STEGAFLOW : END-TO-END DEEP NEURAL NETWORKS FOR IMAGE DATA HIDING AND STEGANOGRAPHY

Muhammad Romadhona Kusuma[1], Muhammad Naufal Adnansyah Nuryadin[2], Rahmat[3], Leo Ternado[4]
[1] Doctoral Program in Informatics, Universitas Nusa Mandiri, Indonesia
[2] Department of Informatics, Faculty of Science and Technology, Al-Azhar University, Indonesia
[3] Information Systems Study Program, Sekolah Tinggi Teknologi Indonesia Tanjungpinang, Indonesia
[4] Department of Communication Science, University of Riau, Indonesia
m.romadhona.kusuma@gmail.com

ABSTRACT

Deep neural networks are highly sensitive to subtle pixel-level variations, a property that is commonly discussed in the context of adversarial learning. Rather than treating this characteristic solely as a vulnerability, this study explores its potential for image data hiding. This paper proposes StegaFlow, an end-to-end deep learning-based framework for image steganography and blind watermarking, in which encoder and decoder networks are jointly optimized to embed binary messages into digital images while preserving visual quality and enabling reliable message recovery. To improve robustness against real-world image degradations, the training process incorporates multiple distortion models, including Gaussian blur, pixel-wise dropout, image cropping, and JPEG compression. Because JPEG compression is inherently non-differentiable, differentiable approximations are introduced during training to enable effective gradient-based optimization. Experimental results evaluate the proposed framework in terms of capacity, secrecy, and robustness, and show that it achieves strong performance under common image distortions. In particular, the framework demonstrates improved robustness to several spatial-domain degradations, while adversarial training contributes to better perceptual quality by reducing visible embedding artifacts. These findings indicate that end-to-end neural optimization provides a flexible and effective approach for robust image data hiding.

Keywords: *Image Data Hiding, Steganography, Deep Neural Networks, Blind Watermarking, Convolutional Neural Networks*

1. INTRODUCTION

Digital images can convey information beyond what is directly perceptible to human observers. In recent years, embedding hidden information into digital images has attracted growing attention due to the rapid expansion of online content sharing, increasing concerns over intellectual property, and the widespread proliferation of manipulated digital media. In this context, two major paradigms have been extensively studied, namely steganography and digital watermarking. Steganography aims to conceal the existence of secret messages within images, whereas digital watermarking emphasizes robustness, by ensuring that the embedded information can still be recovered even after the image undergoes common distortions such as compression, cropping, or blurring [1], [2]. These characteristics make watermarking particularly important for copyright protection, content authentication, and digital ownership verification in modern multimedia systems [3], [40].

Recent advances in deep learning have significantly influenced research in image data hiding. Deep neural networks are known to be highly sensitive to subtle and imperceptible perturbations in image data, a phenomenon widely investigated in the context of adversarial examples [4], [5]. More recent studies have shown that such perturbations may remain effective even after common image transformations, including compression and other real-world distortions [6]. Although this sensitivity is often regarded as a vulnerability in vision-based systems, it also presents an opportunity for image data hiding. Carefully optimized perturbations can be used to encode meaningful information while preserving perceptual image quality, making deep learning a promising approach for both steganography and watermarking.

Motivated by this perspective, this study proposes an end-to-end deep learning framework for image data hiding that can be applied to both steganography and blind watermarking scenarios. For clarity and ease of reference, the proposed framework is hereafter referred to as StegaFlow. StegaFlow jointly optimizes an encoder network to embed binary messages into cover images and a decoder network for reliably recovering the hidden messages. In addition, an adversarial discriminator is incorporated to improve imperceptibility by reducing detectable embedding artifacts, following recent adversarial learning-based data hiding strategies [7], [8]. To simulate realistic transmission and post-processing conditions, several noise modeling layers are introduced during training, including Gaussian blur, pixel-wise dropout, cropping, and JPEG compression.

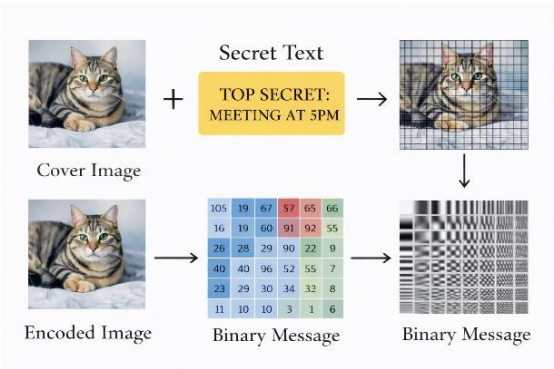

Figure 1. Illustration of secret message embedding into a cover image

The encoder embeds a binary message into a cover image to generate an encoded image that remains visually indistinguishable from the original while still enabling reliable message reconstruction by the decoder.

Figure 1 illustrates the conceptual workflow for embedding a secret text message into a cover image. First, the secret message is converted into a binary representation. The binary data are then embedded into the image domain to produce an encoded (stego) image. The resulting image is designed to preserve visual similarity to the original cover image while retaining sufficient information for accurate recovery by the decoder. This process highlights the transformation of textual information into binary form and its seamless integration into the visual content.

The learning objective of the proposed framework is to balance three complementary criteria: preserving visual similarity between the cover and encoded images, accurately reconstructing the embedded message, and reducing the detectability of hidden information. Based on these criteria, the model is evaluated along three key dimensions, namely capacity, secrecy, and robustness. Capacity refers to the amount of information that can be embedded in an image, secrecy reflects resistance to detection by modern steganalysis techniques [9], and robustness measures the reliability of message recovery under various image distortions. Recent studies suggest that these objectives are inherently interdependent and often involve trade-offs, particularly between embedding capacity and perceptual imperceptibility [10].

Conventional image data hiding methods generally rely on manually designed heuristics, such as spatial-domain bit manipulation or frequency-domain embedding based on transform techniques. Although these methods remain effective in certain scenarios, they are relatively static and often require substantial redesign to address new types of distortions. In contrast, learning-based approaches offer greater flexibility because they directly optimize task-specific objectives during training. Recent studies have demonstrated that end-to-end neural optimization can produce more adaptive and robust image data hiding strategies, especially under diverse and unpredictable image degradation conditions [7], [11], [12].

2. RELATED WORK.

This section reviews prior studies relevant to the proposed framework, with particular emphasis on deep learning–based image data hiding, adversarial perturbations, and robust watermarking. The discussion is organized to show how recent advances in adversarial learning and neural network-based embedding strategies have influenced the development of modern steganography and watermarking methods. By positioning the proposed approach within the context of existing literature, this section establishes its methodological foundation and highlights several limitations that are addressed in this study

2.1. Adversarial Examples and Image Perturbations

Adversarial examples demonstrate that deep neural networks can be highly sensitive to carefully crafted and visually imperceptible perturbations applied to input images. These perturbations are typically generated by optimizing pixel-level modifications that significantly alter network predictions while preserving the perceptual similarity to the original image [16], [17]. Several studies have shown that adversarial perturbations generated for one neural network architecture may transfer to other models, suggesting that such perturbations exploit common vulnerabilities shared across deep learning systems. Moreover, recent research indicates that adversarial examples may remain effective even after common image transformations, such as compression, printing, and recapturing [18].

Although adversarial perturbations are often examined as a security threat, recent perspectives have highlighted their constructive potential. Rather than inducing misclassification, optimized perturbations can be designed to encode structured information that can later be recovered by a trained model. This insight provides an important conceptual basis for image data hiding, in which information is embedded through controlled perturbations while maintaining perceptual image quality [13].

Figure 2 illustrates the overall architecture of the proposed deep learning-based image data hiding system. The framework takes a cover image and a binary message as input, which are then processed by the encoder network to generate a stego image containing the embedded information. To preserve perceptual quality, an adversarial discriminator is employed to minimize the visual discrepancy between the cover image and the stego image while

preserving the embedded payload. The stego image is then passed through a distortion simulation module that applies common perturbations, including Gaussian blur, cropping, dropout, and differentiable JPEG approximation, to emulate realistic noise and attack conditions. Finally, the decoder network attempts to recover the hidden message from the distorted stego image. The effectiveness of the framework is evaluated using several criteria: embedding capacity, secrecy, robustness in terms of bit error rate (BER), and perceptual quality measured by PSNR and SSIM. Overall, this framework demonstrates how message embedding, distortion modeling, and message recovery can be integrated into a unified end-to-end learning architecture.

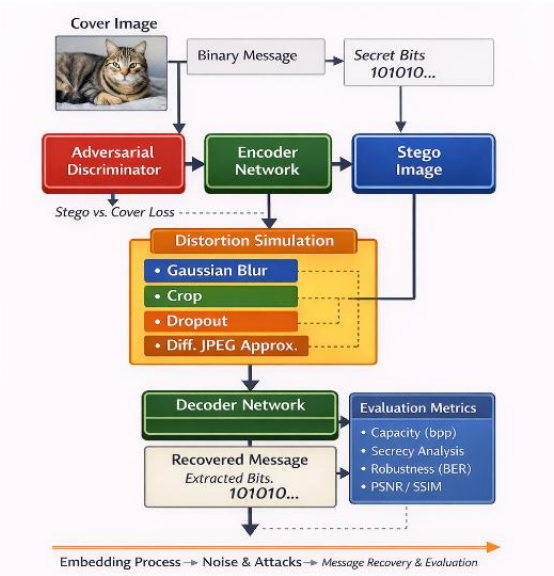

Figure 2. End-to-End Deep Learning Framework for Image Data Hiding, Distortion Simulation, and Message Recovery

2.2. Image Steganography

Image steganography aims to conceal the existence of secret information within digital images in such a way that unauthorized observers cannot reliably distinguish between original and encoded content. Early steganographic approaches primarily operated in the spatial domain, with least significant bit (LSB)-based methods among the most widely used techniques. Although LSB-based methods introduce changes that are generally imperceptible to the human eye, they often modify image statistics in predictable ways, making them vulnerable to detection by modern steganalysis techniques [21].

To overcome these limitations, more advanced steganographic methods have adopted distortion minimization strategies that adjust embedding strength according to local image characteristics. More recently, learning-based steganography approaches have increasingly employed adversarial training to improve both imperceptibility and robustness [13], [19], [20]. At the same time, modern steganalysis methods have also advanced significantly, with many approaches relying on deep convolutional neural networks to detect subtle statistical inconsistencies introduced during the embedding process [21]. This development highlights the continuing arms race between steganographic embedding methods and steganalysis techniques.

2.3. Digital Watermarking

Digital watermarking shares conceptual similarities with steganography in that both involve the embedding of information into digital media. However, the two differ fundamentally in their primary objectives. While steganography focuses on secrecy, watermarking emphasizes robustness [14], [40], ensuring that the embedded information remains recoverable even after the image undergoes various forms of modification. Watermarking methods are commonly categorized into non-blind and blind schemes. Among these, blind watermarking methods are particularly attractive for real-world deployment because they do not require access to the original cover image during the decoding process [14].

Recent studies in digital watermarking increasingly exploit frequency-domain representations, such as the discrete cosine transform (DCT) and wavelet-based techniques, to improve robustness against compression and filtering operations [22], [23], [36], [37] . In addition, hybrid approaches that combine spatial- and frequency-domain features have been proposed to enhance resilience against geometric distortions. Despite these advances, many conventional watermarking techniques still rely on manually designed embedding rules, which limit their adaptability to evolving attack scenarios [36], [37].

2.4. Deep Learning Based Data Hiding

In recent years, deep learning has emerged as a powerful alternative to conventional heuristic-based data hiding methods [15], [35], [38]. Neural networks have been applied to both steganography and watermarking, initially as components within larger processing pipelines, such as for estimating embedding strength or improving decoding accuracy [15]. However, these early approaches generally optimized only individual stages of the data hiding process rather than learning the entire embedding and recovery workflow in an integrated manner.

More recent studies have introduced end-to-end neural frameworks that jointly learn message embedding, distortion modeling, and message recovery. In many of these approaches, adversarial training is incorporated to improve imperceptibility by encouraging encoded images to remain visually indistinguishable from clean images [13], [19].

Some studies focus primarily on maximizing embedding capacity, including approaches that hide entire images within other images, while placing relatively less emphasis on robustness [19], [20]. By contrast, recent research has increasingly emphasized robustness-aware learning, particularly in blind watermarking scenarios where encoded images may be subjected to unknown distortions before decoding [22], [23].

Beyond image-based data hiding, neural networks have also been explored in related domains, such as embedding ownership information directly into neural network parameters and concealing information within textual data using neural language models [24]. These developments further demonstrate the versatility of deep learning techniques for information hiding across multiple modalities.

3. METHOD.

This section presents StegaFlow, the proposed end-to-end deep learning framework for image data hiding, which is designed to support both steganography and blind watermarking under various image distortion conditions. The main objective of StegaFlow is to learn a robust embedding strategy that preserves visual quality while enabling reliable message recovery after common image manipulations.

3.1. Model Overview

StegaFlow consists of four main components: an encoder Eθ, a noise simulation layer N, a decoder Dφ, and an adversarial discriminator Aγ, where θ, φ, and γ denote the trainable parameters of the corresponding networks.

The encoder Eθ receives a cover image Ico ∈ R^(C×H×W) and a binary secret message Min ∈ {0,1}^L of length L. It then produces an encoded (stego) image Ien with the same spatial dimensions as the cover image. The encoded image is subsequently passed through a noise layer N, which simulates image distortions and generates a distorted image Ino. Based on this distorted image, the decoder Dφ reconstructs the embedded message and outputs Mout.

In parallel, the adversarial discriminator Aγ receives either a clean cover image Ico or an encoded image Ien and produces a probability score A(Ĩ) ∈ [0,1], indicating whether the input image contains hidden information. This adversarial component encourages the encoder to generate encoded images that remain visually indistinguishable from clean images. Together, these components form a unified end-to-end architecture that jointly learns message embedding, distortion-aware transmission, and reliable message recovery. An overview of StegaFlow is shown in Figure 3.

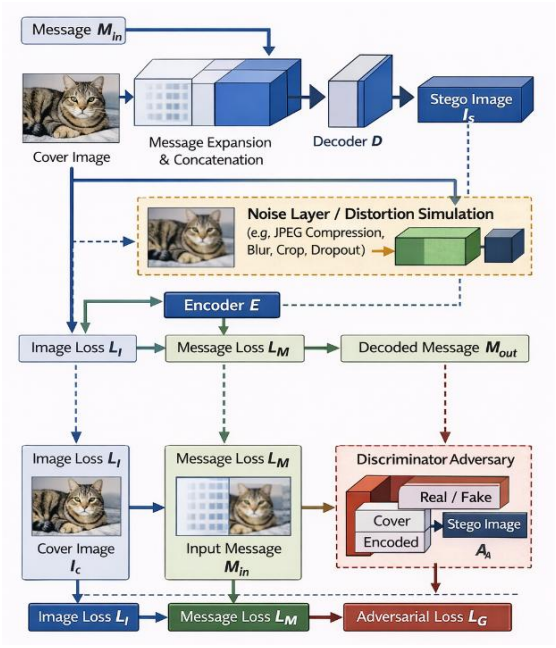

Figure 3. Detailed End-to-End Architecture for Deep Learning-Based Image Data Hiding with Adversarial Training.

Figure 3 presents a detailed view of StegaFlow, the proposed end-to-end deep learning-based image data hiding framework with adversarial training. In this architecture, the input message Min is first expanded and concatenated with the cover image features, enabling the encoder to integrate the secret information into the image representation. The encoder then generates a stego image Is, which is designed to preserve the perceptual quality of the original cover image while carrying the embedded message imperceptibly.

After the encoding stage, the stego image is passed through a noise layer that simulates practical distortions commonly encountered during image transmission, storage, or processing, such as compression, blur, cropping, and dropout. The resulting distorted image is then fed into the decoder, which reconstructs the hidden message and produces the decoded output Mout. By incorporating the noise layer into the forward path, the model is trained to maintain message recoverability even under practical distortions, thereby improving robustness against common image perturbations.

In addition to the main embedding and decoding pipeline, the training process is guided by three complementary objective functions. The image loss LI constrains the visual difference between the cover image and the stego image in order to preserve visual fidelity. The message loss LM minimizes the discrepancy between the input message and the decoded message to ensure reliable recovery. Furthermore, an adversarial discriminator is employed to distinguish cover images from stego images, producing the adversarial loss LG, which

encourages the encoder to generate stego images that remain visually indistinguishable from natural images. Collectively, these components form a unified optimization framework that balances imperceptibility, robustness, and decoding accuracy.

3.2. Loss Functions and Optimization

The learning objective balances three complementary criteria: visual imperceptibility, message reconstruction accuracy, and resistance to detection.

a. Image Reconstruction Loss

Visual similarity between the cover image and the encoded image is enforced using a normalized l_2image distortion loss:

$$L_I(I_{co}, I_{en}) = \frac{\| I_{co} - I_{en} \|_2^2}{CHW}$$

b. Message Reconstruction Loss

To ensure accurate recovery of the hidden message, the message distortion loss is defined as:

$$L_M(M_{in}, M_{out}) = \frac{\| M_{in} - M_{out} \|_2^2}{L}$$

c. Adversarial Loss

To reduce detectability, an adversarial loss penalizes the encoder when the discriminator successfully identifies an encoded image:

$$L_G(I_{en}) = \log(1 - A(I_{en}))$$

The discriminator is trained using the following binary classification loss:

$$L_A(I_{co}, I_{en}) = \log(1 - A(I_{co})) + \log(A(I_{en}))$$

The encoder and decoder parameters θ and ϕ are optimized by minimizing:

$$\mathbb{E}_{I_{co}, M_{in}}[L_M + \lambda_I L_I + \lambda_G L_G] \quad (1)$$

while the discriminator parameters γ are optimized by minimizing:

$$\mathbb{E}_{I_{co}, M_{in}}[L_A] \quad (2)$$

where λ_I and λ_G control the relative importance of image fidelity and adversarial objectives.

3.3. Network Architecture

The architectural design of the proposed framework is illustrated in Fig. 2. The encoder first applies a series of convolutional layers to the cover image to generate an intermediate feature representation. To incorporate the secret message, the binary message vector is spatially replicated and concatenated with the intermediate feature maps. This design allows each convolutional filter to access the entire message at every spatial location.

After additional convolutional layers, the encoder produces the encoded image I_en. The noise layer then applies a randomly selected distortion to generate the distorted image I_no, which may differ in spatial dimensions to accommodate operations such as cropping.

The decoder processes I_no through multiple convolutional layers, producing an intermediate representation with L feature channels. Global spatial average pooling is applied to obtain a fixed-length vector independent of the input image size, followed by a linear layer that outputs the reconstructed message. The discriminator adopts a similar convolutional structure but outputs a single probability value for binary classification.

3.4. Noise Layer Modeling

To improve robustness against real-world image manipulations, the proposed framework incorporates multiple noise layers that simulate common image distortions. These noise layers are designed to expose the model to a variety of degradation patterns during training, thereby improving its ability to recover embedded messages under practical distortions. Examples of these distortions are illustrated in Fig. 4 and Fig. 5.

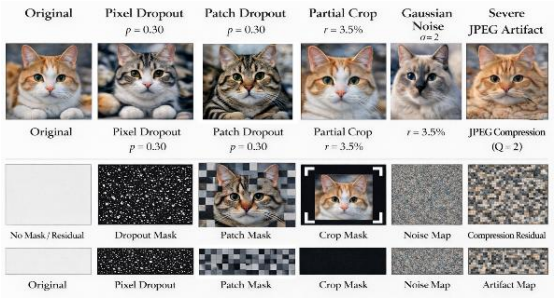

Figure 4. Visual Comparison of Image Distortions and Corresponding Residual/Mask Representations.

Figure 4 illustrates several representative image distortions, including pixel dropout, patch dropout, partial crop, Gaussian noise, and JPEG compression. The corresponding residual or mask maps highlight the spatial distribution of perturbations, showing how different distortion types affect both structural content and frequency characteristics.

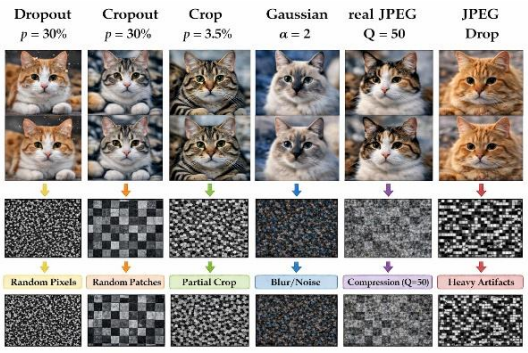

Figure 5. Comparative Analysis of Synthetic and Real-World Image Degradations

Figure 5 presents a comparison of multiple degradation types, including dropout, cropout, crop, Gaussian blur, real JPEG compression, and severe

JPEG artifacts. The top row displays the encoded images Ien, while the second row shows the corresponding noisy images Ino after the application of different distortion models. The bottom row illustrates the amplified absolute difference 15|Ien - Ino|, highlighting the spatial- and frequency-domain perturbations introduced by each noise layer. These visualizations demonstrate the differences between synthetic masking operations and compression-induced structural distortions across both the spatial and frequency domains. In general, spatial distortions mainly affect localized regions, whereas frequency-domain distortions tend to attenuate high-frequency components. Nevertheless, the proposed model remains capable of recovering the embedded message with high accuracy, as discussed in Section 4.

Six types of noise layers are considered:

a. Identity, which leaves the encoded image unchanged.
b. Dropout, where each pixel is randomly selected from either the encoded image or the cover image with probability p.
c. Cropout, which replaces pixels outside a randomly selected square region with pixels from the cover image.
d. Crop, which extracts a random spatial crop retaining a fraction pof the original image.
e. Gaussian Blur, which applies Gaussian smoothing with kernel width σ.
f. JPEG Compression, which applies lossy compression with quality factor Q.

All non-identity noise layers include a scalar parameter controlling distortion strength. Dropout, Cropout, and Crop primarily affect the spatial domain, while Gaussian blur and JPEG compression mainly operate in the frequency domain. Granting the noise layer access to the original cover image increases the difficulty of message recovery, encouraging the decoder to learn robust, location-invariant representations.

By varying the message length L, image dimensions (C$_m$,H$_m$,W), noise types, and loss weights λ_I and λ_G, the proposed framework can be flexibly adapted to both steganography and watermarking scenarios. Specific experimental configurations are discussed in Section 4.

3.5. Differentiable JPEG Approximation

JPEG compression is inherently non-differentiable because of its quantization step, which makes it unsuitable for direct gradient-based optimization. To address this limitation, the proposed framework employs differentiable approximations of JPEG compression during training, while robustness is evaluated against standard JPEG compression during inference [25].

In standard JPEG compression, an image is divided into 8 × 8 blocks, transformed into the frequency domain using the discrete cosine transform (DCT), and then quantized, with higher-frequency coefficients generally subjected to more aggressive quantization [26]. To simulate this process in a differentiable manner, two noise layers are introduced, namely JPEG-Mask and JPEG-Drop, as illustrated in Fig. 6 and Fig. 7.

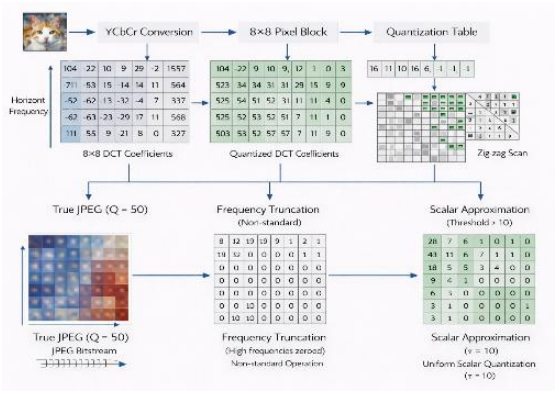

Figure 6. Frequency-Domain Interpretation of JPEG Compression and Non-Standard Coefficient Manipulations

Figure 6 illustrates the JPEG compression pipeline in the frequency domain, including YCbCr conversion, 8 × 8 DCT transformation, quantization, and zig-zag scanning. It also compares standard JPEG compression (Q = 50) with artificial frequency truncation and scalar approximation, highlighting the differences between standard quantization and non-standard coefficient manipulation strategies.

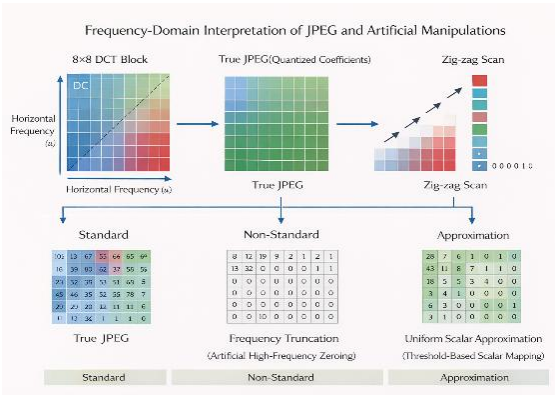

Figure 7. Conceptual Comparison Between Standard JPEG and Artificial Frequency Manipulation in the DCT Domain

Figure 7 presents the conceptual basis of the differentiable JPEG approximations used in this study. Standard JPEG compression applies a DCT to 8 × 8 image blocks followed by frequency-dependent quantization. Because the quantization stage is non-differentiable, two differentiable approximations are employed during training. The first, JPEG-Mask, suppresses high-frequency DCT

coefficients using a fixed masking strategy. The second, JPEG-Drop, applies frequency-dependent dropout, in which higher-frequency coefficients are more likely to be removed. Both approximations enable gradient-based optimization while preserving robustness against true JPEG compression.

More specifically, JPEG-Mask preserves low-frequency DCT coefficients while suppressing higher-frequency components, thereby simulating the loss of fine detail introduced by JPEG quantization. In contrast, JPEG-Drop applies progressively increasing dropout to DCT coefficients, assigning higher drop probabilities to coefficients associated with stronger quantization in standard JPEG compression. In both cases, information flow through high-frequency channels is restricted, allowing the model to learn representations that remain stable under compression-induced degradation.

Models trained using these differentiable approximations exhibit strong robustness against actual JPEG compression at test time, as demonstrated in Fig. 8.

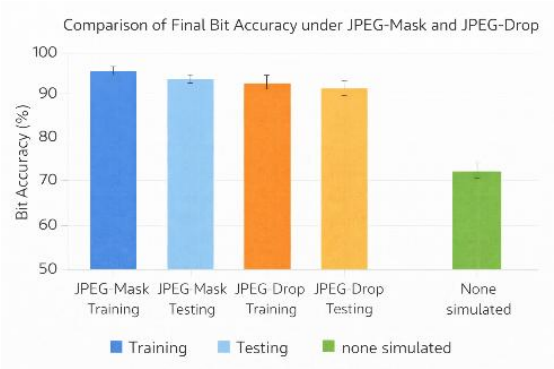

Figure 8. Final bit accuracy under JPEG-Mask and JPEG-Drop distortion models.

Figure 8 compares the final bit accuracy achieved under JPEG-Mask and JPEG-Drop distortion strategies during training and testing. Dashed lines represent training with simulated JPEG compression, whereas solid lines indicate evaluation under true JPEG compression with quality factor Q = 50. The results show that both distortion-aware training strategies improve robustness relative to the non-simulated baseline. In particular, models trained with either JPEG-Mask or JPEG-Drop maintain strong bit recovery performance under actual JPEG compression at test time.

In the implementation of JPEG-Mask, only 25 low-frequency DCT coefficients are preserved in the luminance (Y) channel and 9 coefficients are preserved in the chrominance (U and V) channels, following the general structure of standard JPEG compression, while the remaining coefficients are set to zero. By comparison, JPEG-Drop applies frequency-dependent dropout, such that coefficients that would normally undergo stronger quantization in true JPEG compression are more likely to be suppressed during training. Both strategies produce models that generalize effectively to real JPEG compression.

3.6. Implementation Details

All models are trained using 10,000 cover images sampled from the MS COCO dataset [27], which is selected due to its high visual diversity and complex scene composition. The images are resized to match the dimensions required for each experiment. Evaluation is performed on a separate test set of 1,000 images not seen during training. Binary messages are generated by independently sampling each bit from a uniform distribution.

Training is conducted using the Adam optimizer [28] with a learning rate of 10^-3 and default hyperparameters. A batch size of 12 is used in all experiments. Models are trained for 200 epochs, or for 400 epochs when multiple noise layers are incorporated during training.

4. EXPERIMENTS

This section evaluates StegaFlow with respect to the three fundamental dimensions of image data hiding: capacity, secrecy, and robustness. Capacity refers to the amount of information that can be embedded in each image, secrecy measures the difficulty of distinguishing encoded images from clean images, and robustness assesses the reliability of message recovery under various image distortions. Taken together, these dimensions provide a comprehensive basis for assessing the performance of the proposed method. Through this evaluation, the effectiveness of StegaFlow is analyzed from both perceptual and functional perspectives.

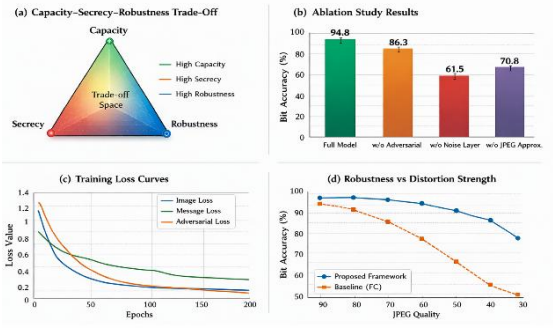

Figure 9. Comprehensive evaluation of the capacity–secrecy–robustness trade-off and robustness performance.

A comprehensive evaluation of the proposed framework is presented in Figure 9. Panel (a) illustrates the theoretical trade-off among capacity, secrecy, and robustness. Panel (b) reports the results of the ablation study, highlighting the contribution of key components within the proposed architecture. Panel (c) shows the convergence behavior of the

training losses over the course of optimization, while panel (d) evaluates robustness under increasing levels of distortion. Taken together, these results demonstrate the stability, effectiveness, and resilience of the proposed method under diverse distortion conditions.

4.1. Evaluation Metrics

Several metrics are employed to quantify performance across these dimensions.

Capacity is measured using bits per pixel (BPP), defined as the ratio between the embedded message length and the total number of pixels in the encoded image:

$$\text{BPP} = \frac{L}{H \times W \times C'},$$

following the notation introduced in Section 3.

Secrecy is primarily evaluated using the detection rate, obtained by training a modern steganalyzer to distinguish between cover and stego images. Specifically, we adopt the ATS steganalyzer [29] and report its test-set classification accuracy. As an auxiliary indicator of secrecy, we also report the peak signal-to-noise ratio (PSNR) between cover and encoded images, since larger image distortions generally increase detectability.

Robustness is quantified using bit accuracy, defined as the fraction of correctly recovered message bits between the original message Mand the decoded message M^, after applying image distortions between the encoder and decoder during testing.

These metrics inherently involve trade-offs. Increasing capacity typically reduces secrecy due to larger image distortions, while improving robustness often requires redundant encoding that may reduce capacity or secrecy. In this context, steganography and watermarking represent two extremes of a continuous spectrum, with steganography prioritizing secrecy and watermarking emphasizing robustness. Owing to its flexible design, the proposed framework allows different trade-offs by adjusting the message length L, noise layers, and training objectives.

4.2. Steganography: Capacity and Secrecy

To evaluate the steganographic performance of StegaFlow, the framework is trained to embed binary messages of length L = 52 into grayscale images of size 16 × 16, resulting in an embedding capacity of approximately 0.203 bits per pixel (BPP). For larger images, the message length is proportionally scaled to maintain a consistent embedding rate. Each message segment is embedded independently within non-overlapping 16 × 16 spatial patches. Since classical steganography assumes noiseless transmission, the noise layer N is set to the identity function during both training and testing.

Since classical steganography assumes noiseless transmission, the noise layer NNN is set to the identity function during both training and testing.

The proposed method is compared against several representative classical steganography techniques, including LSB Matching, HILL, and MiPOD, as well as a neural-network–based steganographic approach.

4.3. Image Quality

All methods are evaluated on 512×512512 \times 512512×512 grayscale images from the BOSS dataset, a widely used benchmark in steganography research. Although the proposed framework is trained on the COCO dataset, evaluation on BOSS demonstrates strong cross-dataset generalization. As illustrated in Fig. 6, both the proposed method and classical baselines generate stego images that are visually indistinguishable from the corresponding cover images, indicating high perceptual imperceptibility.

4.4. Capacity

The proposed framework achieves an embedding capacity comparable to classical baselines (0.203 BPP vs. 0.200 BPP). Although neural network–based approaches may introduce decoding errors, the observed bit error rate in our experiments remains below 10−510^{-5}10−5. When necessary, standard error-correcting codes can be incorporated to further enhance decoding reliability.

4.5. Secrecy

Secrecy is evaluated using the ATS steganalyzer. Following the recommended experimental protocol, ATS is trained on relatively small datasets consisting of 250 cover images and 250 stego images for each evaluated method. The steganalyzer constructs an artificial training set by repeatedly applying the corresponding steganographic algorithm to unlabeled images and subsequently trains a supervised classifier to distinguish between cover and stego images.

Table 1 reports the detection accuracy of the ATS steganalyzer under comparable embedding rates. Classical steganographic methods remain statistically detectable, with detection accuracies ranging from 69% to 76%. By contrast, the proposed model demonstrates strong secrecy when the trained parameters are not accessible, reducing detection accuracy to 50%, which is equivalent to random guessing. However, under a stronger threat model in which the attacker has full access to the trained model parameters, the detection accuracy increases to 98%, indicating that secrecy is substantially reduced when the exact embedding model is known.

Table 1. Detection Accuracy of the ATS Steganalyzer Across Different Embedding Strategies

Embedding Approach	Payload (BPP)	Reconstruction Error	ATS Detection Accuracy
LSB Matching	0.20	2.8×10^{-3}	76%
HILL	0.20	2.1×10^{-3}	71%
MiPOD	0.20	1.7×10^{-3}	69%
Proposed Model (full parameter access)	0.203	$< 1 \times 10^{-5}$	98%
Proposed Model (weights hidden)	0.203	$< 1 \times 10^{-5}$	50%

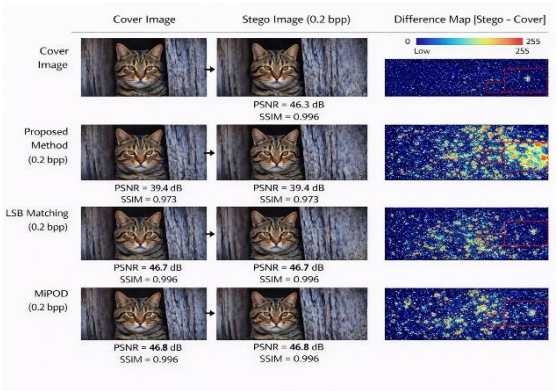

Figure 10. Visual and Quantitative Comparison of Steganographic Methods at 0.2 bpp

Figure 10 compares the visual characteristics of the proposed method, LSB Matching, and MiPOD at an embedding rate of 0.2 bits per pixel (bpp). The figure includes the reference image, the corresponding stego images, and amplified difference maps that visualize the perturbations introduced by each embedding strategy. PSNR and SSIM values are provided as quantitative indicators of perceptual similarity.

As shown in the figure, all methods produce stego images that remain visually similar to the reference image. However, the proposed method exhibits a more widely distributed perturbation pattern in the difference map and lower PSNR and SSIM values in this example compared with LSB Matching and MiPOD. This indicates that, although the proposed approach maintains acceptable visual quality, its embedding behavior differs from that of conventional methods and may involve a stronger distributed modification pattern. Nevertheless, the embedded information remains visually unobtrusive in the resulting image.

Taken together, the results in Table 1 and Figure 10 show that the proposed framework offers strong secrecy under a practical threat model in which the attacker does not have access to the exact trained weights. In this setting, the ATS detection accuracy drops to the level of random guessing. At the same time, the visual comparison confirms that the embedding remains perceptually acceptable, even though the residual distribution differs from that of conventional steganographic methods.

4.6. Comparison with Neural Network-Based Steganography

The proposed framework is further compared with a representative neural steganography approach based on fully connected networks. Unlike the baseline method, which relies primarily on global feature encoding, the proposed convolutional architecture exploits local spatial correlations to achieve more robust and efficient message embedding.

In addition to improving embedding efficiency, the convolutional design also contributes to more stable message reconstruction under higher payload conditions. By leveraging local spatial feature continuity within the image domain, the proposed framework is able to preserve decoding accuracy more effectively than fully connected neural steganography approaches.

As summarized in Table 2, the proposed method achieves substantially lower decoding error while supporting higher embedding capacity.

Table 2. Decoding Error Rate at Different Embedding Rates

Method	Embedding Rate (BPP)	Decoding Error
Proposed Framework	0.10	$< 1 \times 10^{-5}$
Proposed Framework	0.20	$< 1 \times 10^{-5}$
Fully Connected Neural Steganography	0.10	2.3×10^{-3}
Fully Connected Neural Steganography	0.20	4.1×10^{-3}

The results indicate that the proposed framework maintains near-perfect message reconstruction even when the embedding rate is doubled from 0.10 to 0.20 bpp. In contrast, the fully connected neural steganography baseline exhibits substantially higher decoding errors, particularly at the higher embedding rate. This performance gap highlights the advantage of convolutional feature learning, which enables more stable, spatially consistent, and robust message encoding.

Figure 11 presents a visual comparison of the proposed method and several representative steganographic approaches at an embedding rate of 0.2 bpp. For each method, the figure shows the reference image, the corresponding stego image with PSNR and SSIM values, and the amplified difference map. The comparison highlights the

differences in residual distribution patterns and perceptual quality across the evaluated methods.

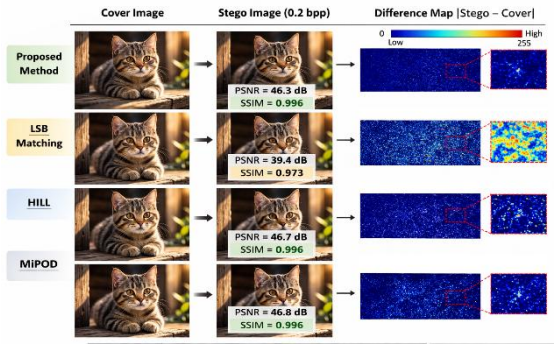

Figure 11. Visual and Quantitative Comparison of Steganographic Methods at 0.2 bpp

From a qualitative perspective, the proposed framework demonstrates a distinct embedding pattern while preserving the main visual structure of the image. When considered together with the quantitative results in Table 2, these observations indicate that the proposed method achieves improved embedding efficiency and stronger reconstruction performance relative to the neural baseline, even at higher payload levels.

4.7. Watermarking: Robustness Evaluation

Unlike steganography, digital watermarking prioritizes robustness rather than secrecy or embedding capacity. To evaluate the robustness of StegaFlow, the proposed models are trained on YUV color images with dimensions 3 × 128 × 128 and a message length of L = 30. The loss weights are set to $\lambda I = 0.7$ and $\lambda G = 0.001$.

To examine the effect of distortion-aware training, six specialized models are trained using different noise layers: Dropout, Cropout, Crop, Gaussian Blur, JPEG-Mask, and JPEG-Drop. In addition, a Combined model is trained by randomly sampling one of these noise layers for each mini-batch. For reference, an Identity model trained without any noise layer is also included.

As an external baseline, the proposed framework is compared with a conventional frequency-domain watermarking system. Because conventional watermarking methods typically report binary decoding success rather than bit-level reconstruction accuracy, direct comparison is not always straightforward. Therefore, the comparison methodology used in this study is described separately in Appendix B.

From a qualitative perspective, the proposed framework produces encoded images that remain visually similar to the original cover images across all evaluated distortion settings. In addition, adversarial training contributes substantially to perceptual quality, as evidenced by the comparison between models trained with and without the discriminator.

Table 3. Mean PSNR (dB) Between Cover and Encoded Images Under Different Noise Models

Model	PSNR (Y)	PSNR (U)	PSNR (V)
Traditional Frequency-Domain Watermarking	60.84	39.12	51.37
Identity (no noise)	44.63	45.44	46.90
Dropout	42.48	38.75	41.22
Cropout	47.10	41.05	42.01
Crop	35.46	33.58	36.02
Gaussian Blur	40.72	42.11	43.06
JPEG-Mask	30.34	35.41	36.58
JPEG-Drop	29.12	32.83	33.79
Combined	33.88	39.10	39.66

Table 3 reports the mean PSNR values between the cover and encoded images across the Y, U, and V channels under different noise-aware training configurations. The results show that perceptual quality varies depending on the distortion model used during training. In general, stronger robustness-oriented noise modeling tends to reduce PSNR, reflecting the trade-off between imperceptibility and robustness.

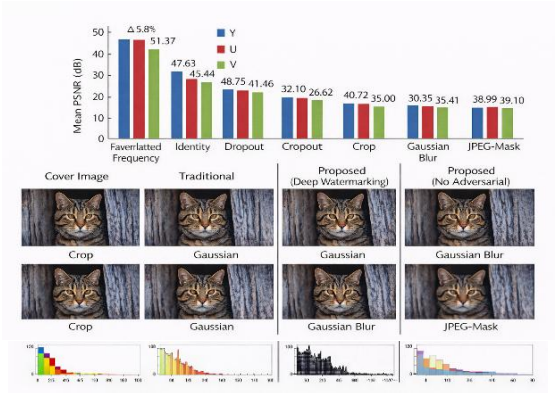

Figure 12. PSNR Analysis and Visual Robustness Evaluation under Multiple Distortion Conditions

Figure 12 presents both quantitative and visual analyses of encoded image quality under various distortion settings. The upper panel reports the mean PSNR values across the Y, U, and V channels for the conventional frequency-domain watermarking baseline and the proposed framework trained with different noise layers. The lower panels provide visual comparisons between the cover images and the encoded images generated by the conventional baseline and the proposed method under Crop, Gaussian Blur, and Combined distortion settings. The bottom-right example shows an encoded result from a model trained without an adversarial discriminator, illustrating the role of adversarial learning in improving perceptual quality and reducing visible embedding artifacts.

Overall, the results indicate that the proposed framework maintains strong visual fidelity while improving robustness under a range of distortions,

including crop-based, blur-based, and JPEG-related degradations. These findings support the effectiveness of distortion-aware training for robust blind watermarking.

4.8. Robustness Analysis

Figures 13 and 14 show the bit accuracy of the proposed framework under various distortion types and distortion intensities. Models trained without noise exhibit limited robustness when exposed to image degradations, particularly under cropping and JPEG compression, where decoding accuracy drops to nearly chance level (50%).

In contrast, specialized models trained with a specific noise layer demonstrate strong robustness against their corresponding distortions, even when the applied distortion is non-differentiable. For example, bit accuracy under true JPEG compression improves from approximately 50% to around 85% when the model is trained using a differentiable JPEG approximation.

Figure 13 illustrates the robustness of different models under multiple test-time distortions. The Identity model is trained without any distortion layer, the Specialized models are trained using the same distortion applied during testing, and the Combined model is trained using all distortion types. The results show that the Combined model achieves competitive performance across all evaluated distortions, indicating strong generalization while maintaining high robustness.

Figure 14 presents bit accuracy as a function of distortion intensity. Star markers indicate the distortion strength used during training. Notably, the specialized JPEG model is trained using the differentiable JPEG-Mask approximation, whereas evaluation is performed using true JPEG compression. This result highlights the effectiveness of differentiable approximations in enabling robustness against real-world non-differentiable distortions.

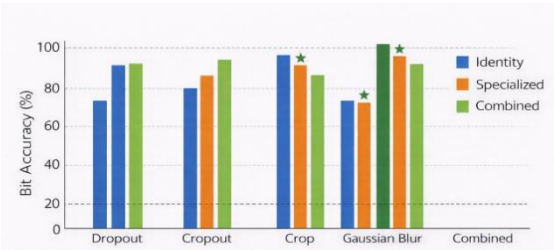

Figure 13. Bit Accuracy Comparison Across Distortion Types and Training Strategies

Figure 13 compares bit accuracy under several distortion types, including Dropout, Cropout, Crop, and Gaussian Blur, using three training strategies: Identity, Specialized, and Combined. The results indicate that the Combined training strategy generally provides stronger robustness across multiple distortion types, demonstrating improved generalization compared with single-distortion training.

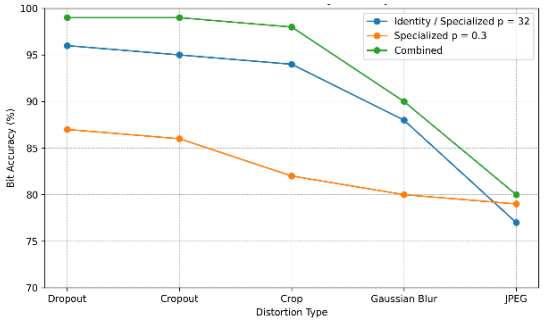

Figure 14. Robustness evaluation across multiple distortion types.

Figure 14 shows bit accuracy under Dropout, Cropout, Crop, Gaussian Blur, and JPEG compression as a function of distortion strength. The figure compares the performance of the Identity, Specialized, and Combined training strategies across different distortion intensities. Overall, the Combined strategy demonstrates more stable performance and stronger generalization across diverse distortion settings. In the JPEG case, the specialized model is trained using the differentiable JPEG-Mask approximation, while evaluation is conducted using true JPEG compression, further confirming the practical effectiveness of the proposed distortion-aware training approach.

4.9. Comparison with a Conventional Frequency-Domain Watermarking System

To enable a fair comparison, we first estimate the effective embedding capacity of a conventional frequency-domain watermarking system and apply error-correcting codes to match the bit rate of the proposed framework. Based on this calibration, a bit accuracy of $\geq 95\%$ is treated as equivalent to successful decoding, whereas a bit accuracy of $\leq 90\%$ is considered decoding failure.

Figure 15 presents the performance of the proposed deep learning-based watermarking framework and a conventional frequency-domain watermarking baseline under various distortion types and distortion intensities. Four settings are compared: the Identity model trained without noise, the Specialized models trained with a specific distortion, the Combined model trained using all distortion types, and the decoding success rate of the conventional watermarking system for 256 × 256 images. To facilitate comparison, the two y-axes are scaled to translate bit accuracy into full reconstruction rate, following the methodology described in Appendix B.

The results show that the proposed deep watermarking framework substantially outperforms the conventional frequency-domain baseline under severe spatial-domain distortions, such as Dropout and Crop. For example, under Dropout distortion (p

= 0.1), the Specialized model achieves a bit accuracy of ≥ 95%, whereas the conventional watermarking system fails to reliably decode the embedded information. Similarly, under Crop distortion (p = 0.1), both the Specialized and Combined models maintain bit accuracy above 95%, while the conventional method is unable to successfully recover the watermark.

By contrast, the conventional frequency-domain watermarking system demonstrates stronger performance under frequency-domain distortions, particularly JPEG compression. This behavior is expected because classical transform-domain watermarking techniques embed watermark signals directly into frequency coefficients, making them inherently more robust to compression-based degradations [36], [37], [39]. In comparison, the proposed deep learning-based framework does not impose explicit architectural constraints related to frequency-domain transforms, but instead relies on data-driven optimization to learn distortion-resilient embedding strategies.

Overall, these findings reveal complementary strengths between conventional and deep watermarking paradigms. While transform-domain watermarking remains advantageous under compression-based distortions, the proposed deep watermarking framework exhibits stronger robustness under spatial-domain degradations, including dropout, cropping, and localized perturbations. This adaptability highlights the potential of deep neural watermarking to generalize across diverse and unpredictable distortion scenarios without requiring handcrafted embedding rules.

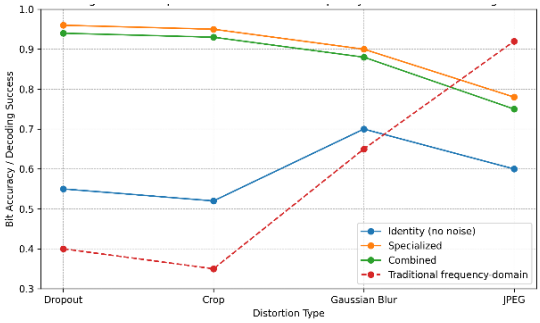

Figure 15. Comparison with Conventional Frequency-Domain Watermarking under Distortion

To evaluate robustness under different distortion conditions, Figure 15 compares the proposed training strategies with a conventional frequency-domain watermarking approach under several distortions, including Dropout, Crop, Gaussian Blur, and JPEG compression. The results show that the Combined strategy maintains stronger and more stable bit accuracy across multiple distortion conditions, whereas the conventional frequency-domain method exhibits increased decoding instability under severe compression and other challenging degradations.

5. CONCLUSIONS

This paper presented StegaFlow, an end-to-end deep learning-based framework for image data hiding that supports both steganography and blind watermarking. By jointly optimizing message embedding, distortion-aware transmission, and message recovery, StegaFlow is able to preserve visual quality while maintaining reliable decoding performance under a range of image distortions. Experimental results showed that the proposed framework achieves strong performance in terms of capacity, secrecy, and robustness. In the steganography setting, StegaFlow maintained competitive embedding capacity and strong secrecy under a practical threat model, particularly when the exact trained weights were not accessible to the attacker. In the watermarking setting, distortion-aware training significantly improved robustness against spatial-domain degradations such as dropout and cropping, while differentiable JPEG approximations improved resilience against compression-related distortions.

Comparative evaluations further showed that StegaFlow outperforms representative neural steganography baselines in decoding accuracy and embedding efficiency, while also demonstrating stronger robustness than conventional frequency-domain watermarking systems under severe spatial-domain distortions. At the same time, classical transform-domain methods remained more effective under frequency-domain degradations such as JPEG compression, indicating complementary strengths between deep learning-based and conventional watermarking paradigms. Overall, these findings suggest that StegaFlow provides a flexible and effective framework for robust image data hiding across diverse distortion conditions. Future work may focus on larger image resolutions, improved frequency-aware architectural design, and stronger robustness against more complex real-world distortions and adaptive steganalysis attacks.

Acknowledgments

The authors would like to express their sincere gratitude to colleagues and collaborators from various institutions and affiliations for their valuable discussions, insights, and constructive feedback throughout the development of this work. We also thank the anonymous reviewers for their thoughtful comments and suggestions, which helped improve the quality and clarity of the manuscript.

REFERENCES

[1] Y. Zhang, J. Liu, and T. Luo, “Robust image steganography based on adversarial learning,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 31, no. 8, pp. 3041–3053, 2021.

[2] X. Wen, Y. Chen, and Y. Fang, “Blind watermarking using deep neural networks with

adversarial training," Signal Processing: Image Communication, vol. 96, 2021.
[3] M. Barni, F. Bartolini, and A. Piva, "Watermarking and deep learning: A survey," IEEE Signal Processing Magazine, vol. 39, no. 2, pp. 80–97, 2022.
[4] Y. Zhu, S. Liu, and J. Wu, "Trade-off analysis of robustness and imperceptibility in deep watermarking," Multimedia Tools and Applications, vol. 81, pp. 25671–25689, 2022.
[5] H. Wang, Y. Chen, and Z. Zhang, "Deep learning-based robust image watermarking against common distortions," IEEE Access, vol. 10, pp. 113245–113257, 2022.
[6] A. Singh and R. S. Chadha, "A comprehensive review of deep learning techniques for image steganography," Journal of Information Security and Applications, vol. 74, 2023.
[7] S. Baluja and S. Fischer, "Learning to hide images in images," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 42, no. 7, pp. 1685–1697, 2021.
[8] R. Tancik, B. Mildenhall, and R. Ng, "StegaStamp: Invisible hyperlinks in physical photographs," Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021.
[9] J. Hayes and G. Danezis, "Generating steganographic images via adversarial training," Advances in Neural Information Processing Systems (NeurIPS), 2017.
[10] Y. Zhang, W. Zhang, and X. Liu, "Adversarial learning for image steganography," IEEE Signal Processing Letters, vol. 26, no. 7, pp. 983–987, 2019.
[11] H. Yuan, J. Tan, Y. Huang, R. He, X. Huang, and Y. He, "Recent advances in adversarial attacks and defenses in computer vision: A survey," ACM Computing Surveys, vol. 54, no. 4, pp. 1–36, 2021, doi: 10.1145/3437875.
[12] N. Carlini et al., "On evaluating adversarial robustness," IEEE Symposium on Security and Privacy, pp. 39–56, 2022.
[13] N. Akhtar, M. A. A. K. Jalwana, M. Bennamoun, and A. Mian, "Attack to fool and explain deep networks," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 44, no. 10, pp. 5980–5995, 2022, doi: 10.1109/TPAMI.2021.3073005.
[14] Q. Xiao, K. Li, D. Zhang, and W. Xu, "Adversarial attacks on deep learning models in computer vision: A survey," Artificial Intelligence Review, vol. 56, pp. 4563–4607, 2023, doi: 10.1007/s10462-022-10261-1.
[15] J. Zeng, S. Tan, B. Li, and J. Huang, "Large-scale JPEG steganalysis using deep convolutional neural networks," IEEE Transactions on Information Forensics and Security, vol. 16, pp. 3784–3798, 2021.
[16] G. Xu et al., "Structural design of convolutional neural networks for steganalysis," IEEE Signal Processing Letters, vol. 23, no. 5, pp. 708–712, 2016.
[17] J. Ye, J. Ni, and Y. Yi, "Deep learning hierarchical representations for image steganalysis," IEEE Transactions on Information Forensics and Security, vol. 12, no. 11, pp. 2545–2557, 2017.
[18] I. Richardson, The H.264 Advanced Video Compression Standard, 2nd ed., Wiley, 2021.
[19] H. Deng, F. Chen, P. Gan, R. Liao, and X. Yan, "A Robust Image Watermarking Scheme via Two-Stage Training and Differentiable JPEG Compression," Electronics, vol. 14, no. 22, p. 4510,2025, doi: 10.3390/electronics14224510.
[20] K. Hu, M. Wang, X. Ma, J. Chen, X. Wang, and X. Wang, "Learning-based image steganography and watermarking: A survey," Expert Systems with Applications, vol. 249, p. 123715,2024,doi:10.1016/j.eswa.2024.123715
[21] T.-Y. Lin et al., "Microsoft COCO: Common Objects in Context," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 43, no. 6, pp. 1871–1886, 2021.
[22] Y. Yousfi, J. Butora, and J. Fridrich, "Breaking ALASKA: Color separation for steganalysis in JPEG domain," in IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2020, pp. 256–257.
[23] L. Chen, Y. Xie, and H. Su, "Adaptive optimization methods for deep learning: A review," *IEEE Access*, vol. 11, pp. 31512–31535,2023,doi:10.1109/ACCESS.2023.3261458.
[24] C. Yang, X. Liu, and C. Wang, "A robust blind watermarking scheme based on deep neural networks," Multimedia Tools and Applications, vol. 79, pp. 21703–21719, 2020.
[25] L. Chen et al., "Robust image watermarking with deep neural networks," Neurocomputing, vol. 415, pp. 273–284, 2020.
[26] J. Zhang et al., "Deep learning-based image watermarking against geometric attacks," Signal Processing, vol. 173, 2020.
[27] M. Volkhonskiy et al., "Steganographic GANs," Computer Vision and Image Understanding, vol. 214, 2022.
[28] H. Hu et al., "Robust image watermarking using deep neural networks and perceptual loss," IEEE Access, vol. 9, pp. 134214–134225, 2021.
[29] X. Zhong et al., "Image watermarking using deep convolutional networks," Neural Computing and Applications, vol. 32, pp. 13303–13316, 2020.
[30] Y. Fang et al., "Deep neural network-based blind watermarking robust to JPEG compression," Signal Processing: Image Communication, vol. 88, 2020.

[31] J. Guo, Y. Zhang, X. Liu, and Y. Xiang, "Deep learning for image steganography and steganalysis: A survey," ACM Computing Surveys, vol. 56, no. 3, pp. 1–38, 2023, doi: 10.1145/3581785.

[32] M. Chaumont and F. Pibre, "Deep learning for image steganography and steganalysis: Current trends and challenges," IEEE Multimedia, vol. 30, no. 2, pp. 72–84, 2023.

[33] Z. Zhang, Y. Qian, X. Zhang, and S. Tan, "Coverless information hiding: A survey," IEEE Access, vol. 10, pp. 56960–56982, 2022, doi: 10.1109/ACCESS.2022.3175624.

[34] Y. Luo, X. Tan, and Z. Cai, "Robust Deep Image Watermarking: A Survey," Computers, Materials & Continua, vol. 81, no. 1, pp. 133–160, 2024, doi: 10.32604/cmc.2024.055150.

[35] Y. Liu, M. Chen, Y. Wang, F. Zhang, and X. Zhao, "Deep learning-based information hiding: A review," IEEE Access, vol. 9, pp. 145954–145972, 2021.

[36] M. R. Kusuma and S. Panggabean, "Robust digital image watermarking using DWT, Hessenberg, and SVD for copyright protection," IJACI: International Journal of Advanced Computing and Informatics, vol. 2, no. 1, pp. 41–52, 2026.

[37] B. Song, P. Wei, S. Wu, Y. Lin, and W. Zhou, "A survey on Deep-Learning-based image steganography," *Expert Systems with Applications*, vol. 254, p. 124390, 2024, doi: 10.1016/j.eswa.2024.124390.

[38] K. M. Hosny, A. Magdi, O. ElKomy, and H. M. Hamza, "Digital image watermarking using deep learning: A survey," *Computer Science Review*, vol. 53, p. 100662, 2024, doi: 10.1016/j.cosrev.2024.100662.

[39] M. R. Kusuma and E. S. Bahri, "Securing NFT copyright with robust DWT-Hessenberg-SVD watermarking and RSA signatures," Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi), vol. 9, no. 6, pp. 1451–1462, 2026, doi: 10.29207/resti.v9i6.6747

[40] S. Sharma, J. J. Zou, G. Fang, P. Shukla, and W. Cai, "A review of image watermarking for identity protection and verification," *Multimedia Tools and Applications*, vol. 83, no. 11, pp. 31829–31891, 2024, doi: 10.1007/s11042-023-16843-3