GIF image and the decoded watermark. The proposed method is more suitable for the carrier of the animated GIF image compared with the previous strategy. It is essential to point out that our method treats the animated image as a whole rather than just a set of single frames, which is useful for designing to gain robustness at frame-level.

### 4.4. Ablation experiment on the preprocessing methods

In this paper, we design a Pre-network to preprocess the watermark image. The Pre-network accepts the watermark image $I_{co}$ and outputs $I_{pre}$, which is consistent with the animated GIF in the temporal dimension. The cover image should be concatenated with the watermark image before inputting the Encoder. Accordingly, the preprocessing method is required to expand the sized-$W \times H$ watermark image into the sized-$W \times H \times T$ image. Especially, the Pre-network achieve the feature extraction of the watermark image in different level and upsample the temporal dimension of the watermark image. To verify the Pre-network performance, we conduct an ablation experiment on the preprocessing methods for the still watermark image. There are intuitive expanding methods for still images, such as filling with zero vectors and copying the original image directly multiple times. This proposed method is compared with the following processing methods:

(1) Zero-padding: The watermark image is preprocessed by filling zero vector, which can be defined as:

$$I_{pre} = \left\{ i_{co}, C_{zero}, C_{zero}, \ldots, C_{zero} \right\}, \tag{18}$$

where $C_{zero} = 0^{W \times H}$ denotes the zero vector and padding $T-1$ times, $T$ is the frame of the GIF image.

(2) Image-reproduction: Copying the watermark image multiple times for preprocessing, which can be expressed as:

$$I_{pre} = \left\{ i_{co}, i_{co}, i_{co}, \ldots, i_{co} \right\}, \tag{19}$$

where copy the original watermark image $T-1$ times and fill after the image.

We conduct an ablation experiment about the Pre-network by replacing the preprocessing method in GIFMarking. The above preprocessing methods experiment on the Dataset2. The preprocess method of the Zero-padding result is shown in Table 10, and the visual quality of the watermarked image can reach PSNR 39.81 and SSIM 0.9723. The result of the Image-reproduction method is demonstrated in Table 11, and the visual quality of the watermarked image can reach PSNR 39.67 and SSIM 0.9731. Experimental results compared with ours in Table 6, show that the Pre-network is superior to other preprocessing methods and can improve the visual quality of the watermarked image. Among them, the Pre-network expands the time dimension of the watermark image by up-sampling and learn the feature that is more suitable for the embedding pattern under the training of the whole framework. It is obvious that the Pre-network is helpful for achieving preprocess the watermark image, which can improve the visual quality of the watermarked image.

## 5. Conclusion

In this paper, we propose a robust watermarking method for animated GIF images, which hides images robustly against several common image distortions. The experiments demonstrate that this model has an excellent visual performance on the watermarked GIF image and is able to decode the watermark image while gaining robustness against mixed noise. Also, the Adv-network can optimize the invisibility of the watermarked image. In the comparative experiment, our model achieves superior performance for the watermarked image. Finally, we conduct ablation experiments on the preprocessing method and show the Pre-network achieves exceptional in processing still watermark image. Furthermore, the proposed method has the potential to become a competitive choice for copyright protection of the animated image product, which can apply in real life.

In the future, the trend of animated GIF watermarking can survive a variety of noise attacks will become the direction of our efforts. We hope to explore more robustness at the frame-level to improve security while maintaining the high-quality watermarking image.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgments

This work is supported by National Natural Science Foundation of China (Grant Nos. 61972142, 61872356, 61772191), Hunan Provincial Natural Science Foundation of China (Grant No. 2020JJ4212), Key Lab of Forensic Science, Academy of Forensic Science, Ministry of Justice, China (Grant No. KF202118).

## References

[1] J. Tompkin, F. Pece, K. Subr, et al., Towards moment imagery: automatic Cinemagraphs, in: Proceedings of IEEE Conference on Visual Media Production, 2011, pp. 87-93.
[2] I. Cox, M. Miller, J. Bloom, et al., Digital watermarking, in: Proceedings of International Workshop on Digital-Forensics and Watermarking, 2005, pp. 15-17.
[3] M.D. Swanson, B. Zhu, A.H. Tewfik, Transparent robust image watermarking, in: Proceedings of IEEE International Conference on Image Processing, 1996, pp. 211-214.
[4] J. Fridrich, M. Goljan, D. Rui, Lossless data embedding for all image formats, in: Proceedings of SPIE, Security and Watermarking of Multimedia Contents, 2002, pp. 572-583.
[5] Y. Yang, F. Bao, An invertible watermarking schemes for authentication of electronic clinical brain atlas, in: Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, Vol. 3, 2003, pp. III-533.
[6] H. Chi, H. Wen, A combined approach to integrity protection and verification of palette images using fragile watermarks and digital signatures, IEICE Trans. Fund. Electron. Commun. E87-A (6) (2004) 1612–1619.
[7] C. Yang, Y. Yang, M. Cai, Valid digital watermarking of palette-based image, J. Acta Sci. Nat. Univ. Sunyatseni 43 (2) (2004) 128–131.
[8] P. Chen, W. Tsai, Copyright protection of palette images by a robust lossless visible watermarking technique, in: The Fifth Workshop on Digital Archives, 2005.
[9] C. Chang, P. Lin, A color image authentication method using partitioned palette and morphological operations, IEICE Trans. Inf. Syst. 91 (1) (2008) 54–61.

[10] D. Chao, C. Chen, K. He, et al., Learning a deep convolutional network for image super-resolution, in: Proceedings of European Conference on Computer Vision, 2014, pp. 184-199.
[11] K. He, X. Zhang, S. Ren, Deep residual learning for image recognition, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770-778.
[12] K. He, G. Gkioxari, D. Piotr, et al., Mask R-CNN, IEEE Trans. Pattern Anal. Mach. Intell. (2018) 1.
[13] S. Baluja, Hiding images in plain sight: deep steganography, in: Proceedings of Conference on Neural Information Processing Systems, 2017, pp. 2069-2079.
[14] J. Zhu, R. Kaplan, J. Johnson, et al., HiDDeN: hiding data with deep networks, in: Proceedings of European Conference on Computer Vision, Vol. 11219, (1) 2018, pp. 682-697.
[15] Y. Liu, M. Guo, J. Zhang, et al., A novel two-stage separable deep learning framework for Practical Blind Watermarking, in: Proceedings of International Conference on Multimedia, 2019, pp. 1509-1517.
[16] R. Zhang, S. Dong, J. Liu, Invisible steganography via generative adversarial network, Multimedia Tools Appl. 78 (7) (2019) 8559–8575.
[17] O. Ronneberger, P. Fischer, T. Brox, U-Net: convolutional networks for biomedical image segmentation, in: Proceedings of International Conference Medical Image Computing and Computer-Assisted Intervention, Vol. 9351, 2015, pp. 234-241.
[18] I. Cox, M. Miller, J. Bloom, et al., Digital watermarking and steganography, in: The Morgan Kaufmann Series in Multimedia Information and Systems, 2007.
[19] J. Yang, X. Liao, An embedding strategy on fusing multiple image features for data hiding in multiple images, J. Vis. Commun. Image Represent. 71 (2020) 102822.
[20] X. Liao, J. Yin, M. Chen, et al., Adaptive payload distribution in multiple images steganography based on image texture features, IEEE Trans. Dependable Secure Comput. (2020) http://dx.doi.org/10.1109/TDSC.2020.3004708.
[21] X. Liao, Y. Yu, B. Li, et al., A new payload partition strategy in color image steganography, IEEE Trans. Circuits Syst. Video Technol. 30 (3) (2020) 685–696.
[22] N. Nikolaidis, I. Pitas, Copyright protection of images using robust digital signatures, in: Proceedings of International Conference on Acoustics, Speech and Signal Processing, 1996.
[23] L. Hsu, H. Hu, Robust blind image watermarking using crisscross inter-block prediction in the DCT domain, J. Vis. Commun. Image Represent. 46 (2017) 33–47.
[24] J. Wang, Z. Du, A method of processing color image watermarking based on the haar wavelet, J. Vis. Commun. Image Represent. 64 (2019) 102627.
[25] H.M. Al-Otum, Secure and robust host-adapted color image watermarking using inter-layered wavelet-packets, J. Vis. Commun. Image Represent. 66 (2020) 102726.
[26] R. Machado, Ezstego, 1997, http://ezstego.com.
[27] I. Goodfellow, J.P. Abadie, M. Mirza, et al., Generative adversarial nets, in: Proceedings of Conference on Neural Information Processing Systems, 2014, pp. 2672-2680.
[28] X. Mao, Q. Li, H. Xie, et al., Least squares generative adversarial networks, in: Proceedings of IEEE International Conference on Computer Vision, 2017, pp. 2813-2821.
[29] H. Zhang, I. Goodfellow, D. Metaxaset, et al., Self-attention generative adversarial networks, in: Proceedings of Machine Learning Research, 2018, pp. 7754-7363.
[30] G. Qi, Loss-sensitive generative adversarial networks on lipschitz densities, Int. J. Comput. Vis. 128 (5) (2020) 1118–1140.
[31] T. Karras, S. Laine, T. Aila, A style-Based generator architecture for generative adversarial networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2020, pp. 4401-4410.
[32] I. Hamamoto, M. Kawamura, Neural watermarking method including an attack simulator against rotation and compression attacks, IEICE Trans. Inf. Syst. 103 (2020) 33–41.
[33] M. Ahmadi, A. Norouzi, N. Karimi, et al., Redmark: Framework for residual diffusion watermarking based on deep networks, Expert Syst. Appl. 146 (2020) 113157.
[34] X. Luo, R. Zhan, H. Chang, et al., Distortion agnostic deep watermarking, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Vol. 57, 2020, pp. 13545-13554.
[35] C. Yu, Attention based data hiding with generative adversarial networks, in: Proceedings of AAAI Conference on Artificial Intelligence, 2020, pp. 1120-1128.
[36] S. Ioffe, C. Szegedy, Batch normalization: accelerating deep network training by reducing internal covariate shift, in: Proceedings of IEEE International Conference on Machine Learning, Vol. 37, 2015, pp. 448-456.
[37] K. He, X. Zhang, S. Ren, et al., Delving deep into rectifiers: surpassing human-level performance on imagenet classification, in: Proceedings of IEEE International Conference on Computer Vision, 2015, pp. 1026-1034.
[38] H. Liu, Y. Li, Y. Zhou, et al., Impulsive noise suppression in the case of frequency estimation by exploring signal sparsity, Digit. Signal Process. 57 (2016) 34–45.
[39] M. Tancik, B. Mildenhall, Ren Ng, StegaStamp: invisible hyperlinks in physical photographs, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2020, pp. 2117-2126.
[40] T. Miyato, T. Kataoka, M. Koyama, et al., Spectral normalization for generative adversarial networks, in: Proceedings of International Conference on Learning Representations, 2018.
[41] Y. Li, Y. Song, L. Cao, et al., TGIF: a new dataset and benchmark on animated GIF description, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 4641-4650.
[42] J. Wang, W. Min, S. Hou, et al., Logo-2K+: a large-Scale Logo dataset for scalable Logo classification, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, (4) 2020, pp. 6194-6201.
[43] T.Y. Lin, M. Maire, S. Belongie, et al., Microsoft COCO: common objects in context, in: Proceedings of European Conference on Computer Vision, Vol. 8693, 2014, pp. 740-755.
[44] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, in: Proceedings of International Conference on Learning Representations, 2015.
[45] A. Paszke, S. Gross, S. Chintala, et al., Automatic differentiation in pytorch, in: Proceedings of Conference on Neural Information Processing Systems, 2017.
[46] Z. Wang, A.C. Bovik, H.R. Sheikh, et al., Image quality assessment: from error visibility to structural similarity, IEEE Trans. Image Process. 13 (4) (2004) 600–612.
[47] H.R. Sheikh, A.C. Bovik, Image information and visual quality, IEEE Trans. Image Process. 15 (2) (2006) 430–444.
[48] O. Russakovsky, J. Deng, H. Su, et al., Imagenet large scale visual recognition challenge, Int. J. Comput. Vis. 115 (3) (2015) 211–252.