Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.052745

ARTICLE

# MarkINeRV: A Robust Watermarking Scheme for Neural Representation for Videos Based on Invertible Neural Networks

**Wenquan Sun[1,2], Jia Liu[1,2,*], Lifeng Chen[1,2], Weina Dong[1,2] and Fuqiang Di[1,2]**

[1]Department of Cryptographic Engineering, Engineering University of PAP, Xi'an, 710086, China

[2]Department of Cryptographic Engineering, Key Laboratory for Cryptology and Information Security, Xi'an, 710086, China

*Corresponding Author: Jia Liu. Email: liujia1022@gmail.com

Received: 13 April 2024 Accepted: 14 July 2024 Published: 12 September 2024

## ABSTRACT

Recent research advances in implicit neural representation have shown that a wide range of video data distributions are achieved by sharing model weights for Neural Representation for Videos (NeRV). While explicit methods exist for accurately embedding ownership or copyright information in video data, the nascent NeRV framework has yet to address this issue comprehensively. In response, this paper introduces MarkINeRV, a scheme designed to embed watermarking information into video frames using an invertible neural network watermarking approach to protect the copyright of NeRV, which models the embedding and extraction of watermarks as a pair of inverse processes of a reversible network and employs the same network to achieve embedding and extraction of watermarks. It is just that the information flow is in the opposite direction. Additionally, a video frame quality enhancement module is incorporated to mitigate watermarking information losses in the rendering process and the possibility of malicious attacks during transmission, ensuring the accurate extraction of watermarking information through the invertible network's inverse process. This paper evaluates the accuracy, robustness, and invisibility of MarkINeRV through multiple video datasets. The results demonstrate its efficacy in extracting watermarking information for copyright protection of NeRV. MarkINeRV represents a pioneering investigation into copyright issues surrounding NeRV.

## KEYWORDS

Invertible neural network; neural representations for videos; watermarking; robustness

## 1 Introduction

Implicit Neural Representation (INR) [1–3] emerges as a novel paradigm for parameterizing diverse signals encompassing images, audio, video, and 3D models. This methodology entails representing the signal as a continuous function, mapping the signal's domain to attribute values at respective coordinates, known as coordinate-based representation. Post-training, INR weights facilitate various tasks such as content distribution, streaming, and downstream inference, all devoid of transmitting or storing raw data. The advent of INR technology, exemplified by Neural Representation for Videos (NeRV) [4], manifests considerable promise across domains like data compression and multimedia data processing. Its continual evolution furnishes a fresh technical underpinning for information hiding technology. Viewing video as a chronological depiction of the visual realm, we

can discern the RGB (Red, Green, Blue) state corresponding to each timestamp. For a video $V_t$, each moment $T$ aligns with an RGB video frame. Employing a neural network function $f$ to glean the mapping between time $t$ and RGB video frames, denoted as $V_t = f_\Theta(t)$, $f$ serves as the implicit neural representation of video $V_t$, as depicted in Fig. 1a. Implicit representation offers merits such as independence from spatial resolution, robust representational prowess, strong generalization capabilities, and ease of acquisition. Diverging from conventional deep neural networks, typically employed as tools for processing data classes, implicit representation techniques deploy neural networks to represent specific multimedia entities. Various multimedia data types can thus be transmuted into implicit neural network representations, as depicted in Fig. 1b.

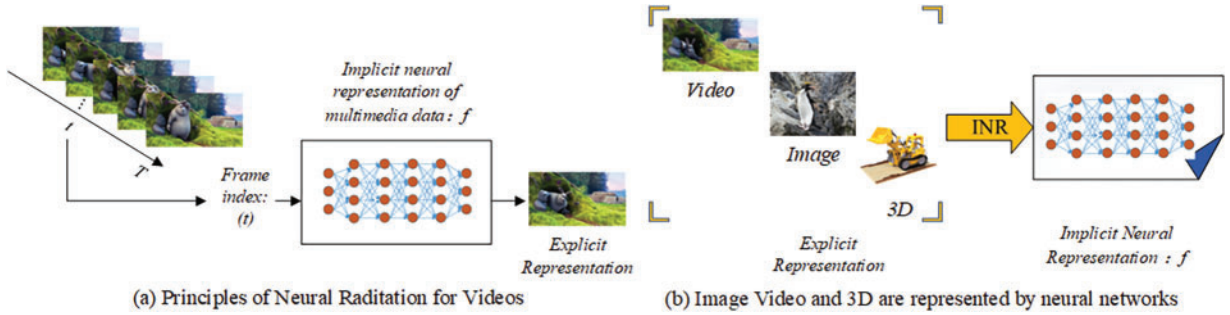


**Figure 1:** Implicit neural representation

With such representation, video can be conceptualized akin to a neural network, streamlining various video-related tasks. As NeRV technology continues to evolve in video representation, we anticipate a future where individuals frequently share their trained video content online, akin to the current sharing of images and videos. This transition replaces the traditional explicit approach with the direct sharing of network structures learned through training. Consequently, the issue of copyright protection for implicit representation-oriented NeRVs becomes increasingly pertinent. Given that NeRV segments videos into frames for training, and Invertible Neural Networks can embed and extract watermark information within these frames using a single network, we propose a scheme to address the copyright protection of NeRVs utilizing Invertible Neural Network watermarking. However, acknowledging potential distortion during rendering and malicious attacks during transmission, we incorporate a video frame quality enhancement module to mitigate these effects. The scenarios for use depiction of this scheme are illustrated in Fig. 2. Alice segments the video into frames, selects a frame for watermark embedding using invertible neural network techniques, trains the NeRV, and subsequently shares the NeRV model online for public access. Bob acquires the NeRV model and redistributes it under his name. Upon discovering this, Alice downloads the NeRV model, retrieves the embedded watermark information through the inverse process of invertible neural networks, validates her copyright, and prompts Bob to retract the unauthorized publication.

The contribution of this paper is outlined as follows:

1. We propose a watermarking scheme designed specifically to safeguard the copyright of NeRV.
2. Leveraging the reversibility inherent in invertible neural networks, we model the recovery of secret information as an inverse process of concealment. This enables both the sender and the receiver to conceal and recover secret information utilizing the same networks, thereby streamlining the process. Notably, only one network needs to be trained to achieve both functionalities simultaneously.

3. Recognizing the potential loss of video frames during the rendering process, as well as the potential for human-induced damage, we introduce a noise layer to simulate losses attributed to human factors. Additionally, we devise a frame quality enhancement module aimed at mitigating these losses, thus ensuring accurate extraction of watermark information.

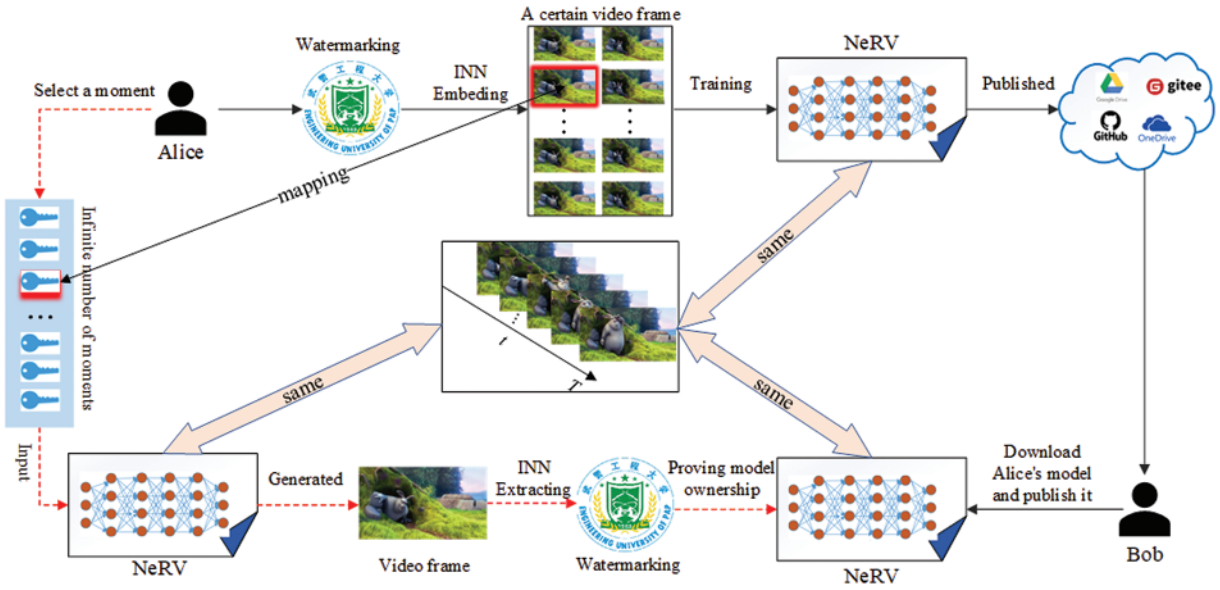


**Figure 2:** Application scenarios

## 2 Related Work

### *2.1 Video Watermarking*

Early techniques for information hiding in videos primarily focused on the original domain of the video, where the carrier for information hiding is an uncompressed video file. In this approach, the video sequence is considered as a series of consecutive frames, and a subset of frames is selected from this sequence as the carrier for information hiding. Subsequently, the pre-processed information to be embedded (e.g., following encryption or error correction code processing) is incorporated into these selected video frames using various methods. Such techniques can be categorized into those based on the null domain and those based on the transform domain.

Information-hiding algorithms operating in the original domain of videos are relatively simpler to implement. However, in practical applications, video files are often compressed to reduce their size, storage requirements, and network transmission costs. Consequently, the importance of information-hiding techniques compatible with compressed video formats has become increasingly significant and has garnered widespread attention from researchers. Notably, the H.264/AVC (Advanced Video Coding) standard (also known as MPEG-4 (Moving Pictures Experts Group) Part 10) is one of the most widely used and mature video coding standards, and information-hiding techniques based on H.264/AVC hold a predominant position in current research endeavors.

Diverging from previous studies, this paper delves into the unexplored challenge of embedding watermark information within NeRV video frames. This endeavor is crucial for safeguarding copyright and preserving ownership rights, given the increasing deployment of NeRV projects in real-world

applications, with a projected rise in their usage in the future. Thus, addressing copyright concerns associated with NeRV technology is particularly timely and pertinent.

### *2.2 Implicit Neural Representation*

Implicit neural representation (INR) introduces a novel approach to parameterizing diverse signals. The fundamental concept involves representing an object as a function approximated by a neural network, which maps coordinates to corresponding values (e.g., pixel coordinates of an image and RGB values of pixels). This methodology finds widespread application across various 2D and 3D visual tasks, including images [5,6], videos [4–8], 3D shapes [9,10], 3D scenes [3,11], and 3D structural appearance [12,13]. In contrast to explicit data representations, continuous implicit neural representations can efficiently encode high-resolution signals in a memory-efficient manner. Previously, implicit neural representations were often approximated using multilayer perceptrons, which take spatial or spatiotemporal coordinates as input and output signals (e.g., RGB values, volume density) for each point. The current NeRV representation employs a purposefully designed neural network comprising MLPs (Multilayer Perceptron) and convolutional layers. It takes the frame index as input and directly outputs all RGB values for that frame.

### *2.3 Invertible Neural Network*

Jing et al. [14] pioneered the combination of invertible neural networks with information-hiding techniques, modeling secret message recovery as the inverse operation of message hiding. They utilized the same network to concurrently achieve both information hiding and extraction. Addressing capacity limitations, Guan et al. [15] and Chen et al. [16] proposed serial and parallel methods, respectively, for concealing multiple graphs. Xu et al. [17] introduced normalized flow principles and devised DGM (Distortion-Guided Modulation) and CEM (Container Enhancement Module) modules to enhance scheme robustness. Yang et al. [18] introduced a three-step training methodology, incorporating an augmentation module and considering rounding errors to bolster scheme robustness. Luo et al. [19] introduced invertible neural networks into watermarking to safeguard artwork copyrights. Ma et al. [20] identified the susceptibility of invertible network schemes to low robustness due to their reliance on reversibility. To address this, they proposed a hybrid watermarking scheme combining invertible and non-invertible networks. This included a watermark extractor based on attention mechanisms, facilitating watermark extraction from multiple channels and selecting the optimal result to enhance scheme robustness. Lu et al. [21] broke new ground by applying invertible neural networks to video steganography, realizing a selectable video steganography scheme with substantial capacity.

## 3 Method

With the advent of neural implicit representation, the future of information communication, encompassing text, images, videos, and more, may experience a transition towards neural implicit representation. We introduce a novel approach employing invertible neural network watermarking for copyright protection of neural representation for videos. Our methodology utilizes NeRV as the cover, leveraging the strengths of invertible neural network watermarking. Specifically, watermark information is embedded within the video frames used for neural network fitting, replacing original frames to generate NeRV, subsequently disseminated on public platforms. Subsequently, extractors retrieve NeRV, input timestamp to reconstruct video frames, and employ the inverse process of

invertible neural networks to extract watermarking information, thereby acquiring the embedded watermark information, as depicted in Fig. 3.

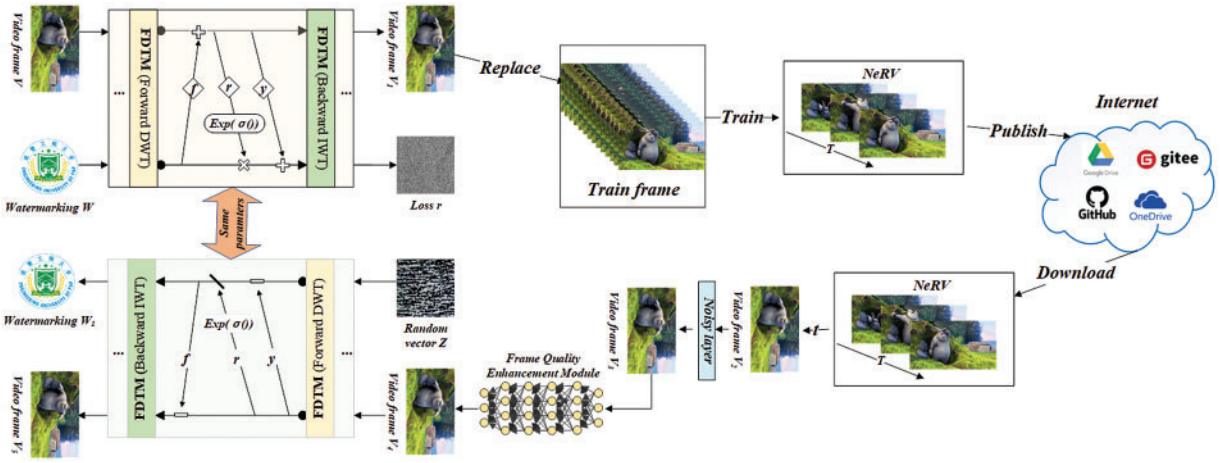


**Figure 3:** Flowchart of MarkINeRV program structure

The video frame corresponding to a specific moment $T$ is selected, and the watermark information is initially embedded using an invertible neural network to generate the watermarked video frame. These frames are utilized to train the fitted neural network, producing NeRV subsequently disseminated online. Upon retrieval, the receiver executes NeRV to generate the video frame rendered at time $t$. Due to NeRV's training process causing some watermark loss and potential interference from noise, this study simulates various noise attacks. Additionally, a video frame quality enhancement module employing a U-net network is devised to ameliorate corrupted video frames. Loss constraints are incorporated to ensure rendered frames closely resemble watermarked counterparts, yielding enhanced frames. Finally, these enhanced frames undergo an invertible neural network for watermark extraction. In contrast to alternative methods that directly embed watermarks into video frames or other locations, NeRV rendering or malicious attacks during the delivery processes can lead to significant damage to the watermark information, rendering successful recovery unattainable. Additionally, non-reversible methods often fall short in achieving blind watermarking capabilities and superior watermark extraction performance when compared to reversible neural network watermarking techniques, particularly in terms of watermark invisibility and extraction accuracy.

### *3.1 Network Architecture*

#### *3.1.1 Invertible Block*

The concealment and retrieval processes employ identical sub-blocks with shared network parameters, albeit with information flow in reverse. The network architecture in this study comprises 8 invertible blocks, each structured as follows:

$$V^{l+1} = V^l + f\left(W^l\right) \tag{1}$$

$$W^{l+1} = W^l \otimes \exp\left(\sigma\left(r\left(V^{l+1}\right)\right)\right) + y\left(V^{l+1}\right) \tag{2}$$

For the $l^{th}$ concealment block in the forward process, the inputs consist of $V^l$ and $W^l$, yielding outputs $V^{l+1}$ and $W^{l+1}$, as expressed in Eqs. (1)–(2), where $\sigma$ denotes the activation function, with

LeakyReLU (Leaky Rectified Linear Unit) employed herein. Additionally, $f(\cdot)$, $r(\cdot)$, and $y(\cdot)$ represent densely connected networks. The outputs of the final invertible block, $W^k$ and $V^k$, transform the IWT (Inverse Wavelet Transform), yielding the stego image, $V_1$, and loss information, $r$.

In the reverse recovery process, the $l^{th}$ display block with inputs ${V_4}^{l+1}$ and $Z^{l+1}$ and outputs ${V_4}^{l}$ and $Z^l$ is depicted in Eqs. (3)–(4).

$$Z^l = \left(Z^{l+1} - y\left(V_4^{l+1}\right)\right) \otimes \exp\left(-\sigma\left(r\left(V_4^{l+1}\right)\right)\right) \tag{3}$$

$$V_4^l = V_4^{l+1} - f\left(Z^l\right) \tag{4}$$

Information flow proceeds in the opposite direction, traversing through the $(l+1)^{th}$ layer and then the $l^{th}$ layer. Subsequently, following the initial layer of reversible transformation, the result undergoes IWT to yield the recovered image, $V_5$, and the recovered watermark, $W_1$.

### *3.1.2 Frequency Domain Transform Module*

Watermark information embedded within the pixel domain is susceptible to texture replication artifacts and color distortion [22]. The frequency domain and high-frequency domain offer superior suitability for watermark embedding compared to the pixel domain [23]. In this study, we employ the FDTM (Frequency Domain Transform Module) to partition the video frame into low and high-frequency wavelet subbands before the invertible transformation. The high-frequency subbands encapsulate the video frame's details, while the low-frequency subbands encompass its overall features, thereby facilitating the network's seamless fusion of watermark information into the cover video frame. Relative to direct operations within the original video frame, wavelet transformation exhibits enhanced visual fidelity, embedding the watermark information into only a few subbands, thus minimally affecting the video frame as a whole, making detection challenging. Furthermore, the commendable reconstruction properties of wavelets mitigate information loss and augment watermark embedding capabilities. Preceding entry into the invertible block, the video frame undergoes processing by the FDTM, and after Discrete Wavelet Transform, the feature map of size (B, C, H, W) is transformed into (B, 4C, H/2, W/2), where B represents batch size, H denotes height, W signifies width, and C indicates the number of channels. Discrete Wavelet Transformation (DWT) reduces computational costs, thereby expediting the training process [24]. Following the final invertible block, the feature map (B, 4C, H/2, W/2) undergoes processing by the FDTM for Inverse Wavelet Transform resulting in the generation of the watermark frame $V_1$ by restoring the feature map size to (B, C, H, W).

### *3.1.3 Neural Radiance Video*

The architecture of the NeRV network is depicted in Fig. 4. NeRV comprises two main components: the MLP block and the NeRV block. The MLP block comprises two linear layers, each followed by a GELU (Gaussian Error Linear Units) activation layer. On the other hand, the NeRV block is composed of a 3 × 3 convolutional layer with a step size of 1, a PixelShuffle operation, and a GELU activation function. The NeRV architecture consists of five NeRV blocks, with magnifications set to 5, 3, 2, 2, 2 for 1080p videos, and 5, 2, 2, 2, 2 for 720p videos. A video $V = \{v_t\}_{t=1}^{T} \in R^{T\times H\times W\times 3}$ is represented as a function where the input is the timestamp Timestamps(t), and the output is the corresponding video frame $v_t \in R^{H\times W\times 3}$ at that time. The neural network $\theta$ is employed to fit a coding function $v_t = f_\theta(t)$, which correlates to a neural network $f_\theta$ that executes the coding of the video by feeding the corresponding Timestamps(t) to the respective RGB image. However, while deep neural networks excel in numerous complex tasks across various domains, serving as versatile

function approximators [12], training timestamps $t$ directly as inputs to the neural network may yield suboptimal results [25]. Mapping low-dimensional timestamps to a high-dimensional space enhances the neural network's capability to fit video data with high-frequency information. Therefore, NeRV employs position coding [12] as the embedding function:

$$\Gamma(t) = \left(\sin\left(b^0 \pi t\right), \cos\left(b^0 \pi t\right), \ldots, \sin\left(b^{l-1} \pi t\right), \cos\left(b^{l-1} \pi t\right)\right) \tag{5}$$

where $b$ and $l$ are hyperparameters of the neural network, and the input Timestamps(t) are normalized between (0, 1]. The resulting output is then fed into the neural network.

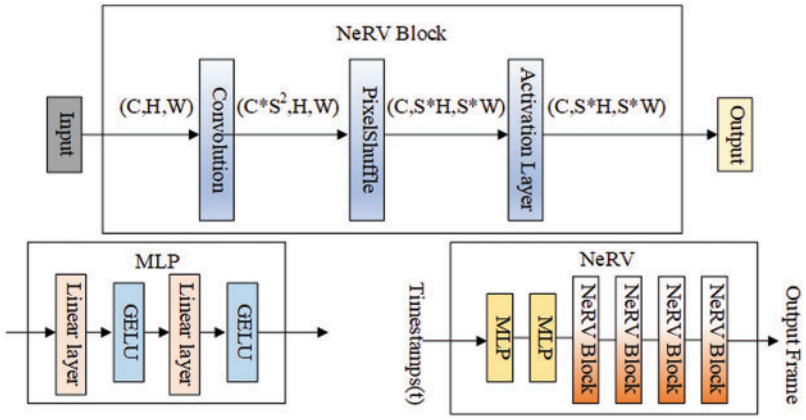


**Figure 4:** NeRV module structure

### *3.1.4 Frame Quality Enhancement Module*

Before initiating the reverse process for watermark extraction, this paper introduces a Frame Quality Enhancement Module (FQEM) to counteract the distortion effects caused by the NeRV rendering process. FQEM employs a residual convolutional coding and decoding network, as depicted in Fig. 5. On the left side, six convolutional encoders are utilized to extract features from various layers of the distorted video frame $V_3$. Subsequently, these features are inputted into the inverse convolutional decoder on the right side, along with residuals passed from the preceding layer. The resulting output is then superimposed on top of each other, completing the restoration of the video frame. By incorporating FQEM into the watermark extraction process, the rendered video frame $V_3$ undergoes preprocessing before entering the invertible neural network. This preprocessing aims to ensure sufficient similarity between it and the video frame $V_1$, facilitating the invertible neural network in more comprehensively extracting the watermark information.

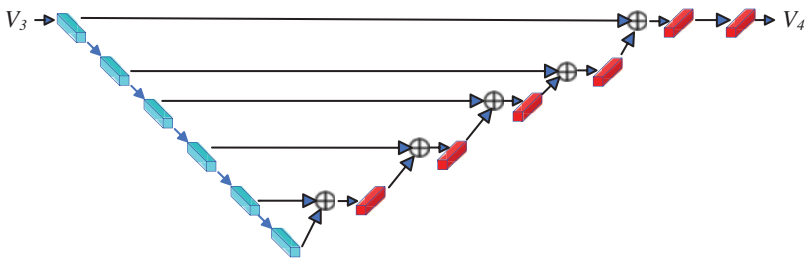


**Figure 5:** Frame quality enhancement module architecture

### 3.2 Loss

The network model training loss proposed in this paper comprises five main components.

#### 3.2.1 Embedding Loss $L_{Emb}$

The purpose of the embedding loss is to ensure that the generated watermarked frame $V_1$ is indistinguishable from the original training frame $V$.

$$L_{Emb}(\theta) = \sum_{n=1}^{N} \ell_{Emb}\left(V_1^{(n)}, V^{(n)}\right) \tag{6}$$

In Eq. (6): $N$ represents the number of training samples, and $\ell_{Emb}$ calculates the difference between the watermarked frame $V_1$ and the original training frame $V$, using the $l_2$ paradigm.

#### 3.2.2 Low-Frequency Wavelet Loss $L_{low\text{-}f}$

Reference [26] verified that watermark information embedded in high-frequency components is less detectable than that in low-frequency components. The loss aims to ensure higher visual fidelity and minimize the impact on the video frame as a whole due to embedding.

$$L_{low-f}(\theta) = \sum_{n=1}^{N} \ell_{\mathrm{f}}\left(H\left(V^{(n)}\right)_{ll}, H\left(V_1^{(n)}\right)_{ll}\right) \tag{7}$$

In Eq. (7): $N$ represents the number of training samples, $\ell_{\mathrm{f}}$ calculates the low-frequency difference between the watermarked frame $V_1$ and the original training frame $V$ and $H(\cdot)_{ll}$ represents the low-frequency sub-band operation of extracting video frames.

#### 3.2.3 Extraction Loss $L_{Ext}$

This loss ensures the consistency of the recovered watermark information $W_1$ with the embedded watermark information $W$. Minimizing the difference between the recovered watermark $W_1$ and the embedded watermark information $W$ improves the model's watermark extraction accuracy.

$$L_{Ext}(\theta) = \sum_{n=1}^{N} E_{z\sim p(z)}\left[\ell_{Ext}\left(W^{(n)}, W_1^{(n)}\right)\right] \tag{8}$$

In Eq. (8): $N$ represents the number of training samples, and $\ell_{Ext}$ computes the difference between the watermark information $W$ and the recovered watermark $W_1$.

The total loss function of the invertible neural network comprises a weighted combination of three components: the Embedding loss, the Low-frequency wavelet loss, and the Extraction loss.

$$L_{total}(\theta) = \lambda_1 L_{Emb} + \lambda_2 L_{low-f} + \lambda_3 L_{Ext} \tag{9}$$

During the training process, initially, $\lambda_2$ is set to 0, meaning that the network model undergoes direct pre-training without considering the effect of $L_{low\text{-}f}$ on the network. This allows the network model to first acquire the fundamental embedding-extraction capabilities. Subsequently, the $L_{low\text{-}f}$ constraint term is gradually introduced to further refine the network model, enabling it to embed watermark information in the high-frequency domain of the training video frames. This step aims to minimize the overall impact on the video frames resulting from the embedding of watermark information.

#### 3.2.4 Neural Radiation Video Loss L

To ensure the effectiveness of the recovered video frames, the $l_1$ and Structural Similarity (SSIM) losses are used as the loss function of the neural radiation video.

$$L = \frac{1}{T}\sum_{t=1}^{T} \alpha \parallel f_\theta(t) - v_t \parallel_1 + (1-\alpha)\left(1 - SSIM\left(f_\theta(t), v_t\right)\right) \tag{10}$$

In Eq. (10): $T$ is the number of video frames, $f_\theta(t)$ is the predicted frame, $v_t$ is the true frame and $\alpha$ is a hyperparameter to balance the weight of the $l_1$ and SSIM losses.

#### 3.2.5 MSE Loss for Frame Quality Enhancement Module

In this paper, the training of frame quality enhancement modules and invertible neural networks are treated as independent processes. The loss of the frame quality enhancement module is constrained by the Mean Squared Error (MSE), to guarantee that the video frame $V_3$, enhanced by the frame quality enhancement module, can be accurately reconstructed into the watermarked frame $V_1$ generated by the invertible neural network. This ensures that the generated video frame $V_4$ and the watermarked frame $V_1$ maintain a sufficiently close resemblance, thereby resisting potential damage and loss of watermarking information caused by the rendering process and any intentional alterations.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(V_{4i}, V_{1i}\right) \tag{11}$$

In Eq. (11): $V_{4i}$ is the $i^{th}$ video frame generated by the enhancement module, and $V_{1i}$ is the $i^{th}$ watermarked frame.

## 4 Experiments

### 4.1 Setting

In this study, we employed the PyTorch platform with Cuda version 11.6 and an Nvidia GeForce RTX2070 GPU (Graphics Processing Unit) for network modeling. The training utilized NeRV [4]'s scikit-video dataset "Big Buck Bunny" sequence and UVG (Ultra Video Group) dataset [27], with citation of NeRV source code. The invertible neural network, which embeds watermarks solely on the trained dataset, was structured based on modifications of Half Instance Normalization Network (HINet) [14]. Considering the diversity, high resolution, and authenticity of the DIV2K (Diverse 2K) [28] dataset, we employed it for training the invertible neural network model. Specifically, the DIV2K training dataset, comprising 800 images at a resolution of 1024 × 1024, was utilized for training, while the DIV2K validation dataset, consisting of 100 images at a resolution of 1024 × 1024, was used for model validation. Furthermore, the DIV2K test dataset, containing 100 images at a resolution of 1024 × 1024, was employed to assess the effectiveness of the network model. We employed the Adam optimizer with hyperparameters set as follows: $\lambda_1 = 5$, $\lambda_2 = 0.5$, $\lambda_3 = 1$, learning rate $= 1 \times 10^{-4.5}$, and batch size $= 2$. The entire network model comprises 8 invertible blocks, with each block incorporating three DenseNet blocks, encompassing 7 layers of convolutional blocks designated as $f(\cdot)$, $r(\cdot)$, and $y(\cdot)$, respectively.

### 4.2 Robustness

Four traditional noise attacks—Poisson, Gaussian, Pepper, and Speckle—are applied individually to corrupt Video frame $V_2$, aiming to assess the scheme's robustness, as depicted in Fig. 6. Although

Video frame $V_2$ is subjected to these four noises individually, Video frame $V_3$, which has been processed by the frame quality enhancement module, exhibits no visually detectable interference.

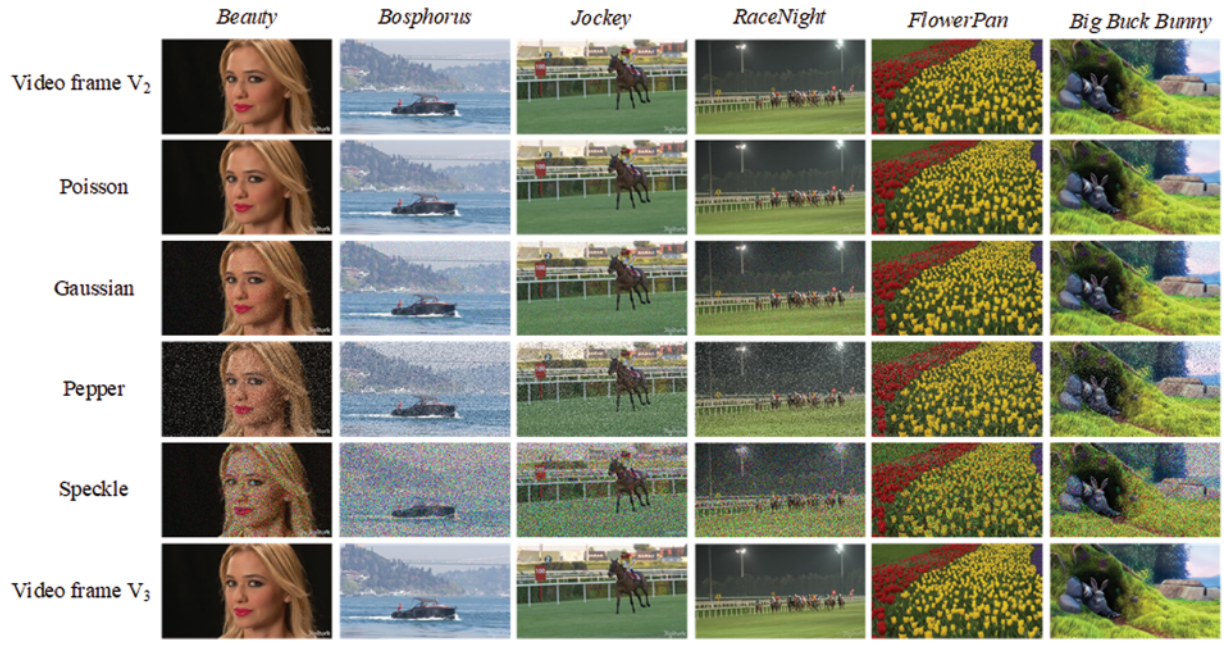


**Figure 6:** Visualisation of the dataset after noise treatment and after the enhancement module

Table 1 presents the experimental results, comparing the PSNR (Peak Signal-to-Noise Ratio)/SSIM (Structural Similarity) values of video frames, employing various watermarking methods, after being subjected to the noise layer with the original video frames. The results indicate a notable resilience of the method proposed in this paper against the aforementioned four types of noise interference.

**Table 1:** Comparison of anti-interference capability

| Noises | MarkINeRV | | Hidden [29] | | Baluja [30] | | Rehman et al. [31] | |
|---|---|---|---|---|---|---|---|---|
| | PSNR | SSIM | PSNR | SSIM | PSNR | SSIM | PSNR | SSIM |
| Poisson | 52.02165 | 0.99662 | 35.56660 | 0.93387 | 27.35738 | 0.80127 | 25.44673 | 0.75654 |
| Gaussian | 51.99809 | 0.99661 | 33.18629 | 0.90325 | 26.84235 | 0.78555 | 24.62278 | 0.75032 |
| Pepper | 51.98431 | 0.99660 | 31.23473 | 0.87287 | 26.31545 | 0.76067 | 23.94537 | 0.74642 |
| Speckle | 50.65157 | 0.99589 | 30.43941 | 0.85938 | 25.37853 | 0.75373 | 23.27536 | 0.73974 |

In addition, the video frames generated by NeRV during the rendering process may lead to the loss of original watermark information embedded within them. Consequently, we propose the incorporation of a video frame quality enhancement module, distinct from existing approaches. This module is designed not only to withstand various types of noise but also to mitigate the loss of watermark information incurred during rendering. As depicted in Fig. 7, it is evident that methods lacking this enhancement module fail to extract the watermark.

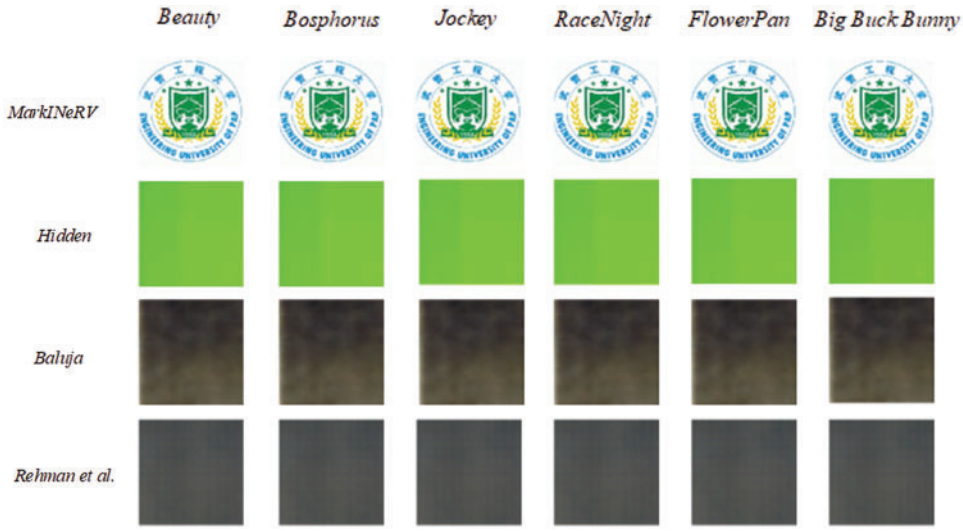


**Figure 7:** Comparison of visualization results of different methods for recovered watermark

### *4.3 Accuracy*

MarkINeRV integrates watermarking information $W$ into the scikit-video dataset "Big Buck Bunny" sequence and the UVG dataset via a forward invertible neural network. Subsequently, NeRV is employed for training. Timestamps are then inputted into NeRV to generate video frames corresponding to these timestamps. These generated video frames undergo processing by a frame quality enhancement module and finally, the recovered watermarking $W_1$ is retrieved via the inverse invertible neural network. Evaluation of the recovered watermarking $W_1$ is conducted for accuracy against the original watermarking using four metrics. As depicted in Fig. 8, the values of each metric under each dataset indicate favorable performance: PSNR surpasses 29 dB, SSIM surpasses 0.91, MAE (Mean Absolute Error) falls below 8, and RMSE (Root Mean Square Error) falls below 11.

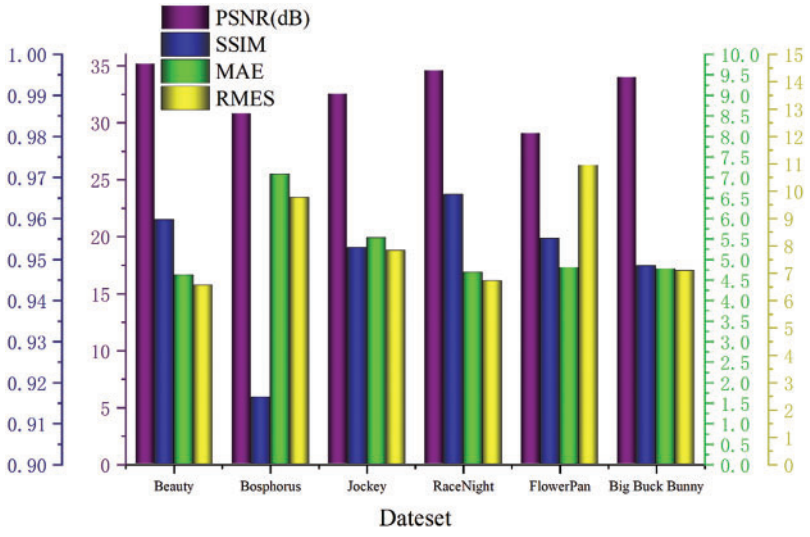


**Figure 8:** The accuracy of the recovered watermark $W_1$ with the watermark message $W$

The watermarking information, Watermarking $W_1$, extracted by this scheme exhibits visual consistency with the original embedded watermarking information, Watermarking $W$, as demonstrated in Fig. 9. Furthermore, even upon local enlargement, the watermarking information remains indiscernible, underscoring the exceptional performance of this paper's scheme in safeguarding NeRV's copyright and validating its feasibility.

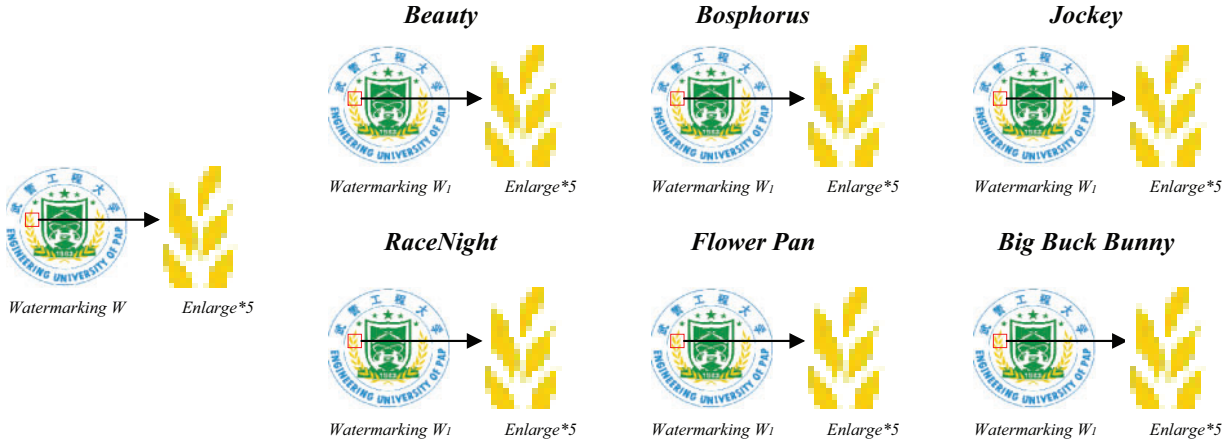


**Figure 9:** Visualisation results of recovered watermarks for different datasets

### *4.4 Imperceptibility*

The invertible network watermarking scheme (MarkINeRV) achieves blind watermarking, aiming to minimize the distortion rate between Video frame $V$ and Video frame $V_1$. In evaluating the imperceptibility of our method and three alternative schemes, we employ the PSNR metric. The Video frame $V$ is compared with Video frame $V_1$, as depicted in Table 2, and the experimental data show that the MarkINeRV scheme outperforms the other schemes in terms of imperceptibility.

**Table 2:** Comparison of imperceptibility indicators

| Methods | Video frame $V$/Video frame $V_1$ (PSNR) | | | | | |
|---|---|---|---|---|---|---|
| | Beauty | Bosphorus | Jockey | RaceNight | FlowerPan | Big buck bunny |
| MarkINeRV | 48.99 | 46.52 | 44.60 | 52.86 | 46.98 | 46.78 |
| Hidden | 35.21 | 36.71 | 34.79 | 36.43 | 37.68 | 35.70 |
| Baluja | 36.77 | 36.38 | 36.59 | 35.88 | 35.01 | 34.13 |
| Rehman et al. | 32.91 | 30.97 | 29.68 | 32.92 | 29.17 | 33.35 |

Analysis depicted in Fig. 10 reveals that embedding the watermark within video frames from both the scikit-video dataset's 'Big Buck Bunny' sequence and the UVG dataset yields inconspicuous alterations. A comparison of Video frame $V_1$ generated by four methods against Video frame $V$ demonstrates the difficulty in discerning the presence of watermark information. Experimental findings confirm the imperceptibility of the watermark embedded via our method, achieving the objective of blind watermarking regardless of its presence within the video frame.


**Figure 10:** Visualisation results comparing Video frame $V_1$ with Video frame $V$

### *4.5 Ablation Study*

Traditional deep learning robust watermarking techniques such as HiNet [14] and Initial Sequence Number (ISN) [21] are not directly applicable to our task. These methods rely on reversibility and do not adequately consider the susceptibility of video frames to corrupting watermarks embedded within video frames during the NeRV training process. Hence, MarkINeRV introduces a video frame quality enhancement module before the watermark extraction operation to mitigate the impact of the NeRV training process. By incorporating the FQEM structure, the PSNR between WaterMarking $W$ and WaterMarking $W_1$ is enhanced from 6.94 dB to 29.06 dB, as demonstrated in Table 3. The experimental findings underscore the significant contribution of FQEM in ensuring the successful extraction of watermark information.

**Table 3 :** Effectiveness of network architecture and design strategies

| FQEM | Low-frequency wavelet loss | FDTM | Comparison of WaterMarking $W$ and WaterMarking $W_1$ (PSNR) |
|---|---|---|---|
| × | √ | √ | 6.94 dB |
| √ | × | √ | 15.83 dB |
| √ | √ | × | 17.28 dB |
| √ | √ | √ | 29.06 dB |

## 5 Conclusion

In this paper, we propose for the first time a scheme to protect Neural Representations for Videos using invertible neural network watermarking (MarkINeRV), which achieves copyright protection

for NeRV. MarkINeRV uses invertible neural networks to embed and extract watermarks on video frames, modeling the embedding and extraction of the watermarks as a forward and inverse process of the invertible network, while adding a video frame quality enhancement module to the intermediate process to compensate for the loss of watermark information caused by the NeRV rendering process and to achieve copyright protection for neural radiation videos in implicit neural representations. Experiments are conducted to evaluate the feasibility of the coming scheme in terms of accuracy, robustness, and invisibility, respectively.

(1) Accuracy: Due to the excellent performance of the invertible neural network in the information hiding direction, the PSNR of watermark information recovery reaches 35 dB.
(2) By designing the video frame quality enhancement module to offset the effects caused by the rendering process as well as vandalism, the scheme has good robustness.
(3) An invertible neural network embeds the watermark information into the high-frequency region by frequency domain transform module to achieve good invisibility.

The results show that the copyright owner can achieve the embedding and extraction of the watermark by using MarkINeRV to verify the copyright of the neural radiation videos. The future will be devoted to the reduction or even elimination of loss information in invertible networks to ensure higher extraction accuracy.

**Acknowledgement:** In the process of designing this thesis, I am grateful to my school for providing me with the opportunity to learn. Throughout my studies, Liu Jia has provided meticulous guidance, from selecting the topic and structuring the thesis framework to detailed revisions, offering invaluable advice and suggestions. With a rigorous and pragmatic approach, Liu Jia's dedication, conscientiousness, diligence, and innovative spirit have profoundly influenced me. His profound knowledge, broad perspective, and sharp insights have deeply inspired me. This thesis was completed under his careful guidance and unwavering support. I would like to sincerely thank all the teachers who took time out of their busy schedules to review this paper.

**Funding Statement:** This study is supported by the National Natural Science Foundation of China, with Fund Numbers 62272478, 62102451, the National Defense Science and Technology Independent Research Project (Intelligent Information Hiding Technology and Its Applications in a Certain Field) and Science and Technology Innovation Team Innovative Research Project "Research on Key Technologies for Intelligent Information Hiding" with Fund Number ZZKY20222102.

**Author Contributions:** The authors confirm the following contributions to this article: Research concept and design: Wenquan Sun and Weina Dong; Experimental, analytical, and interpretive results: Lifeng Chen and Wenquan Sun; Drafted by Wenquan Sun. Jia Liu and Fuqiang Di reviewed the results and approved the final version of the manuscript. All authors reviewed the results and approved the final version of the manuscript.

**Availability of Data and Materials:** The archived version of the code described in this manuscript can be freely accessed through GitHub (https://github.com/swq797/MarkINeRV-A-Robust-Watermarking-Scheme.git, accessed on 4 April 2024).

**Ethics Approval:** Not applicable.

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

## References


[1] Z. Chen and H. Zhang, "Learning implicit fields for generative shape modeling," in *2019 IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, 2019, pp. 5932–5941.

[2] J. Ye, Y. Chen, N. Wang, and X. Wang, "GIFS: Neural implicit function for general shape representation," in *2022 IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, New Orleans, LA, USA, 2022, pp. 12819–12829.

[3] C. Jiang, A. Sud, A. Makadia, J. Huang, M. Niebner and T. Funkhouser, "Local implicit grid representations for 3d scenes," in *2020 IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Seattle, WA, USA, 2020, pp. 6000–6009.

[4] H. Chen, B. He, H. Wang, Y. Ren, S. N. Lim and A. Shrivastava, "NeRV: Neural representations for videos," *Neural Inf. Process. Syst.*, vol. 34, pp. 21557–21568, 2021. doi: 10.48550/arXiv.2110.13903.

[5] S. Yang, M. Ding, Y. Wu, Z. Li, and J. Zhang, "Implicit neural representation for cooperative low-light image enhancement," in *2023 IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Paris, France, 2023, pp. 12872–12881.

[6] M. D. Li, J. Deng, S. P. Xiao, and S. W. Chen, "Semi-supervised implicit neural representation for polarimetric ISAR image super-resolution," *IEEE Geosci. Remote Sens. Lett.*, vol. 20, pp. 3505505, 2023. doi: 10.1109/LGRS.2023.3287283.

[7] H. Chen, M. Gwilliam, S. N. Lim, and A. Shrivastava, "HNeRV: A hybrid neural representation for videos," in *2023 IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Vancouver, BC, Canada, 2023, pp. 10270–10279.

[8] Z. Chen *et al.*, "VideoINR: Learning video implicit neural representation for continuous space-time super-resolution," in *2022 IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, New Orleans, LA, USA, 2022, pp. 2037–2047.

[9] K. Genova, F. Cole, D. Vlasic, A. Sarna, W. T. Freeman and T. A. Funkhouser, "Learning shape templates with structured implicit functions," in *2019 IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Seoul, Republic of Korea, 2019, pp. 7153–7163.

[10] K. Genova, F. Cole, A. Sud, A. Sarna, and T. Funkhouser, "Deep structured implicit functions," arXiv:1912.06126v1, 2019. doi: 10.48550/arXiv.1912.06126.

[11] R. Chabra *et al.*, "Deep local shapes: Learning local SDF priors for detailed 3D reconstruction," in *Comput. Vis.–ECCV 2020*, Glasgow, UK, 2020, pp. 608-605.

[12] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi and R. Ng, "NeRF: Representing scenes as neural radiance fields for view synthesis," in *Comput. Vis.–ECCV 2020*, Glasgow, UK, 2020, pp. 405–421.

[13] M. Niemeyer, L. M. Mescheder, M. Oechsle, and A. Geiger, "Differentiable volumetric rendering: Learning implicit 3D representations without 3d supervision," in *2020 IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Sattle, WA, USA, 2020, pp. 3501–3512.

[14] J. Jing, X. Deng, M. Xu, J. Wang, and Z. Guan, "HiNet: Deep image hiding by invertible network," in *2021 IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Montreal, QC, Canada, 2021, pp. 4713–4722.

[15] Z. Guan *et al.*, "DeepMIH: Deep invertible network for multiple image hiding," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 1, pp. 372–390, 2023. doi: 10.1109/TPAMI.2022.3141725.

[16] Z. Chen, T. Liu, J. J. Huang, W. Zhao, X. Bi and M. Wang, "Invertible mosaic image hiding network for very large capacity image steganography," in *2024 IEEE Int. Conf. Acoust. Speech Sig. Process. (ICASSP)*, Seoul, Republic of Korea, 2024, pp. 4520–4524.

[17] Y. Xu, C. Mou, Y. Hu, J. Xie, and J. Zhang, "Robust invertible image steganography," in *2022 IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, New Orleans, LA, USA, 2022, pp. 7865–7874.

[18] H. Yang, Y. Xu, X. Liu, and X. Ma, "PRIS: Practical robust invertible network for image steganography," *Eng. Appl. Artif. Intell.*, vol. 133, 2024. doi: 10.1016/j.engappai.2024.108419.

[19] Y. Luo, T. Zhou, F. Liu, and Z. Cai, "IRWArt: Levering watermarking performance for protecting high-quality artwork images," in *Proc. ACM Web Conf. 2023*, New York, NY, USA, 2023, pp. 2340–2348.

[20] R. Ma *et al.*, "Towards blind watermarking: Combining invertible and non-invertible mechanisms," in *Proc. 30th ACM Int. Conf. Multimed.*, New York, NY, USA, 2022, pp. 1532–1542.
[21] S. P. Lu, R. Wang, T. Zhong, and P. L. Rosin, "Large-capacity image steganography based on invertible neural networks," in *2021 IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Nashville, TN, USA, 2021, pp. 10811–10820.
[22] X. Weng, Y. Li, L. Chi, and Y. Mu, "High-capacity convolutional video steganography with temporal residual modeling," in *Proc. 2019 Int. Conf. Multimed. Retri.*, New York, NY, USA, 2019, pp. 87–95.
[23] B. Ma, Z. Tao, R. Ma, C. Wang, J. Li and X. Li, "A high-performance robust reversible data hiding algorithm based on polar harmonic fourier moments," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 34, no. 4, pp. 2763–2774, 2024. doi: 10.1109/TCSVT.2023.3311483.
[24] N. R. Zhou, L. L. Hu, Z. W. Huang, M. M. Wang, and G. S. Luo, "Novel multiple color images encryption and decryption scheme based on a bit-level extension algorithm," *Expert. Syst. Appl.*, vol. 238, no. 42024. doi: 10.1016/j.eswa.2023.122052.
[25] N. Rahaman *et al.*, "On the spectral bias of neural networks," in *Int. Conf. Mach. Learn.*, 2018, pp. 5301–5310.
[26] S. Baluja, "Hiding images in plain sight: Deep steganography," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, Red Hook, NY, USA, 2017, pp. 2066–2076.
[27] A. Mercat, M. Viitanen, and J. Vanne, "UVG dataset: 50/120fps 4K sequences for video codec analysis and development," in *Assoc. Comput. Mach.*, New York, NY, USA, Association for Computing Machinery, 2020, pp. 297–302.
[28] E. Agustsson, "Challenge on single image super-resolution: Dataset and study," in *2017 IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Honolulu, HI, USA, 2017, pp. 1122–1131.
[29] J. Zhu, P. Kaplan, J. Johnson, and F. F. Li, "Hidden: Hiding data with deep networks," in *European Conf. Comput. Vis.*, Berlin, Heidelberg, 2018, pp. 682–697.
[30] S. Baluja, "Hiding images within images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 7, pp. 1685–1697, 2020. doi: 10.1109/TPAMI.2019.2901877.
[31] A. Ur Rehman, R. Rahim, S. Nadeem, and S. U. Hussain, "End-to-end trained cnn encoder-decoder networks for image steganography," in *European Conf. Comput. Vis.*, Berlin, Heidelberg, 2018, pp. 723–729.