Adversarial Attacks on Invisible Watermarks

Jan Gorczyński, Michał Opioła, Marta Figurska

January 2026

1 Introduction

Watermarking techniques are widely used to protect intellectual property, verify authenticity, and trace the origin of digital content. Recently, they have gained importance in identifying AI-generated images, which are increasingly difficult to distinguish from real images by visual inspection alone. In this study, we investigated several watermarking techniques and corresponding adversarial attack methods, and evaluated their effectiveness against one another.

2 Invisible Watermarking Methods

2.1 Invisible watermarker

The `add_watermark_png` function from the *image-invisible-watermarker* library implements a spatial-domain invisible watermarking technique designed specifically for PNG images. This method embeds a watermark by modifying the least significant bits (LSB) of pixel values, which allows information to be hidden without causing visible changes to the image.

During the embedding process, the watermark is converted into a bit sequence. These bits are then inserted into the least significant bits of selected color channels (typically RGB) of the host image. Because changes in the least significant bits only slightly alter pixel intensities, the watermark remains imperceptible to the human eye.

This watermarking method is simple, fast, and easy to implement, making it suitable for basic copyright protection. However, since it operates directly in the spatial domain, it is less robust to image processing operations such as resizing, heavy compression, or noise addition when compared to frequency-domain or deep-learning-based watermarking methods.

2.2 DWT

The Discrete Wavelet Transform (DWT) is a frequency-domain watermarking technique that takes advantage of how images represent information at multiple resolutions. DWT decomposes an image into several sub-bands corresponding to different frequency components, typically separating low-frequency (approximation) and high-frequency (detail) information in both horizontal and vertical directions.

To embed a watermark, the image is first transformed using DWT, and the watermark is inserted by modifying coefficients in selected sub-bands, usually the mid-frequency bands. These bands are chosen because changes in low-frequency components can noticeably affect image quality, while high-frequency components are more sensitive to noise and compression. After embedding, the inverse DWT is applied to obtain the watermarked image.

DWT-based watermarking provides good robustness against common attacks such as compression.

2.3 DCT

The Discrete Cosine Transform (DCT) is another widely used watermarking method that operates in the frequency domain. In DCT-based watermarking, the image is divided into small blocks, and each block

is transformed from the spatial domain into frequency coefficients using DCT. This process separates image content into low, mid, and high frequency components.

The watermark is embedded by modifying selected mid-frequency DCT coefficients within each block. This ensures that the watermark is not visually noticeable and remains robust to compression, especially JPEG compression, which also relies on DCT. Once the watermark is embedded, the inverse DCT is applied to reconstruct the watermarked image.

DCT-based watermarking is computationally efficient and well suited for images that are frequently compressed or transmitted. However, its block-based nature can make it sensitive to geometric attacks such as rotation or cropping.

2.4 SSL-based watermarking

SSL-based watermarking embeds watermarks in the latent feature space of a neural network trained using self-supervised learning. In the method proposed in *Watermarking Images in Self-Supervised Latent Spaces*, a ResNet-50 architecture trained with the DINO self-supervised framework is used as a feature extractor.

The role of the ResNet is to transform an input image into a high-dimensional feature representation that captures its semantic content. Because the network is trained using self-supervision, these features are naturally robust to common image transformations such as cropping, resizing, and color changes. This property is essential for watermark robustness.

To embed a watermark, the image is slightly modified so that its feature representation, produced by the frozen ResNet network, moves toward a predefined target region in the latent space. These modifications are visually imperceptible and are computed using an iterative optimization process. Data augmentations are applied during embedding to ensure that the watermark remains detectable even after image transformations.

During detection, the same ResNet-based feature extractor is used to analyze the image and verify the presence of the watermark or decode the hidden message. This approach achieves strong robustness and imperceptibility, but it requires access to the same pretrained network and has a higher computational cost compared to traditional watermarking methods.

2.5 VideoSeal

The VideoSeal model uses convolutional neural networks with residual connections to embed hidden information into images. In this project, the approach originally designed for video is applied only to single images.

The embedder is based on an U-Net architecture with 16M parameters in total, while the extractor is based on a vision transformer with 24M parameters. The embedder takes as input a batch of images or video frames and a binary message and produces watermarked images or frames. The extractor then attempts to recover the original message from them.

There is a multistage training strategy, where the model is first pre-train on images and then continue training on a mix of images and videos using a scheduled approach. This approach has few benefits, first it allows us to leverage the faster training times of image-based models while still adapting to video-specific distortions. Second, this approach provides more stable training and yields significant improvements in terms of bit accuracy, and robustness to higher compression rates.

Figure 1: Image after watermarking with VideoSeal

3 Attack Methods

3.1 Simple Computer Vision Methods

We evaluated several basic image distortion techniques commonly used in computer vision to assess their impact on invisible watermarks. The following transformations were applied:

- **Rotation**: Images were rotated by 10 degrees in the clockwise direction.
- **Noise**: Additive Gaussian noise sampled from a normal distribution $\mathcal{N}(0, 10)$ was applied independently to each pixel.
- **Blur**: A Gaussian blur with a kernel size of $(7, 7)$ was applied to the image.
- **Combined Method**: A composite attack that applies rotation, noise addition, and Gaussian blur sequentially.

3.2 Diffusion-Based Attacks

After reviewing existing research on adversarial attacks against invisible watermarks, we concluded that diffusion-based attacks are generally among the most effective. Consequently, we adopted a diffusion-based approach using the pretrained latent diffusion model `runwayml/stable-diffusion-v1-5`.

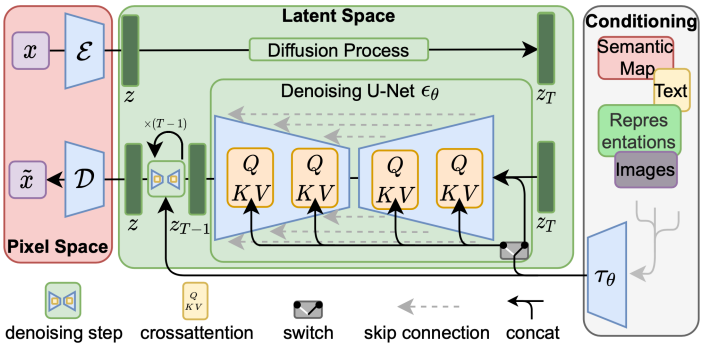

Figure 2: Latent diffusion model architecture

To perform the attack, noise was injected into the latent representation of the image, followed by reconstruction through the decoder. As a post-processing step, gamma correction was applied to mitigate the characteristic "fog-like" artifacts introduced during the diffusion process.

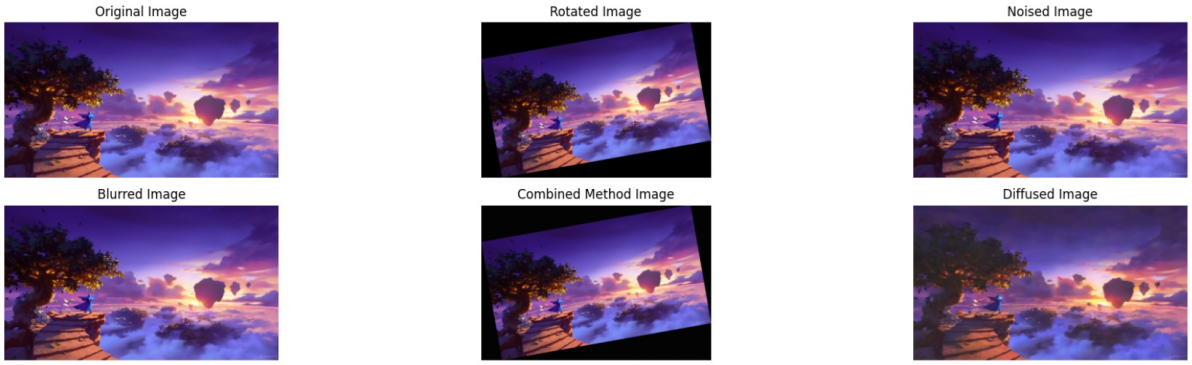

Figure 3: Different attacks visualized

4 Results and Summary

In summary, our experiments indicate that diffusion-based attacks are the most effective among the evaluated methods for disrupting invisible watermark detectors. However, this approach was still insufficient to reliably defeat a state-of-the-art deep watermarking model. Achieving stronger attack performance would likely require training adversarial methods specifically against a targeted watermarking model.

Attack Method	**LSB**	**DCT**	**DWT**	**SSL**	**Video Seal**
Rotation	1	1	1	0	0
Noise	1	0	0	0	0
Blur	1	1	1	0	0
Combined Method	1	1	1	0	0
Diffusion	1	1	1	1	0

Table 1: Effectiveness of attack methods against different watermarking techniques (1 = attack successful, 0 = attack unsuccessful)