Invisible Watermarking with Deep Neural Networks: A Saliency-Aware Embedding Approach

Anonymous submission

Paper ID

Abstract

Invisible image watermarking must hide ownership data without introducing visible artifacts, yet most methods either become noticeable when embedding is strong enough to resist attacks or break under common distortions. To solve this, we propose an end-to-end deep learning framework in which a SaliencyNet learns per-pixel attention maps that are inverted to steer a 64-bit watermark into low-attention regions; this guided map is concatenated with the cover image and processed by a convolutional encoder, decoder, and adversarial discriminator trained jointly under on-the-fly Gaussian noise simulation—maximizing imperceptibility while maintaining robustness.

1. Introduction

1.1. Problem Motivation

With the rise of digital content sharing, protecting ownership of visual media has become increasingly critical. Invisible watermarking enables embedding ownership information directly into images without disrupting their appearance, but it faces a fundamental challenge: balancing imperceptibility with robustness. Most existing techniques either leave visible artifacts or fail under common distortions like noise.

1.2. Novelty

To address this, we propose a novel saliency-aware deep learning framework that adaptively guides watermark embedding away from visually sensitive regions using a learned pixel-level saliency map. Unlike uniform or frequency-domain methods, our approach jointly trains a saliency network, encoder, decoder, and adversarial discriminator in an end-to-end fashion, ensuring the watermark remains both hidden and recoverable under real-world noise.

2. Method

Our method is designed to embed a 64-bit binary watermark into natural images in a way that maximizes robustness to distortions while preserving imperceptibility. The full pipeline consists of four core components trained end-to-end: SaliencyNet, Encoder, Decoder, and Discriminator. A dedicated distortion function is used during training to simulate real-world perturbations. Below, we describe each component in detail.

2.1. Distortion Module

The implemented distortion function serves as a comprehensive simulation framework for evaluating watermark robustness against a diverse set of geometric and non-geometric image transformations. This module systematically applies a sequence of parametrized distortions to watermarked images, emulating potential adversarial manipulations or common processing operations that may occur during digital image transmission and storage.

The distortion pipeline incorporates multiple transformation categories: Additive Gaussian Noise: Random noise sampled from a normal distribution ($\mu = 0$, $\lambda = 0.05$) is applied to simulate sensor noise, compression artifacts, and channel transmission errors. The intensity is calibrated to maintain visual integrity while presenting a significant challenge to watermark recovery.

Geometric Rotational Transformations: Angular perturbations within the range of ± 15 are introduced to evaluate invariance to orientation modifications. This transformation simulates both intentional editing and incidental rotations that occur during image capture or alignment.

Reflection Operations: Horizontal and vertical flip operations are stochastically applied with 50% probability for each axis, creating mirror transformations that challenge the spatial coherence of embedded watermarks.

Composite Affine Transformations: A parameterized affine warp combines: Translation vectors within ±10% of image dimensions Scaling factors ranging from 0.9 to 1.1 Shearing operations between ± 10

These transformations are applied sequentially, creating a compound distortion environment that significantly exceeds the complexity of single-transformation attacks. The output is constrained to the valid pixel intensity range through clamping operations.

This rigorous distortion protocol ensures that watermarks deemed robust under this evaluation framework can withstand real-world manipulations encountered in practical deployment scenarios, including incidental adjustments during post-processing, intentional editing operations, and degradations from compression or transmission artifacts.

2.2. Encoder

The proposed encoder network implements a perceptually-aware watermarking approach that strategically embeds information while preserving visual quality in semantically significant image regions. The architecture incorporates three sequential processing stages designed to achieve an optimal balance between watermark robustness and imperceptibility.

Initially, the encoder performs dimensionality expansion of the binary watermark payload through a parametrized fully-connected layer that maps from the compact watermark representation (of dimension `watermark_length`) to a high-dimensional spatial domain (256 × 256). This transformation can be formulated as:

$$W_{\text{map}} = \phi_{\text{FC}}(w) \in^{B\times 1\times 256\times 256} \tag{1}$$

where w represents the input watermark vector and ϕ_{FC} denotes the fully-connected transformation function. This operation distributes the watermark information across the spatial domain, facilitating its integration with the cover image while maintaining recoverability.

A distinctive feature of the proposed architecture is its incorporation of content-adaptive embedding through saliency-guided modulation. The network attenuates watermark strength in perceptually significant image regions by applying a complementary saliency mask:

$$W_{\text{guided}} = W_{\text{map}} \odot (1 - S) \tag{2}$$

where S represents the saliency map and $\odot$ denotes element-wise multiplication. This approach shares conceptual similarities with the cross-attention mechanism described in recent literature, wherein the embedder identifies semantically appropriate embedding locations. The inverse relationship with the saliency map ensures that visually critical regions maintain higher fidelity to the original image, addressing the fundamental imperceptibility requirement of robust watermarking systems.

The final stage of the encoder performs feature fusion and image synthesis through a computationally efficient convolutional architecture:

- Channel-wise concatenation of the original image tensor $I \in^{B\times 3\times 256\times 256}$ with the guided watermark map $W_{\text{guided}} \in^{B\times 1\times 256\times 256}$, forming a 4-channel composite representation.
- Processing through a sequence of convolutional operations:
  - An initial convolution (kernel size 3×3, padding 1) that expands the representation to 64 feature maps
  - Non-linear activation via ReLU function
  - A projection convolution (kernel size 1×1) that synthesizes the final 3-channel output
  - Sigmoid activation ensuring output values are constrained to the normalized range $[0, 1]$

The minimal layer configuration employs parameter-efficient convolutions while maintaining sufficient representational capacity for effective watermark embedding. This architectural design choice addresses computational efficiency considerations without compromising the watermarking performance, making the approach suitable for deployment scenarios with various resource constraints.

The output of this encoder is a visually similar watermarked image I' that maintains the essential characteristics of the original while incorporating a recoverable watermark payload strategically distributed according to perceptual significance:

$$I' = \text{Sigmoid}\big(\text{Conv}_{1\times 1}(\text{ReLU}(\text{Conv}_{3\times 3}([I \| W_{\text{guided}}])))\big) \tag{3}$$

2.3. Decoder

The watermark extraction module employs a computationally efficient convolutional architecture designed to recover embedded watermarks from potentially corrupted or geometrically distorted images. This decoder network demonstrates resilience against various image transformations while maintaining a parameter-efficient design suitable for practical deployment scenarios.

The decoder initiates the watermark recovery process through hierarchical feature extraction, implemented as a sequence of spatial convolutions that progressively isolate watermark-relevant information. Specifically, the architecture employs:

An initial convolutional layer (3 → 64 channels, kernel size 3×3 with padding) that performs multi-scale feature extraction from the input watermarked image, capturing both local texture patterns and global spatial relationships that may encode watermark information.

A subsequent convolutional layer (maintaining 64 channels with identical kernel configuration) that refines and enhances the feature representation through additional nonlinear processing.

Both convolutional operations are followed by ReLU activation functions, introducing non-linearity that enables the network to learn complex mapping functions between the watermarked image domain and the embedded information space. This can be formulated as:

$$F_1 = \sigma(W_1 * I_{\text{wm}} + b_1) \quad (4)$$

$$F_2 = \sigma(W_2 * F_1 + b_2) \quad (5)$$

where I_{wm} represents the input watermarked image, W_i and b_i denote the convolutional weights and biases, $*$ represents the convolution operation, and σ is the ReLU activation function.

Following feature extraction, the network performs:

- Spatial linearization, where the multi-dimensional feature representation $F_2 \in \mathbb{R}^{B\times 64\times 256\times 256}$ is flattened into a high-dimensional vector space.
- Dimensionality reduction via a fully-connected mapping function that projects from the high-dimensional feature space to the original watermark dimensionality:

$$w_{\text{pred}} = \sigma_{\text{sigmoid}}(W_{\text{fc}} \cdot F_{\text{flat}} + b_{\text{fc}}) \quad (6)$$

where F_{flat} represents the flattened feature representation, W_{fc} and b_{fc} denote the fully-connected layer parameters, and σ_{sigmoid} is the sigmoid activation function that normalizes the output to the range $[0, 1]$.

The decoder architecture is deliberately designed with several characteristics that enhance its robustness against image distortions:

- **Sparse Parameterization**: The network employs only two convolutional layers followed by a single fully-connected layer, minimizing overfitting risk while maintaining sufficient capacity for watermark extraction.
- **Spatial Invariance Properties**: The convolutional operations inherently provide some degree of translation invariance through weight sharing and local receptive fields.
- **Global Context Integration**: The flattening operation and subsequent fully-connected layer enable the network to leverage global feature relationships across the entire spatial domain, which is particularly important when watermarks are distributed throughout the image.
- **Output Normalization**: The sigmoid activation function in the final layer produces probabilistic values in the range $[0, 1]$, facilitating threshold-based binary watermark recovery and providing resilience against amplitude variations.

This architecture demonstrates that effective watermark extraction can be achieved with relatively simple network configurations when the feature representation is appropriately designed for the task. The decoder complements the encoder's embedding strategy by focusing on robust feature extraction that maintains watermark recoverability even under significant image transformations.

2.4. Saliency Net

The SaliencyNet module constitutes a critical component within the proposed watermarking framework, implementing a specialized visual attention mechanism that facilitates perceptually-informed watermark embedding. This neural architecture performs hierarchical feature extraction and transformation to generate spatial saliency maps that guide the subsequent watermark embedding process.

The SaliencyNet employs a compact convolutional architecture optimized for perceptual region identification. The network initializes with a 3×3 convolutional layer that transforms the tri-channel input image ($I \in \mathbb{R}^{B\times 3\times H\times W}$) into a 16-dimensional feature space. This initial transformation can be expressed as:

$$F_1 = \sigma(W_1 * I + b_1) \in \mathbb{R}^{B\times 16\times H\times W} \quad (7)$$

where W_1 and b_1 represent the convolutional weights and biases, respectively, $*$ denotes the convolution operation, and σ is the ReLU activation function that introduces non-linearity to capture complex visual patterns. This multi-channel representation encodes various low-level features including intensity gradients, textural patterns, and local contrast variations that contribute to visual saliency.

The second stage of the network performs dimensionality reduction through another 3×3 convolutional operation that projects the 16-dimensional feature representation to a single-channel saliency map:

$$S = \sigma_{\text{sigmoid}}(W_2 * F_1 + b_2) \in \mathbb{R}^{B\times 1\times H\times W} \quad (8)$$

where σ_{sigmoid} represents the sigmoid activation function that normalizes the output to the range $[0, 1]$. This normalization is crucial as it transforms the raw saliency values into a probabilistic representation where higher values (approaching 1) correspond to regions with greater perceptual significance, while lower values indicate areas that are less visually salient and thus more suitable for watermark embedding.

The Saliency Net's design principles are aligned with established models of human visual attention, where certain image regions naturally attract fixation based on their distinctive features, contextual importance, or semantic relevance. By computationally modeling this attentional mechanism, the network identifies regions that would be most sensitive to visual artifacts: typically areas containing high-frequency details, structural elements, or objects of semantic significance.

The generated saliency map S serves as a spatial modulator for the watermarking process, creating an inverse relationship between perceived visual importance and the embedding strength of the watermark. watermark. When integrated with the encoder network, the saliency map guides the spatial distribution of watermark information through the complementary modulation:

$$W_{\text{guided}} = W_{\text{map}} \odot (1 - S) \tag{9}$$

This formulation ensures that the embedding of the watermark is attenuated in visually critical regions (where S approaches 1) and amplified in areas of perceived less significant (where S approaches 0). This adaptive approach significantly differs from traditional watermarking techniques that apply uniform embedding strategies across the entire image, often resulting in perceptible artifacts in visually important regions.

The deliberately streamlined architecture of Saliency Net—utilizing only two convolutional layers with appropriate padding to maintain spatial dimensions—balances computational efficiency with perceptual accuracy. This design choice is particularly relevant in real-time watermarking applications where computational resources may be constrained, yet perceptual quality must be maintained at high standards.

The Saliency Net thus represents a crucial perceptual guidance mechanism within the watermarking framework, ensuring that the imperceptibility requirement is addressed through content-adaptive embedding that preserves the visual integrity of perceptually significant image regions while maximizing watermark robustness in less visually important areas.

2.5. Discriminator

The discriminator component of the proposed framework implements an adversarial learning paradigm inspired by Generative Adversarial Networks (GANs), functioning as a learned perceptual metric that enforces visual fidelity between original and watermarked images. This network instantiates a binary classification architecture optimized for distinguishing subtle artifacts introduced during the watermarking process.

The discriminator employs a progressive feature extraction and dimensionality reduction strategy through a series of strided convolutional operations. This approach can be formalized as:

$$F_1 = \phi_{\text{LeakyReLU}}(W_1 *_s I + b_1) \in \mathbb{R}^{B\times 64\times \frac{H}{2}\times \frac{W}{2}} \tag{10}$$

where $*_s$ denotes a strided convolution with $s = 2$, implementing simultaneous feature extraction and spatial downsampling. The LeakyReLU activation function $\phi_{\text{LeakyReLU}}(x) = \max(0.2x, x)$ introduces non-linearity while mitigating potential gradient vanishing issues during adversarial training—a critical consideration for stable GAN dynamics.

The second convolutional layer further refines the representation while continuing the spatial dimension reduction:

$$F_2 = \phi_{\text{LeakyReLU}}(\beta(W_2 *_s F_1 + b_2)) \in \mathbb{R}^{B\times 128\times \frac{H}{4}\times \frac{W}{4}} \tag{11}$$

where β represents the batch normalization operation that stabilizes training through input normalization at each layer.

Following the convolutional feature extraction stages, the network performs global feature integration through a flattening operation:

$$F_{\text{flat}} = \text{flatten}(F_2) \in \mathbb{R}^{B\times 128\cdot \frac{H}{4}\cdot \frac{W}{4}} \tag{12}$$

This flattened representation is subsequently projected to a single scalar value:

$$D(I) = \sigma(\mathbf{w}_{fc}^T F_{\text{flat}} + b_{fc}) \in \mathbb{R}^{B\times 1} \tag{13}$$

where σ represents the sigmoid activation function that normalizes the output to the interval [0,1].

The discriminator functions within the theoretical framework of two-player minimax games:

$$\min_G \max_D \mathbb{E}_{I\sim p_{\text{data}}}[\log D(I)] + \mathbb{E}_{I\sim p_{\text{data}}}[\log(1 - D(G(I, w)))] \tag{14}$$

where G represents the encoder network and D is the discriminator.

The discriminator serves as a learned perceptual metric specifically tuned to detect watermarking artifacts. By incorporating adversarial loss into the watermarking objective function, the framework optimizes for imperceptibility in a perceptually meaningful space rather than relying solely on pixel-wise distortion metrics.

The relatively shallow discriminator architecture represents

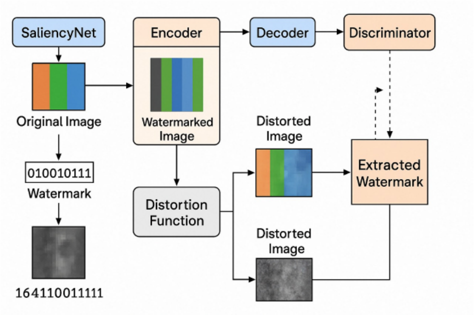

Figure 1. Model Architecture

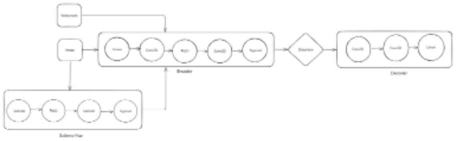

Figure 2. Network Architecture

Attack	LSB+ Patchwork	Frequency Domain(DCT)	Auto Encoders	Advanced Deep Learning (Saliency, Adversarial Training)
Rotation (e.g., small angles like 5–15 degrees)	Very Poor	Poor to Moderate	Poor to Moderate	Moderate to Good
Rotation (e.g., larger angles like 90 degrees)	Fails	Often Fails	Often Fails	Can sometimes recover with specific training
Flipping (Horizontal/Vertical)	Poor	Moderate	Moderate	Good
Gaussian Noise (e.g., low variance)	Poor to Moderate	Moderate to Good	Good	Good to Excellent
Gaussian Noise (e.g., high variance)	Poor	Moderate	Moderate to Good	Good

Figure 3

a deliberate design choice that balances discriminative power with training stability. The incorporation of batch normalization and LeakyReLU activation functions further contributes to training stability, addressing recognized challenges in GAN optimization dynamics.

Through this adversarial mechanism, the framework elevates beyond traditional watermarking approaches that rely on predetermined distortion metrics, instead learning an adaptive perceptual metric that continuously evolves to identify and minimize the most visually relevant artifacts.

3. Towards Our Model

experiments are in the table in figure 3.

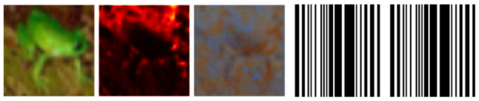

Figure 4. Example 1

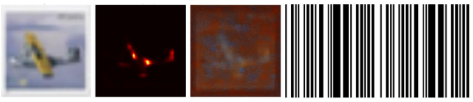

Figure 5. Example 2

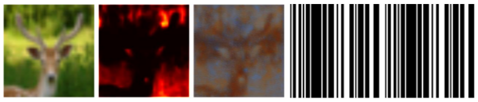

Figure 6. Example 3

4. Appendix

4.1. Code

The complete implementation of our proposed saliency-aware watermarking framework—including model definitions, training pipeline, and evaluation routines—is available at the following Google Colab link:

Full Code (Google Colab)

This notebook contains: data preprocessing, saliency network, encoder-decoder architecture, adversarial training, distortion simulation, and visual result generation.

4.2. Dataset

We use the publicly available CIFAR-10 dataset, which consists of 60,000 natural images across 10 classes. It can be accessed at:

CIFAR-10 Dataset

We use the *torchvision.datasets.CIFAR10* API to download and load the dataset, and resize all images to 256×256 for compatibility with our encoder network.