UNIVERSITY
OF TECHNOLOGY
SYDNEY

Defending Security and Privacy of Image Data from Learning-Based Threats

by Yuan Zhao

Thesis submitted in fulfilment of the requirements for the degree of

Doctor of Philosophy

under the supervision of A/Prof. Bo Liu

University of Technology Sydney
Faculty of Engineering and Information Technology

November 2023

Certificate of Original Authorship

I, *Yuan Zhao*, declare that this thesis is submitted in fulfilment of the requirements for the award of Doctor of Philosophy, in the *School of Computer Science*, *Faculty of Engineering and Information Technology* at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Australian Government Research Training Program.

Production Note:
Signature removed prior to publication.

Signature: ____________________________

Date: 24th November, 2023

Place: Sydney, Australia

Acknowledgments

First and foremost, I would like to express my sincerest gratitude to my supervisors, Prof. Bo Liu and Prof. Tianqing Zhu, as well as my advisor Prof. Ming Ding. for their invaluable guidance and steadfast support throughout my PhD study. Their profound knowledge and abundant experience have consistently illuminated my path in research. In particular, I am indebted to Prof. Bo Liu, whose insightful advice has influenced not just my research, but also my life and career planning.

My heartfelt thanks also go to my circle of friends. Their friendship has been a wellspring of pleasure, helping me maintain a healthy balance with life beyond the demanding nature of research.

And I am also fortunate to have been a part of our dynamic research groups - Cyber Privacy& Safety and AI Security& Privacy. The knowledge-rich and inspiring discussions within these groups have been integral to my academic growth.

Last, but certainly not least, I would like to express my gratitude to my grandfather, parents, elder sister and other family members. Their love, inspiration, and unwavering support in every endeavour I undertake have been my constant source of motivation. Their faith in me has fueled my ambition, and for this, I am eternally thankful.

Yuan Zhao
Sydney, Australia
November, 2023

List of Publications

Publications of the author during the PhD candidature are listed in the following:

Related to the Thesis :

1. **Zhao, Y.**, Liu, B., Zhu, T., Ding, M., & Zhou, W. (2022). Private-encoder: Enforcing privacy in latent space for human face images. Concurrency and Computation: Practice and Experience, 34(3), e6548.

2. **Zhao, Y.**, Liu, B., Ding, M., Liu, B., Zhu, T., & Yu, X. (2023). Proactive Deepfake Defence via Identity Watermarking. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (pp. 4602-4611).

3. **Zhao, Y.**, Liu, B., Ding, M., Liu, B., Zhu, T., & Yu, X. Deep Proactive Watermark-based Image Manipulation Detection and Localization. under review in IEEE Transactions on Information Forensics and Security.

4. **Zhao, Y.**, Liu, B., Zhu, T. ROSIN: Robust Semantic Image Hiding Network. under review in IEEE Transactions on Image Processing.

Others :

5. Liu, C., Zhu, T., **Zhao, Y.**, Zhang, J., & Zhou, W. Multi-level model fingerprinting for task-specific forensics on GAN-generated images. under review in Pattern Recognition

6. Chen, H., Zhu, T., **Zhao, Y.**, Low-frequency Image Deep Steganography: Manipulate the Frequency Distribution to Hide Secrets with Tenacious Robustness. under review in IEEE Transactions on Dependable and Secure Computing.

Abstract

Learning-based techniques have revolutionized various media aspects, enabling content generation, editing, and personalization advancements. However, the widespread adoption of these methods has also given rise to significant security and privacy concerns. This thesis primarily focuses on two major threats posed by learning-based methods in image data: 1. surveillance and tracking and 2. malicious forgery and tampering. To tackle these threats, we adopt four straightforward and effective strategies: sensitive information sanitization, forgery media detection, media authenticity protection, and media authorship proof.

We first highlight the challenges arising from the ubiquitous use of learning-based image analysis techniques for surveillance and tracking. For instance, facial recognition algorithms, which have found wide-ranging applications, can track and identify individuals without their consent, compromising their privacy. To mitigate this threat, we propose a novel learning-based method for semantically sanitizing the face image's identity information in the generative network's latent space. It achieves a balanced trade-off between privacy protection and image utility preservation.

Regarding malicious forgery and tampering, we underscore the limitations of traditional forensic methods that primarily rely on detecting artifacts or distortions left by tampering, leading to poor detection accuracy and generalization performance. In response, we introduce an innovative framework that proactively defends against malicious forgery by watermarking face identity features and identity whether watermarked images have been forgeries according to the watermark existence. This novel detection mechanism solves the limitations of traditional forensic methods, thus providing a reliable measure of media content authenticity.

In line with the adage that prevention is better than cure, we also propose a proactive strategy for safeguarding media content at the pixel level. The designed scheme embeds an invisible watermark into a target image that is pixel-by-pixel entangled with it, which acts as an indicator of tampering trails. Once the watermarked image is counterfeited, the embedded watermark will exhibit changes accordingly, so we can locate the tampered regions by comparing retrieved and original watermarks. This proactive authentication mechanism makes our method effective against various image tamper techniques, including image copy&move, splicing and in-painting.

Lastly, in the realm of authorship proof, we emphasize the importance of preserving the rights of original content creators, thereby preventing plagiarism and copyright infringement. To this end, we introduce a novel method that uses the semantic information in images to boost the robustness of watermarks, thereby ensuring reliable attribution of authorship even under conditions of common distortions. The extensive experiment results validate the effectiveness of our design and demonstrate that the proposed method achieves superior robustness while maintaining comparable imperceptibility and capacity when compared to state-of-the-art techniques.

By exploring and applying the latest learning-based techniques, this thesis aims to fill critical research gaps in solving security and privacy concerns from learning-based methods to media data. Our work seeks to strike a delicate balance between fostering innovation and preserving user privacy and security. This research paves the way for developing secure, privacy-preserving approaches, thereby contributing to the ongoing evolution of media technologies.

Keywords: Face image de-identification, Deepfake detection, Image tampering detection, Image manipulation localization, Invisible watermark, Robust image steganography.

Table of Contents

List of Figures

List of Tables

Table **Page**

Chapter 1

Introduction

1.1 Backgrounds of the thesis

The contemporary digital age, marked by the rise of learning-based technologies, presents incredible opportunities and profound challenges. The extensive application of these technologies has transformed the media landscape, offering innovative methods for content generation, editing, and personalization. However, the flip side of this progress is the manifestation of significant security and privacy threats. Specifically, some of these threats are: 1. Surveillance and tracking, 2. Malicious forgery and tampering. This dichotomy forms the primary motivation for this thesis.

With the development of learning-based techniques, existing surveillance and tracking systems have become significantly powerful, enabling sophisticated pursuit operations. For example, some facial recognition algorithms can identify and track individuals without consent. This process has raised serious privacy concerns as these techniques can be deployed in public spaces or online, infringing upon people's privacy rights. However, the existing countermeasures, such as traditional image blurring or mosaic, have vulnerabilities in removing all sensitive information and are inadequate for preserving the image's utility.

Another motivation arises from the alarming increase in malicious forgery and tampering of image data. Highly realistic fake or tampered media can be easily created using learning-based techniques, leading to the spread of misinformation, identity theft, and privacy violations. The consequences of these forgeries are far-reaching, as they undermine trust in media content, impact journalism, politics, and social communication, and hinder the reliable dissemination of information. Traditional forensic methods that detect visual artifacts or distortions from tampering operations fall short in the face of evolving tampering techniques.

In response to these threats, the primary objective of this thesis is to develop effective strategies for defending the security and privacy of image data. By investigating and utilizing the latest learning-based techniques, this research designs robust and reliable methodologies that protect image data while ensuring its authenticity and integrity. These solutions include:

Sensitive Information Sanitization: Given the risk of sensitive information leakage from visual data, this thesis proposes a learning-based method for semantic sanitizing. The method aims to strike a balanced trade-off between privacy protection and utility preservation.

Forgery Detection: To counter the increasing prevalence of visual data manipulation or forgery, we design a proactive framework to combat malicious forgery by watermarking face identity features.

Authenticity Protection: A proactive approach towards safeguarding media content before malicious actions occur is essential. Our method aims to reduce the risk of data breaches and other privacy and security threats by proactively defending media content.

Authorship Proof: Given the ease with which visual data can be copied and distributed without proper attribution, establishing authorship is essential to prevent plagiarism and copyright infringement. We propose a novel method that leverages the semantic information in images to boost the robustness of watermarks, enabling reliable authorship attribution.

In summary, the motivation for this thesis stems from two key concerns: the enhanced capabilities of surveillance and tracking systems that infringe on privacy and the alarming rise in malicious forgery and tampering with media, leading to widespread misinformation and privacy violations. In response, the thesis aims to develop robust strategies for safeguarding the security and privacy of image data. Its objectives include:

- Designing learning-based methods to sanitize sensitive information achieves a balance between privacy and utility.
- Developing proactive frameworks for forgery detection and authenticity protection.
- Establishing authorship proof to combat plagiarism and copyright infringement.

By accomplishing these objectives, this research hopes to navigate the tension between leveraging AI innovations in media technologies and preserving user privacy and security, thereby contributing to developing secure, privacy-preserving approaches in the media technology domain.

1.2 Existing challenges

The threats posed by surveillance and tracking, and malicious forgery and tampering present complex challenges that must be addressed to ensure the privacy and security of image data. The current major challenges to solving these threats are:

1.2.1 Surveillance and Tracking

The widespread use of visual data, particularly on social media, can lead to breaches of user privacy. Sensitive information can be extracted from this data, making it vulnerable to misuse. To protect citizens' privacy, many governments have introduced regulations or laws, such as the General Data Protection Regulations (GDPR) in the European Union or the Australian Privacy Principles (APPs) in Australia. However, these data are also important sources for forming large-scale face image datasets, crucial for developing advanced computer vision techniques like face recognition or identity tracking systems. The above regulations will challenge the conventional use of images in computer vision research because they require consent from every person in the dataset. It is nearly impossible to obtain consent from every individual when using some large-scale image datasets like Celeba or FFHQ. What is worse, traditional obfuscation-based face privacy protection techniques, such as Mosaic or Blur, not only have vulnerabilities in removing sensitive information but are also inadequate for preserving the image's utility.

Therefore, the challenge lies in developing techniques that ensure privacy while preserving the quality and utility of visual data. Specifically, such a method should satisfy the following two criteria:

Anonymization: The method must remove the face image's identity features and reduce the possibility of re-recognition by both computer vision techniques and human observers.

Realism: The processed facial dataset should maintain a similar statistical distribution to the original, and individual images should retain high visual quality to ensure that the dataset remains useful for computer vision research.

1.2.2 Malicious Forgery and Tampering

Learning-based image generative or editing approaches, such as Generative Adversarial Networks (GANs) and Diffusion models, lead to powerful methods of synthesizing or manipulating visually authentic images/videos. Abusing these methods threatens visual information integrity and personal privacy security, e.g., by generating fake news or spreading rumours. The current detection methods are still in their infancy because they mainly rely on leveraging learning techniques by distinguishing feature distribution inconsistency or boundary discrepancy in an image to identify the forgery or any manipulation. Those methods assume that image manipulation techniques may inevitably produce detectable artifacts in their outputs. However, this prerequisite might lead to several inherent drawbacks.

- Various image post-processing operations can easily destroy these artifacts. As a result, detection methods developed to detect these artifacts would be failed when the suspect image is distorted.

- Forgery techniques are developed with an alarming speed, leaving fewer detectable artifacts in their synthesized results. Detection methods are thus struggling to keep up with the development of forgery techniques, making detection even more challenging.

- Atifact-based detection methods are difficult to generalize to unknown scenarios. These methods depend highly on the artifacts learned during training, so they exhibit poor performance in dealing with unknown and strange artifacts.

- Protecting the authenticity of media content before malicious tampering occurs is a crucial challenge. By the time tampering is detected, the damage has already been done. Implementing security measures that safeguard media content in advance can significantly reduce the risk of data breaches and privacy violations. However, the challenge is designing proactive methods resistant to tampering and distortion.

Besides, in the age of digital media, visual data can be easily copied and distributed, leading to plagiarism and copyright infringement. Traditional authorship-proof methods often prove insufficient when faced with distortions common in the digital landscape. Therefore, there is a significant challenge in designing robust methods to prove authorship and protect intellectual property rights.

Addressing these challenges is crucial for developing reliable solutions that can effectively safeguard privacy and security in an era marked by rapid advancements in learning-based methods for media.

1.3 Contributions of the thesis

This thesis makes several substantial contributions towards defending the security and privacy of image data from learning-based threats. The detailed contributions are summarized in the following subsections.

1.3.1 Sensitive Information Sanitization

The thesis presents a novel method for the semantic sanitizing sensitive information, detailed in the paper "A Learning-Based Method for Semantic Sanitizing". This approach sanitizes sensitive information in the neural network's latent space to balance privacy protection with utility preservation, enhancing security while retaining the quality and usefulness of the image data.

In summary, the major contributions of this work are summarized as follows:

- It develops a novel image privacy protection framework that implements the de-identification of face images via editing the identity-related features in the neural network's latent space.
- A dedicated and adjustable privacy-related loss function is designed in this work to regularize the network's training process.
- The proposed framework's superiority is demonstrated over traditional and advanced methods in privacy protection, visual quality, and utility.

1.3.2 Forgery Media Detection

The thesis introduces an innovative approach for detecting manipulated or forged visual data. This contribution, published in the paper "Proactive Defense against Malicious Forgery via Watermarking Face Identity Features", departs from traditional artifact-based detection methods. It presents a novel framework that embeds watermarks in face identity features, providing a robust means of determining media authenticity.

The main contributions of this work are summarized as follows:

- It proposes a novel proactive Deepfake detection method by embedding an anti-counterfeiting watermark into images' identity vectors.
- A simple yet effective encoder-decoder network is designed for invisible anti-Deepfake watermarking without needing pre-annotation or detection information.
- Extensive evaluations are conducted to prove the method's effectiveness, robustness, utility, and security. The experiment design can also serve as a template for similar research in the future.

1.3.3 Media Authenticity Protection

In "Proactive Media Authentication using Deep Learning Semi-Fragile Watermarks", a new method for media authentication is proposed. This method embeds an invisible watermark pixel-by-pixel into an image to pinpoint tampered regions. It effectively against various image tamper techniques, including image copy & move, splicing, and in-painting.

The contributions are summarized as follows:

- It features a novel deep learning-based semi-fragile image watermarking framework to defend against malicious tampering.
- The watermark balances detection performance and imperceptibility, ensuring no impact on real-world image usage.
- A comprehensive evaluation comparing the method with state-of-the-art detection methods to assess and analyze their performance across various aspects, including effectiveness, robustness and security.

1.3.4 Media Authorship Proof

The thesis also proposes a novel method that utilizes the semantic information in images to enhance watermark robustness. This solution, discussed in the paper "Robust Semantic Image-hiding for Authorship Proof", significantly boosts the resilience of watermarks to common distortions, effectively establishing reliable authorship attribution and protecting copyright.

In summary, the main contributions can be listed as follows:

- Development of a semantic image-hiding network highly robust against various distortions, enhancing image steganography practicability.
- Exploration of semantic features' redundancy for carrying hidden information and proof of identity feature invariance to conventional image distortions.
- Achievement of a balanced trade-off between capacity, imperceptibility, and robustness with comprehensive performance evaluation.

By addressing these threats and challenges, our research makes substantial strides in improving security and privacy within learning-based methods for media. Through sensitive information sanitization, forgery detection, authenticity protection, and authorship proof, we provide innovative solutions to the existing challenges, significantly contributing to the advancement of secure and privacy-preserving approaches in media technologies.

1.4 Overview of the thesis

This thesis is organized into seven main chapters detailing our research and findings on defending the security and privacy of image data from learning-based threats. Below is a brief description of what each chapter entails:

Chapter 1: Introduction. This chapter provides the background of our research, outlining the motivation, objectives, existing challenges, and contributions of this thesis.

Chapter 2: Related Work. This chapter reviews the current literature on the existing solutions in sensitive information sanitization, forgery media detection, media authenticity protection, and media authorship proof.

Chapter 3: Sensitive Information Sanitization. This chapter delves into our proposed solution for protecting sensitive information from privacy breaches. We present a novel semantic sanitizing method that leverages the neural network's latent space to balance privacy protection with utility preservation.

Chapter 4: Forgery Media Detection. In this chapter, we introduce an innovative approach for detecting manipulated or forged visual data. Our proposed framework uses watermarking of face identity features, a departure from traditional artifact-based detection methods.

Chapter 5: Media Authenticity Protection. Here, we detail our proactive media authentication method that embeds an invisible watermark entangled pixel-by-pixel with a target image. We discuss the design and effectiveness of our method against various image tampering techniques.

Chapter 6: Media Authorship Proof. This chapter focuses on our proposed solution for establishing reliable authorship attribution. We explain our novel method that leverages semantic information in images to enhance watermark robustness, effectively preventing plagiarism and copyright infringement.

Chapter 7: Conclusion. This final chapter summarizes the key findings of our research, discusses the implications of our work, and suggests directions for future research in defending the security and privacy of image data from learning-based threats.

By following this structure, the thesis provides a comprehensive exploration of the research topic, addressing the challenges, proposing novel techniques, and contributing to the existing knowledge in the field. The chapters collectively offer valuable insights and practical solutions for researchers, practitioners, and policymakers concerned with image data security, privacy, and authenticity in learning-based methods.

CHAPTER 2

LITERATURE REVIEW

2.1 Preface

This chapter presents a comprehensive review of the literature and existing research pertinent to the key areas this thesis aims to address: sensitive information sanitization, forgery media detection, image tamper detection, and media authorship proof.

The sensitive information sanitization section surveys the present landscape of privacy-preserving techniques, their methods, effectiveness, and the associated challenges. We critically examine traditional methods, such as Mosaic and Blur, and the latest learning-based approaches and discuss their limitations in preserving data utility while protecting privacy.

In the forgery media detection and image tamper detection sections, we delve into the current detection techniques, their mechanisms, and their performance against advanced tampering and forgery methods. The discussion encapsulates the limitations of relying on visual artifacts and distortion clues, emphasizing the need for more robust and generalized solutions.

The section on media authorship proof explores the existing strategies and techniques for establishing and preserving authorship. We discuss traditional methods, their efficacy, and their vulnerabilities against common distortions and unauthorized replication.

By examining the current state of research in these areas, this chapter establishes a solid theoretical foundation for our study and highlights the research gaps that this thesis

Table 2.1: Comparative Analysis of Sensitive Information Sanitization Methods.

Method Type	**Techniques**	**Advantages**	**Disadvantages**
Traditional	Mosaic	Simple to implement	Easily reversible; Low privacy protection;
	Blur	Widely used	Significant utility loss;
	Mask	Immediate obfuscation	Vulnerable to re-identification;
Learning-based **[18, 45, 63, 79, 97, 106, 137, 146]**	Attribute manipulation	Enhanced privacy and utility;	Dependency on attributes;
		Preserves some natural appearance;	Unnatural outcomes
	Facial Synthesis	High level of anonymization;	Dependency on attributes;
		Effective in removing identity while keeping context;	Unnatural outcomes
Differential Privacy [40, 141]		Provides mathematical privacy guarantee	Leads to distorted images; loss of utility

aims to address. The comprehensive review prepares the ground for the subsequent chapters, where we present our innovative solutions to these identified challenges.

2.2 Sensitive information sanitization

This work focuses on sanitizing sensitive information in human face images, also known as de-identification, which aims to hide the identity in the face image or video stream for privacy protection. Until recently, a limited number of research works existed, which are also listed in Table 2.1.

2.2.1 Traditional methods

Typically, the standard techniques for protecting face image privacy include Mosaic, Blur, Mask and Pixelation. However, these methods are increasingly seen as inadequate for emerging privacy needs. They protect privacy by directly perturbing images' Regions of Interest (ROIs) pixel values, which can effectively obfuscate corresponding sensitive information, i.e., identity feature in this context, but also incur conspicuous haziness in processed images, leading to significant utility loss [134]. Moreover, these traditional techniques have demonstrated significant vulnerabilities when facing advanced learning-based re-identification attacks [107]. MacPherson et al. [99] presented that faces obfuscated by the aforementioned techniques can be re-identified up to 96% by utilizing body or scene features from images.

2.2.2 Learning-based methods

Consequently, more sophisticated and novel concepts have been employed to enhance processed images' privacy and utility. For instance, Hui-Po et al. [137] and Tao et al. [79] obfuscated images' sensitive information by manipulating face attributes. The rationale of those methods is that facial attributes, such as hairstyle or eye colour, could be an essential reference for faces' identities. Therefore, changing these features, e.g., transforming eye colour from black to blue, appears reasonable for anonymization. Although such approaches render faithful processed images, they heavily rely on predefined attributes, which limits their applicability.

Other works [45, 63, 146] achieve de-identification by maximizing the distance of identity feature embedding through a dedicated designed dissimilarity loss term or multiple discriminators. Some methods [118, 125, 126] even mask the facial area and synthesize a new face via inpainting. These methods, though, often map the original face to a single anonymized counterpart, resulting in faces with similar appearances that often appear unnatural.

Fan [40] introduced calibrated Differential Privacy (DP) noises into the image's SVD features, offering a certain level of protection and ensuring indistinguishability among visually similar images. However, this method often results in overly distorted images, leading to a considerable loss of utility. Wen et al. [141] adopted a similar approach by adding noise to the identity vectors, thereby protecting image privacy, but separated image features into identity and non-identity categories.

Another series of works uses predefined conditional labels to control the content of generated images [97] or decouples images into different representations for manipulation [18, 106]. Despite these methods' ability to generate a variety of anonymized faces, they often alter identity-irrelevant attributes and introduce additional visual artifacts.

2.3 Forgery media detection

With the significant advancement of generative methods, high-quality synthetic images or videos are so realistic that they even can deceive the human eyes. Unfortunately, malicious users can exploit these techniques to create social disruption or political threats. Therefore, numerous detection approaches have been proposed to counteract such risks, primarily by identifying the artifacts introduced by the imperfections of learning-based forgery techniques. A comparative analysis is summarized in Table 2.2.

2.3.1 Low-Level Artifacts

GANs have seen significant advancements, now capable of synthesizing high-fidelity images that can fool the human eye. Despite these improvements, GANs still reveal disparities between generated and natural distributions due to the commonly used

Table 2.2: Comparative Analysis of Forgery Media Detection Methods

Analysis Type	Techniques	Advantages	Limitations
Low-Level Artifacts	GAN Fingerprints [34, 95, 152] Up-sampling artifacts [160]	Early detection of GAN-generated images; Identifies specific GAN flaws.	Less effective as GANs evolve; Becomes obsolete as GANs improve.
Spatial	Color distortion [98] Pixel co-occurrence matrices [101]	Effective against colour manipulation; Robust against various fakes.	Limited to specific types of forgery; May miss subtle forgeries; Computationally intensive.
Biological	Eye blinking, Head poses; [24, 50, 80, 148, 150]	Effective in video Deepfake detection; Exploits natural human characteristics.	Limited to human subjects; May not apply to still images.
Frequency	Azimuthally Averaged Spectrum [35] 2D-FFT, 2D-DCT [36, 152]	Effective in identifying frequency patterns; Robust against various image manipulations.	Requires sophisticated analysis techniques; May miss non-frequency based forgeries.
Proactive	Embedding invisible tags [5, 138, 151, 153]	Traceability of original content.	Doesn't help in tamper detection.

up-convolution (or deconvolution) operation [139], which maps low-resolution tensors to high-resolution ones. The pioneering work AutoGAN [160] first observes that this up-sampling artifact can be used to identify GAN-generated images. This discovery sparked other studies [95, 152] to explore the concept of GAN fingerprints for distinguishing between authentic and GAN-generated images. Later, Durall et al. [34] utilized GAN fingerprints for Deepfake attribution. However, as GANs steadily improve, their problems become short-lived, making it unsustainable to base detection mechanisms solely on these known issues. For example, spectral regularization is proposed [34] to close the gap in the spectral domain. Recently, Jung and Keuper [69] learned an additional discriminator with spectrum inputs using adversarial training to reduce the frequency gap further.

2.3.2 Detection with Spatial Analysis

Because low-level artifacts are unsustainable, other works analyze different spatial artifacts to attribute images as real or fake. For instance, McCloskey [98] first utilizes colour distortions to detect fake images. This approach was later built upon by Nataraj et al. [101], who developed a way to identify Deepfakes by assessing the combinations of pixel co-occurrence matrices. Furthermore, Liu et al. [89] design a network to induce texture representations using a Gram matrix and validate that global textures at different levels of a CNN are effective cues for fake detection. Similarly, the Laplacian of Gaussian (LoG) has been applied to foster fake image and video detection [96]. In addition to texture and colour distortions, some studies have targeted discrepancies across blending boundaries to distinguish manipulated faces [78]. To improve the detection performance, Dang [28] first adopt attention mechanisms on CNN models to detect forgery artifacts. Building upon this approach, Zhao *et al.* [163] reformulate the forgery detection as a fine-grained classification task and propose a new multi-attentional architecture to capture local discriminative features from multiple faces attentive regions. Yu et al.[154] introduced a commonality learning strategy to extract universal forgery features from different databases to better generalize in unknown forgery methods.

2.3.3 Detection with Biological Analysis

Except for the spatial artifacts, the biological signal artifacts are another obvious clue for the forge. This is because authentic facial images and videos, captured with cameras, usually appear more natural than their synthesized counterparts. Lyu [80] first proposes to spot Deepfake videos by observing the lack of eye blinking in the synthesized face. Taking a different approach, Yang et al. [150] utilized inconsistent head poses to uncover forged videos. FakeCatcher [24] combines six biological signals to distinguish natural and fake videos. Haliassos [50] targets the inconsistencies in mouth movements learned via lipreading to detect forged videos. Yang *et al.* [148] employs the multi-task learning scheme to extract more comprehensive and accurate lip features to gain more powerful fake discriminability. Last, according to a patch-level prediction from different stages of a CNN, it has been shown that hair and background are the most informative areas for detecting fake facial images [21], which may help detection across various data distributions.

2.3.4 Detection with Frequency Analysis

Frequency analysis, with its long history in image processing, is also a powerful tool for detecting fake images. Several recent methods based on analyzing frequency patterns of images are adapted to fake detection. For instance, Durall et al. [35] propose a simple yet effective method based on the azimuthally averaged spectrum magnitude and Support Vector Machines (SVM). Dzanic et al. [36] use 2D-FFT magnitudes as input features for binary classification through Convolutional Neural Networks (CNNs). In the same vein, Frank et al. [43] study 2D-DCT as CNN input features, yielding improved detection results compared to the image-based method by Yu et al. [152]. To date, the most advanced detection technique incorporates global and local 2D-DCT features [115], further underscoring the effectiveness of frequency analysis in detecting fake images. In a recent effort to improve the detector's generalization, Luo [92] combines high-frequency features of the image with colour textures to detect forgery. Li et al. [76] design a novel frequency-aware discriminative feature learning framework to reduce intra-class variation of natural faces while increasing interclass differences in the embedding space for face forgery detection. Moreover, Liu et al. [86] introduce a Spatial-Phase Shallow Learning (SPSL) method, which combines spatial image and phase spectrum to capture

Table 2.3: Comparative Analysis of Image Tamper Detection Methods

Method Type	Techniques	Advantages	Disadvantages

Traditional	Local Noise Analysis [25]
	CFA Artifacts [41]
	Illumination Variance [30]
	JPEG Compression Clues [83]
		Detects noise inconsistencies;
		Useful for detecting image origin;
		Exploits lighting inconsistencies;
		Effective on recompressed images.
			May fail with high-quality tampering;
			Limited to CFA-based camera images;
			Performance affected by image quality;
			Ineffective on uncompressed images.

Learning-Based	CNNs [9, 11, 60] RNNs [11, 12] GANs [65] Auto-encoders [14, 162] Multi-Task Learning [4, 60, 145, 166] Edge Artifact Analysis [32, 164]	High accuracy and adaptability; Good at capturing sequence-based features; Effective in generating and detecting fakes; Can learn complex tamper signatures; Captures a wide range of artifacts; Effective at boundary detection.	Requires large annotated datasets; Computationally intensive; May generate convincing fakes itself; Vulnerable to adversarial attacks; Complex to train and implement; May miss non-boundary tampering.

the up-sampling artifacts of face forgery, enhancing the transferability for face forgery detection.

2.3.5 Proactive Detection Measures

Meanwhile, several proactive measures [5, 138, 151] are being developed to combat malicious media forgery. These methods involve embedding an invisible tag into the original image, which remains retrievable after generation. This allows users to retrieve the tag and halt the spread of the manipulated media. For instance, Yu et al. [153] embed artificial fingerprints into the generative model and subsequently into its generated Deepfakes, facilitating detection based on the extracted fingerprints.

2.4 Image tamper detection

Developing image editing techniques makes tampered images widely available and more realistic. Currently, the research community defines three common types of image tampering, which are: Copy-move (i.e., copying and moving elements from one region to another region in a given image), splicing (i.e., copying elements from one image and pasting them on another image), and inpainting (i.e., removal of unwanted elements). All these manipulations could lead to misinterpretation of the visual content. Image manipulation detection aims at detecting and localizing these tamperings, and the recent related works are summarized in Table 2.3.

2.4.1 Traditional tamper detection methods

Initially, many studies in this field relied on hand-crafted or predetermined features such as local noise analysis [25], Colour Filter Array (CFA) artifacts [41], and illumination variance analysis [30].

For instance, early work by Lin et al. [83] utilized the statistics of Discrete Cosine Transform (DCT) coefficients of doubly compressed JPEG images to distinguish between authentic and tampered regions. Ferrara et al. [41] used the colour filter array to detect inconsistencies in an image's tampered regions. Further, local noise features introduced by sensors and post-processing [26, 94] and inconsistencies in illuminant colour or lighting [20, 39] served as cues for image splicing detection.

However, these hand-crafted features were typically designed for a specific type of image manipulation. Consequently, they often struggle to achieve high performance in practice due to their inherent limitations.

2.4.2 Learning-based tamper detection methods

Applying deep learning techniques has revolutionized various fields, including image manipulation detection. Researchers have harnessed the power of deep neural networks, such as Recurrent Neural Networks (RNNs) [11, 12], Convolutional Neural Networks (CNNs) [60], and Generative Adversarial Networks (GANs) [65], for this task. These techniques have improved the generalization capability of image tampering detection across different manipulation types.

A common approach in learning-based image tampering detection involves the estimation of local noise variances [109]. Different regions within an authentic image contain similar intrinsic noise variances, so tampering can be revealed by detecting inconsistencies in local noise variances. Notable works in this area include the proposition by Fridrich et al. [44] to use steganalysis to construct rich models of the noise component and capture numerous quantitative relationships between pixels. Li et al. [75] introduced using an FCN's first convolutional layer with trainable high-pass filters to capture tampering features.

In addition, Zhang et al. [162] employed a stacked autoencoder to learn context features, and Bayar et al. [14] replaced the low-pass filter layer with an adaptive kernel layer to learn the filtering kernel used in tampered regions. There have also been advancements in designing CNN structures to highlight local mosaic inconsistencies [9], exploiting interdependencies between patches [11], and using a hybrid CNN-LSTM network [12] to improve detection performance.

Several methods combine hand-crafted and learning features for image forensics, such as combining BayarConv and SRM features [60, 145], utilizing a two-stream network for detection [166], and merging a spatial domain CNN with a frequency domain CNN [4]. For instance, Wu et al. [145] and Hu et al. [60] use both BayarConv and SRM features as

Table 2.4: Comparative Analysis of Media Authorship Proof Methods

Method Type	**Techniques**	**Advantages**	**Disadvantages**
Traditional	LSB [68, 128]	Simple; easy to implement;	Vulnerable to steganalysis; low robustness.
	DCT [114]	Resilient to JPEG compression;	Can be removed by filtering and compression;
	DWT [6, 13, 133]	Robust against scaling and rotation.	Complex implementation.
Learning-Based	Deep Hiding [100, 167]	High capacity; good fidelity	Computationally intensive;
	GANs [87, 122, 135]	Generates robust watermarks; good at data hiding	Requires careful training; may generate detectable patterns.
	Adversarial Embedding [51, 112, 130]	Improves imperceptibility of hidden content	Complex model tuning required.
	INNs [67, 90, 147]	Reversible; high fidelity and capacity	Still an emerging field; robustness concerns.

detection clues. Given features from distinct views, they develop a two-stream network, which inputs the RGB image and its feature counterpart generated by the SRM filter to identify the tampered pixels. Amerini et al. [4] combines a spatial domain CNN with a frequency domain CNN for splicing forgery detection, inspired by the fact that single and double JPEG compression artifacts differ. Zhou et al. [165] combine SRM features with RGB features by a two-stream Faster R-CNN to perform manipulation detection.

Manipulating specific regions of an image inevitably leaves traces. Therefore, exploiting edge artifacts also contributes to manipulation detection. For instance, Salloum et al. proposed a multitask FCN to predict a tampered area and its boundary [121]. Zhou et al. [164] introduced an edge detection and refinement branch. MVSS-Net [32] replaced non-trainable bilinear pooling with Dual Attention and further concatenated edge features for adaptive prediction.

Proactive measures have also been put forward to combat malicious tampering [5, 138, 153]. These involve embedding an invisible tag into the original image, which remains retrievable post-manipulation, enabling users to block the dissemination of fake information. However, these methods cannot pinpoint the tampered region.

2.5 Media authorship proof

Advancements in information technology and easy internet access have made unauthorized access, alteration, and dissemination of digital information all too common. This unrestricted access to digital information, without any security mechanism, poses severe threats to content security, protection, and integrity. Consequently, digital watermarking and steganography have emerged to address these concerns, which are powerful techniques for data protection and copyright security of digital images over unlicensed usage. Related works are listed in Table 2.4.

2.5.1 Traditional Image Watermarking and Steganography

Traditional image steganography techniques can be broadly classified into three types: spatial-based, frequency-based, and adaptive-based steganography methods. The Least Significant Bit (LSB) [68, 128] is a conventional spatial domain-based method. It replaces the n least significant bits of the cover image with the most significant n bits of the secret image. However, the LSB algorithm often introduces texture-copying artifacts, especially in smooth regions of an image. Other spatial steganography methods are based on pixel value differencing (PVD)[108, 144], histogram shifting [116, 132], multiple bit-planes [64, 104], palettes [64, 105], etc.

In addition to LSB, there are many methods proposed to embed information in frequency domains, such as the discrete Fourier transform (DFT) domain [3, 120], discrete cosine transform (DCT) domain [59], and discrete wavelet transform (DWT) domain [6, 13, 133], etc. For instance, JSteg [114] embeds data into the LSBs of the host image's discrete cosine transform (DCT) coefficients. These methods offer better fidelity than LSB but suffer from limited embedding capacity and lack robustness.

Adaptive steganography typically employs a general framework for data embedding, decomposing the problem into embedding distortion minimization and data coding. A prominent example of this method was proposed by Pevny et al.[112], which utilizes the subtractive pixel adjacency matrix feature[111] and syndrome-trellis codes [42] for adaptive steganography. Several other adaptive methods [48, 49, 56–58, 74] have been designed with different cost functions. While these methods offer good imperceptibility, they commonly fall short in payload capacity.

2.5.2 Learning-based Image Watermarking and Steganography

Recent developments in deep learning have seen its application in image-watermarking and steganography [100, 167], yielding impressive results. These methods, also known as deep hiding, started gaining attention with the work of Baluja [7], one of the first learning-based solutions to conceal an entire RGB image within another. They adopt a preparation network to extract useful features of the secret image and then employ a hiding network to fuse the features of the secret image within the cover image. Finally, a revealing network is adopted to recover the original secret image. Based on this approach, subsequent learning-based image-hiding techniques have emerged and can be categorized into four subclasses: the family by synthesis, the family by regulation, the family by adversarial embedding, and the family by invertible networks.

In the family of image synthesis, Shi et al. [122] and Volkhonskiy et al. [135] utilize generative adversarial networks (GANs) to create a more suitable container. However, these methods do not significantly improve the steganography payload capacity compared to traditional methods. Contrarily, Liu et al. [87] adopt a different strategy, replacing the embedding process with synthesizing a container image based on secret information using a GAN.

In the family of modification probability map generation, most methods focus on generating various cost functions that achieve minimal distortion embedding [112]. For instance, Tang et al. [131] introduce a GAN-based distortion learning framework for steganography, while Yang et al. [149] use a generator with U-Net architecture to convert an input image into a container image. Rahim et al. [100], Zhang et al. [158], and Weng et al. [143] use separate embedding and retrieval networks. Zhang et al. [155] propose a novel universal deep hiding (UDH) meta-architecture, which improves the generalization and interpretability of image hiding by disentangling the encoding of the secret image from the cover image.

In the family of adversarial embedding, Tang et al. [130] propose an adversarial scheme under the distortion minimization framework [112]. By employing an additional discriminative network, Hayes et al. [51] further improve the imperceptibility of hidden content. Luo et al. [91] enhance watermark robustness by integrating adversarial training and channel coding in their framework. Using image texture features, Liao et al. [81] formulate adaptive payload distribution to achieve multiple image steganography. SteganoGAN [157] uses the encoder-decoder structure for information embedding and recovery, and a third network plays the role of an adversary to resist steganalysis.

Recently, an excellent technique called the invertible neural network (INN) has been applied to image-hiding tasks. This network uses one normalizing flow backbone network to learn a bijective mapping between the input and target domains to implement both forward and backpropagation operations. Jing et al. [67] and Lu et al. [90] use this revertible network, serving as both embedding and retrieval networks, achieving high fidelity in both processes. Although these promising methods have achieved outstanding fidelity and capacity, their robustness is insufficient. To address this, RIIS [147] introduces a conditional normalizing flow to model the distribution of the redundant high-frequency component conditioned on the container image, thus improving robustness while maintaining imperceptibility and capacity.

CHAPTER 3

Sensitive information sanitization

3.1 Preface

This chapter engages with the critical issue of sensitive information sanitization in image data, a cornerstone of privacy protection in our increasingly digital age. As we produce a tremendous amount of visual data online daily, courtesy of the explosive growth of various computer vision technologies, we inadvertently reveal a wide range of sensitive information. This widespread data availability poses an unprecedented risk of privacy leakage. The threat is particularly acute in the case of photos containing human faces, which can easily be accessed and misused, leading to serious violations of individual privacy.

Given this landscape, we begin the chapter by exploring the threats of inadequate sanitization methods, mainly traditional anonymization techniques like blurring and Mosaic. These conventional methods have proved weak and ineffective in the face of emerging Deep Learning-based attacks.

Responding to this challenge, we present a novel de-identification approach that harnesses the power of deep learning. The approach is based on a framework comprising two modules: an Encoder and Generator networks. The Encoder transforms a face image into a high-semantic latent vector of codes, which are subsequently de-identified according to the differential privacy criterion. The Generator, leveraging the unconditional Generative Adversarial Network (GAN), then synthesizes high-quality images based on these modified latent codes.

We provide a thorough presentation of this innovative approach, detailing its implementation and discussing its implications for privacy preservation. The chapter concludes with an analysis of extensive experimental results, which indicate that our proposed model can protect image privacy while maintaining the visual realism of processed images. Through this exploration, we hope to contribute to developing robust techniques for sensitive information sanitization and push the boundaries of privacy protection in image data.

3.2 Introduction

With the wide deployment of devices equipped with cameras, our society has witnessed a rapid increase in using and generating visual data. These data are used by people as a new form of daily communication and play a crucial role in developing advanced computer vision technologies, such as face recognition, image detection, etc. However, a great amount of sensitive information, such as human faces and/or plate numbers, are contained in the visual data. Directly sharing and using these images inadvertently pose a serious risk of privacy violation.

Government regulations such as the General Data Protection Regulations (GDPR) has went into effect by the European Union. According to GDPR, every person in the images dataset needs to consent to the use of his/her images. This regulation challenges the conventional way of research in computer vision because obtaining everyone's permission in a large-scale images dataset is nearly impossible. Fortunately, according to GDPR, if the image data does not reveal any specific person's identity information, it will be free to use without any consent. Moreover, most computer vision applications do not rely on images' identity features. For instance, image segmentation and object detection only need to detect, instead of identifying certain people in an image.

Therefore, to achieve a balanced trade-off between privacy protection and practical application, it is necessary to sanitize images' identity information while keeping the processed images real-looking.

However, for face images, anonymizing identity information to satisfy the requirement of GDPR while retaining its utility is a challenging task. Traditional anonymization techniques are mainly based on obfuscation, such as Mosaic or Blur, which are inadequate for removing privacy-sensitive information but substantially alter/destroy the original face [84]. Given a face image, an ideal de-identification method should be able to preserve the appearance features of the original image and just remove its identity characteristics. Consequently, the processed images would still look realistic to human observers and AI-based computer vision tools, such as face detectors, emotion classifiers, but people in those images cannot be identified. To be more specific, we formulate the following criterion to regulate the de-identification methods:

- **Anonymization:** The anonymization techniques should have the ability to remove privacy-sensitive information in the input images and reduce the identification possibility of the processed images by vision methods or human observers;

- **Realistic:** The processed image dataset should keep similar distribution with the original, and each image among the dataset should keep high visual quality;

- **Usability:** The complexity of the Anonymization process should be kept as low as possible;

- **Configurable:** The method should support an adjustable protection mechanism, which offers various levels of Anonymization according to users' requirement.

To satisfy the above-mentioned properties, we propose a novel privacy protection framework enforcing de-identification in latent space. Our network builds upon the unconditional GAN to produce realistic images. Unlike the conventional GAN-based image generation controlled by a random noise vector, we adopt an encoder-decoder architecture to create an operable and high-semantic latent space to implement the anonymization processing step. Besides, an Identity-Level loss function is introduced during the network's training process to regularize the network in latent space so as to provide different de-identification effects from less private to more private. Therefore, the proposed method provides configurable image anonymization.

More specifically, the anonymization process of the proposed method first encodes input image into latent space as latent codes, and then generates a de-identified version of the latent codes according to the privacy requirement. Finally, the Decoder uses the modified latent codes to generate the anonymized image. Different from manipulation in pixel space, the proposed image processing in latent space has the following advantgaes: (1) manipulation in the latent space are more accurate so it can appropriately alter original images' characteristics and features, thus preserving output image's quality and utility; (2) the entire anonymization process is unsupervised, which does not require complicated pre-processing and annotations of face areas; (3) unlike de-identification by directly altering pixels, latent space manipulation can provide rigorous privacy protection because the face information is compressed in the tractable latent vectors.

In summary, the major contributions of our works in this paper are summarized as follows:

- We present a novel face images privacy protection framework that implements de-identification of face images via editing images' identity-related features in the latent space;
- We design a dedicated and adjustable privacy-related loss function to regularize the network's training process;
- We validate that our framework outperforms both traditional protection techniques, such as blur and Mosaic, and the state-of-the-art methods, such as CIAGAN [97] and DeepPrivacy [63], regarding privacy protection as well as visual quality and utility preservation. In addition, we evaluate the impact of the privacy regularization parameter on the performance of our proposed method.

3.3 Image Privacy Embedding Framework

3.3.1 Network Architecture

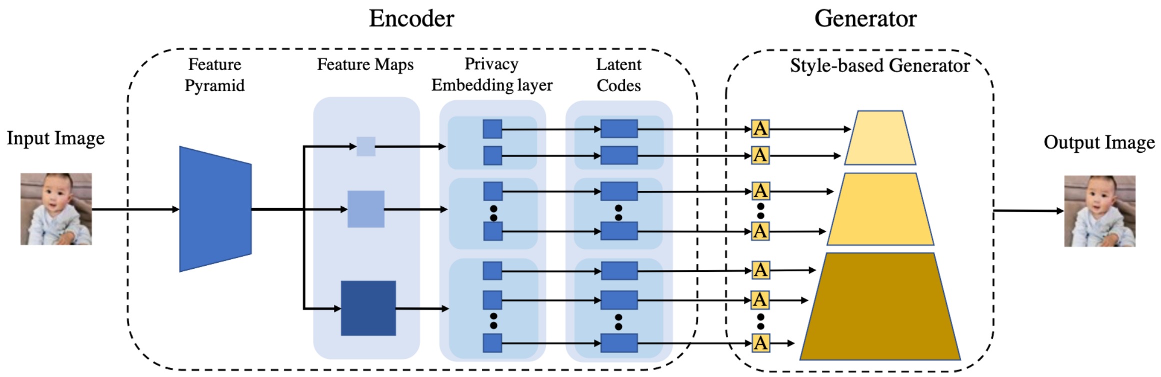


Figure 3.1: The framework of our proposed method. The Encoder consists of a Feature Pyramid network and Privacy Embedding Layers. The Generator utilizes StyleGAN2's generator network. Feature Pyramid network first converts input image into three levels feature maps in latent space. Then, Privacy Embedding layers implement manipulation on feature maps to generate privacy enforced latent code. Finally, Generator employs latent code to synthesize the output image.

The complete architecture of our image privacy embedding framework is illustrated in Fig. 3.1. It adopts an encoder-decoder architecture. We build our model on the one proposed by Richardson *et al.* [119], which aims to reconstruct input images. Nevertheless, the objectives of our works are not only generating images that resemble the original ones, but also limiting the amount of private information revealed in the generated images. Therefore, we perform several alterations. First, the Encoder leverages the Feature Pyramid network to project input images to spatial feature maps in the latent space. Second, it employs Privacy Embedding layers to implement semantic manipulation on feature maps to produce the privacy-anonymization latent codes. Third, the affine transform generates parameters for the fixed and pre-trained Generator network regarding these latent codes to synthesize the de-identification version of input images. The entire image-to-image translation is an end-to-end style that starts from input pixels to latent space feature maps, followed by modified latent codes, then end at output pixels. Hence, different from the state-of-the-art anonymization techniques: CIAGAN and DeepPrivacy, the proposed framework achieves image de-identification in the latent space instead of the pixel space.

3.3.1.1 Encoder

The primary objective of Encoder is to generate latent vectors with respect to the input images and to perform de-identification editing on such vectors. There are two challenges to realize the goal: (1) How to project the image into the latent space accurately; and (2) How to anonymize image in the latent space semantically.

For the first challenge, a simple solution is to directly extract the same dimension vectors with respect to the Generator from the last layer of the Encoder network. However, such an approach presents a substantial bottleneck limiting the reconstruction fidelity and latent space's semantic richness [1, 2]. We attribute this limitation to the absence of original image's spatial information in the latent spaces. This is mainly because low dimension style vectors can not fully reflect the original image's high-level features especially the pixels' relation in images. Without spatial information, the input image's semantics are compressed in an entangled manner, making it difficult for further manipulation and reconstruction. Therefore, our Encoder adopts a Feature Pyramid Network (FPN) as the mapping network to produce latent space with spatial dimensions. FPN projects the input images into three levels of feature maps, representing coarse, medium and fine details of the input image [119]. This property allows Encoder produces high semantic and fidelity latent space, which enable further manipulation and reconstruction.

For the second challenge, we employ a trainable Privacy Embedding Network (PEN) to transform the feature maps into latent codes for future de-identification manipulation. The PEN adopts fully convolutional layers' architecture followed by LeakyReLU activations to best comprehend and interpolate the spatial information of feature maps. Each PEN corresponds to one Latent code vector. The specific layer number of each PEN is aligned with the feature maps' hierarchical scales to guarantee to generate the same dimension latent codes. Feature Pyramid Network and Privacy Embedding Network are jointly trained to protect sensitive information in latent space.

3.3.1.2 Generator

The Generator generates an output image utilizing latent codes extracted by Encoder. Motivated by the state-of-the-art visual synthesis quality and high semantic latent space, we employ a pre-trained StyleGAN2's generator network as our Generator. StyleGAN2 is equipped with re-designed generator architecture, which provides disentangled latent space $\mathbb{W}$ and editing capabilities to synthesize images. To better utilize the representative power of StyleGAN2, followed by common practice [119], we use the extended latent space $\mathbb{W}+$, which composed of the concatenation of 18 vectors , each with a dimension of 512 for each input layer of StyleGAN2, to control image generation.

Consequently, the latent codes, aligned with the hierarchical representation, are fed to Generator through an affine transform to generate the output image. The complete data translation of our framework is an end-to-end image-to-image translation. More specifically, we denote the Encoder's latent space encoding and manipulation process as $\mathbb{F} : \mathbb{R}^{\times} \rightarrow \mathbb{R}^{\mathbb{1}\times\mathbb{1}}$, where the input image x maps to a 18×512-dimension codes. The Generator's reconstruction transform is denoted as $\mathbb{F}^{-\mathbb{1}} : \mathbb{R}^{\mathbb{1}\times\mathbb{1}} \rightarrow \mathbb{R}^{\times}$.

3.3.2 Training and Losses

The left part of Fig. 3.2 illustrates the training scheme of our framework. We use **E** and **G** to denote our Encoder and Generator. Since the Generator network is built upon the representative power of pre-trained StyleGAN2's generator [71]; therefore, only the Encoder is updated during the training to achieve image anonymization. Besides, the entire training scheme does not require any pre-annotations. Encoder implements all image manipulation operations on images' latent space instead of Generator on the pixel

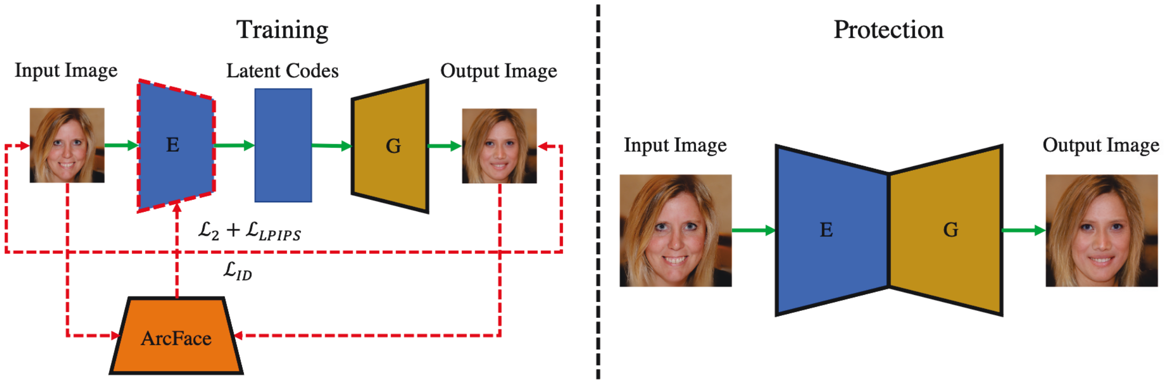


Figure 3.2: The training and protection scheme of our framework. Green arrows refer to data flows from the input image to the generated image. Dashed red lines indicate loss functions. Besides, the trapezoid with a red dash outline indicates a trainable network, while black full line trapezoids represent fixed and pre-trained networks.

level. To semantically guide the training, we utilize a weighted combined loss function, which consists of three dedicated sub-loss functions for different objectives:

Pixel-Level Loss: $\mathscr{L}_2$ loss is adopted to enforce the reconstructed images $\hat{x} = G(E(x))$ to pixel-wise resemble input images x,

$$\mathscr{L}_2(x) = \|x - \hat{x}\|_2,$$

where $E(\cdot)$ denotes Encoder network, $G(\cdot)$ denotes Generator network, x and $\hat{x}$ are original and corresponded processed image.

Perceptual-Level Loss: In addition to preserving perceptual quality, we leverage the Learned Perceptual Image Patch Similarity (LPIPS) [159] loss to encourage the reconstructed images perceptually similar with the originals,

$$\mathscr{L}_{\mathbf{LPIPS}}(x) = \|L(x) - L(\hat{x})\|_2,$$

where $L(\cdot)$ represents the perceptual features extractor.

Identity-Level Loss: To limit the amount of private information presented in the reconstructed images, we regularize the cosine similarity between the input and reconstructed images' identity feature vectors. Specifically, by employing the pre-trained ArcFace network [31], we obtain the identity features vector of images. Then, we set up a privacy regularizer $\beta \in [0,1]$ to restrict the similarity between input and reconstructed images' identity features to reduce the privacy information exposed in the reconstructed images. Formally, the identity loss function is written by:

$$\mathscr{L}_{ID}(x) = \left|\beta - CosineSimilarity(Arc(x), Arc(\hat{x}))\right|,$$

where $Cos(\cdot)$ denotes cosine similarity and $Arc(\cdot)$ denotes pre-trained ArcFace network. Besides, accompany with the privacy regularizer β's, the identity-level loss will impose a different level of privacy protection effects. As $Cos(Arc(x) - Arc(\hat{x})) = 1$ indicates highest similarity between x and $\hat{x}$, a smaller β will enforce larger distance in identities and therefore better privacy protection.

The overall weighted sum loss function is defined as:

$$\mathscr{L}(x) = \lambda_1 \mathscr{L}_2(x) + \lambda_2 \mathscr{L}_{LPIPS}(x) + \lambda_3 \mathscr{L}_{ID}(x),$$

where $\lambda_1, \lambda_2, \lambda_3$ are constant weighting corresponded loss.

3.3.3 Protection Stage

The right-hand-side part of Fig. 2 illustrates the protection scheme of our framework. With our model trained to minimize loss function Eq. (4), the network enables de-identification of the input images. During this stage, both Encoder and Generator are fixed. Therefore, the input image is encoded into latent space, and then processed by the proposed privacy-enhancement mechanism, resulting in an output of a privacy-preserving latent code. The Generator will then synthesize a de-identification image according to the privacy-preserving latent code. The Latent Codes part is omitted in Fig. 2 for brevity.

3.3.4 Attack Model

We consider a robust threat model to validate our framework's privacy protection capability in a worst-case scenario. The adversary's objective is to learn personal identity by accessing images and then using the extracted identity information to match other people's images illegally. For example, an adversary can utilize the face on Google street view to search corresponding individual social network accounts or other personal images published on the Internet to further illegally surveil people. We assume that the adversary can acquire all processed images shared in online social networks but have no access to the original images (which represent corresponding personal images without processed by privacy-enhancement methods). Besides, the adversary is capable of utilizing state-of-the-art face recognition methods to launch identification attacks.

To quantify this risk, we calculate the **Identity Similarity** between the original and processed images,

$$Id\_Similarity = CosineSimilarity(F(x), F(\hat{x})),$$

where $F(\cdot)$ represents identity features extractor which based on pre-trained facial recognition networks.

Specifically, the higher Identity Similarity between the original and processed images indicates a higher possibility of success illegal identification by the adversary, and hence a lower privacy-level, and vice versa. Therefore, the objective of image privacy protection techniques is to reduce their output image's Identity Similarity compared with that of the input one. Given by the dedicated Identity-Level loss, our framework provides adjustable control over the processed images' Identity Similarity with the original images. Hence, our framework could effectively defence the re-identify attack, despite the state-of-the-art facial recognition model.

3.4 Experiment

In this section, we implement extensive and comprehensive experiments to evaluate our framework's effectiveness of identity anonymization. The proposed method is compared with both classic and state-of-the-art anonymization methods on various faces image datasets. The experiment results indicate that the proposed method acquires the best performance regarding various qualitative and quantitative evaluation metrics. Besides, we also present a set of comparisons to reflect how privacy regularizer β affects the anonymization performance of our method. The datasets, baselines and evaluation metrics will introduce in the following:

Datasets. The experiments are conducted on two public well-known faces image datasets to exhibit the performance of the proposed image de-identification framework.

- **CelebA [88]:** the dataset consists of 202599 face images with various features, such as age, gender and race. For a fair comparison, we use the aligned version where each image centred on a point in-between person's eyes and then resized to 256×256 resolution. Only 20k images are randomly selected to train the proposed model for saving time.

- **Flickr-Faces-HQ [71]:** This dataset is composed of 70k high-quality PNG images with 1024×1024 resolution and also provides considerable coverage in terms of personal age, ethnicity, image background, accessories, etc. To reduce training complexity and time, we randomly selected 10k image from this dataset. Every selected image are aligned and cropped to the cetral point, then resized to a resolution of 256×256.

Comparative studies. We compare two classic anonymization methods and two state-of-the-art learning-based techniques.

- **Classic Methods:** We use Mosaic and Blur to compare them with our method. Both of them are current mainstream and most commonly used image privacy-enhanced techniques which well represent the traditional methods.
- **Learning-Based Method:** We select DeepPrivacy [63] and CIAGAN [97] as benchmark schemes. We adopt the official codes and pre-trained models given by the authors. These two methods are selected because they satisfy our proposed de-identification criterion and achieved better performance compared with the other existing learning-based methods.

3.4.1 Evaluation Metrics

To quantitatively evaluate and compare the interested schemes, we employ the following metrics to assess their performance in the aspects of visual quality, privacy protection and utility.

3.4.1.1 Visual Quality Metrics

Three different evaluation metrics are employed to measure the visual quality of the de-identification images:

- **MSE:** This metric calculates the pixel-wise Mean Square Error (MSE) between the input and anonymous images to compare different outputs visual quality at the pixel-level. A lower MSE value indicates a higher similarity between the original and the de-identification images, implying better visual quality preservation.
- **SSIM [140]:** Rather than directly comparing the images pixel by pixel, we use Structural Similarity (SSIM) to measure the perceptual difference between the input and processed images incorporating Luminance, Contrast and Structure. Therefore, a lower SSIM indicates better images' visual quality preservation from the human perceptual perspective.
- **FID [55]:** Different from the previous metrics which measure pair-wise image similarity, Frechet Inception Distance (FID) calculates the Frechet distance between the input and processed image datasets' multidimensional Gaussian distributions

$\mathscr{N}(\mu, \Sigma)$ using the Inception v3 [127] features to quantify their quality similarity. A lower FID represents better quality preservation.

3.4.1.2 Privacy Metrics

The objective of the privacy metrics is to evaluate the performance of privacy protection. There are two different privacy metrics used in our experiments.

- **Identity Similarity:** According to Eq. (5), we define **Identity Similarity** as below. This metric calculates the cosine similarity of the original and anonymized images' identity feature vectors to quantify the effectiveness of privacy protection. As the ArcFace model is employed in our encoder's loss function, we leverage another state-of-the-art Facial Recognition network's pre-trained model, CurricularFace [62], to extract the images' identity features.

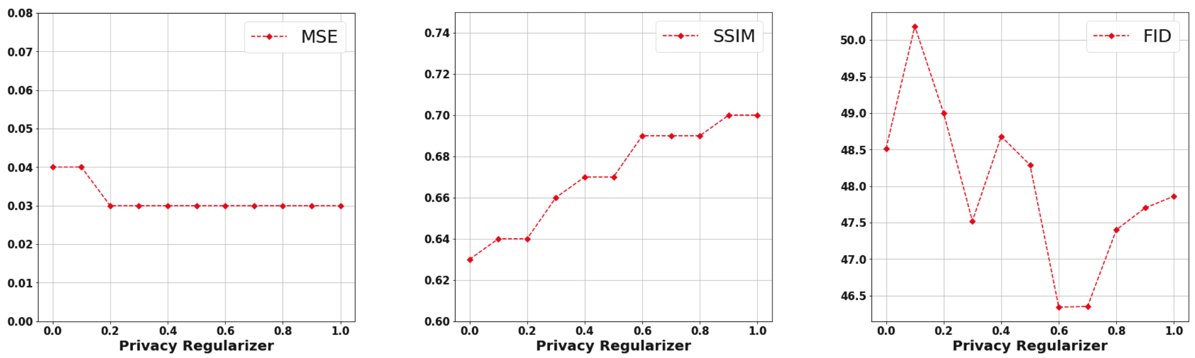
  $$Id\_Similarity = CosineSimilarity(CF(x), CF(\hat{x})),$$

  where $CF(\cdot)$ denotes pre-trained CurricularFace-based identity feature extractor. A lower Identity Similarity value between the original and processed images indicates a higher level of de-identification. Averaging the Identity Similarity among 10k random identities from the FFHQ dataset, we obtain an empirical threshold δ value: 0.19. Hence, in the following experiments, an image pair with an Identity Similarity lower than 0.19 will be regarded as different identities.

- **De-Identity Rate:** Besides the **Id_Similarity**, we present another evaluation metric**: De-Identify Rate** = $\hat{y}/y$, where $\hat{y}$ is the number of image pairs that can be recognized by the pre-trained CurricularFace as different identities, and y is the total number of images pairs in the experiment. This metric is used to measure the ratio of the anonymized images that have completely removed the original identity characteristics.

3.4.1.3 Utility Metric

The processed images using the anonymization methods should maintain a high utility in practical identity-agnostic computer vision tasks, such as face detection. To quantitatively compare the studied methods in terms of utility preservation, we perform face detection using the standard Dlib-ml library‚Äôs HOG-based face detector [27] on their processed images. We measure the percentage of detected faces to evaluate the performance of each anonymization method, with 1.0 representing perfect utility preservation.



(a) MSE (b) SSIM (c) FID

Figure 3.3: The utility metrics corresponding to β from 0 to 1.

3.4.2 Impact of Privacy Regularizer

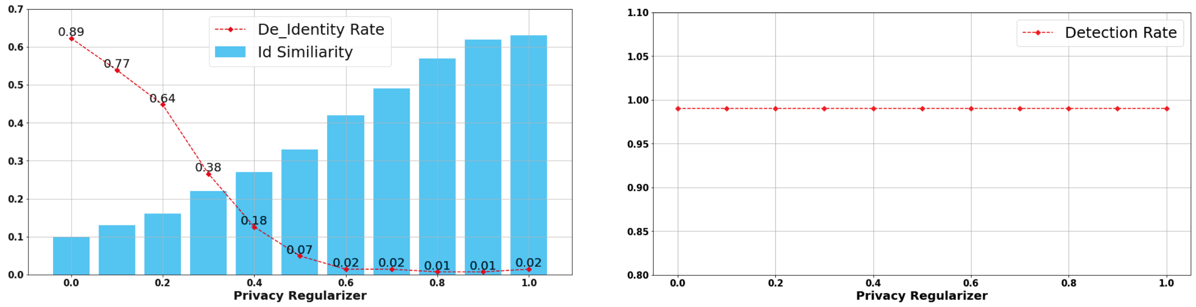
We now discuss the impact of the privacy regularizer β on the visual quality, utility and privacy. Recall that β is incorporated in the identity loss function to regulate the input and processed images' identity similarity. A lower β leads to a higher variation, and hence more potent privacy protection on the processed images, and vice versa. We train our framework by varying β from 0.0 to 1.0 with an interval of 0.1 to construct different models. Then, we calculate the defined metrics over different models to evaluate the impact of β on the performance.

3.4.2.1 Visual Quality Evaluation

First, we show the experiment results of visual quality. As illustrated in Fig. 3.3, the trend of the quality metrics is consistent with each other. The processed images' quality increases with the decrease of the required privacy protection level. This phenomenon indicates that altering the image's identity features will also reduce the quality of the reconstructed images. However, according to quantitative results, the quality reduction is not obvious, which verifies that our method can generate sufficiently high-quality images while providing privacy protection.

3.4.2.2 Privacy Protection Evaluation

The quantitative results of privacy protection are shown in Fig. 3.4a. With the relaxation of privacy regularizer, the average of Id_Similarity continues to rise, and the De_Identify Rate declines, which shows that a smaller privacy regularizer provides a higher privacy protection level, and vice versa. Moreover, there is an "elbow" point appearing at around



(a) Id_Similarity and De_Identify rate.

(b) Detection Rate.

Figure 3.4: Average Id_Similarity, De_Identify, and Detection Rate along with privacy regularizer β increase from 0 to 1.

$\beta = 0.2$ on the De_Identify Rate curve, where the privacy protection level on the processed images starts drops rapidly. Besides, the privacy protection becomes negligible at $\beta = 0.4$ when facing the re-identification attacks using the state-of-the-art facial recognition models.

3.4.2.3 Utility Performance Evaluation

We show in Fig. 3.4b the results of utility with respect to the range of privacy regularizer. The detection rates remain at almost 0.99 with various privacy regularizers, demonstrating that our method achieves a nearly perfect score in preserving utility. Besides, it also proves that some computer vision tasks, such as face detection, are identity-agnostic, which do not rely on people's identity information. Therefore, the proposed anonymization techniques could be employed to protect the privacy in the publicly available large-scale face image datasets while preserving their utility in computer vision tasks.

3.4.2.4 Qualitative Comparison:

Furthermore, we visualize several samples with different β in Fig. 3.5 to conduct a qualitative comparison. As β decreases, the visual identity of the processed image significantly changes comparing with that of the original one, while most of the non-identity features are retained to generate a high fidelity for the processed images.

3.4.3 Comparison with Classic Methods

In this subsection, We present comparison experiments between our method and the mainstream image anonymization techniques, i.e., blurring and mosaic. For the sake

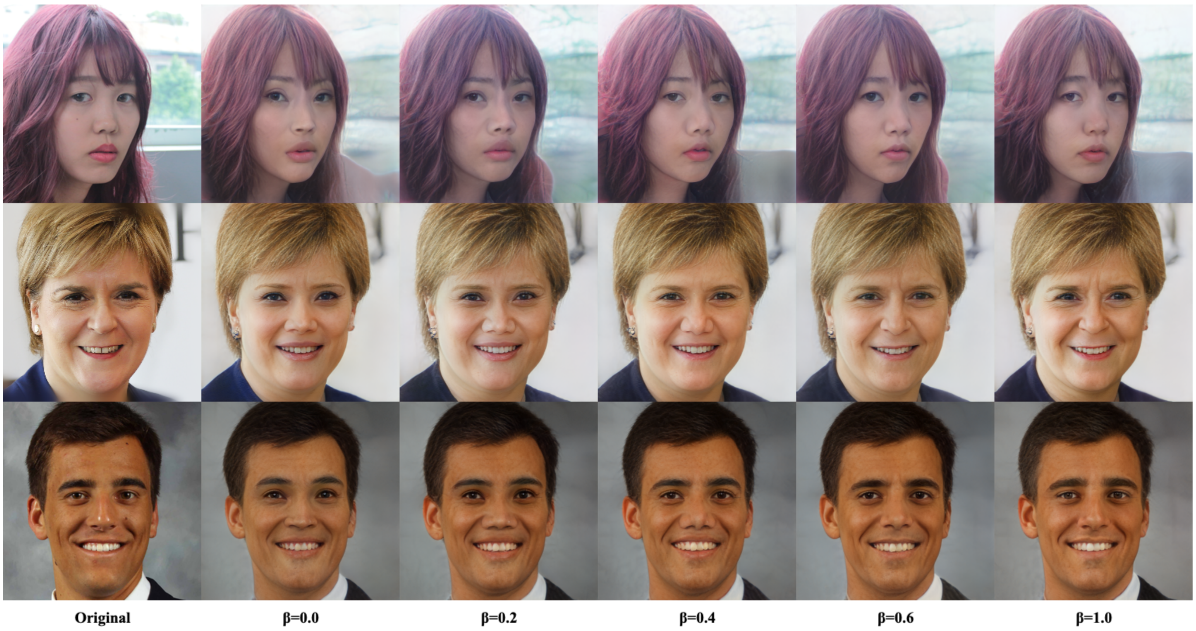


Figure 3.5: Qualitative comparison of different privacy regularizer β. With a lower regularization parameter, the processed image's identity similarity significantly different from the original. Besides, corresponding to discover in De-identify Rate curve, we note that images output by models whose β higher than 0.4 still reserve very similar visual identity with the original. Only images processed by models whose β lower than 0.2 have relatively large difference with the original.

of fairness, all methods will be calibrated to reach a comparable value in terms of a performance metric value, and then we will apply the other performance metrics to evaluate their performance. The experiments in this section are conducted using the **FFHQ** dataset.

3.4.3.1 Visual Quality Evaluation

We first evaluate the visual quality of the anonymized images. Our model and two benchmark methods are fine-tuned to make their privacy metric values reach the following numerical range [0.1, 0.2, 0.4], which represents a variety of privacy protection levels in the order of strength. Then, we evaluate the aforementioned visual quality metrics. The results are summarized in Tables 3.1, 3.2 and 3.3. As shown in these tables, our framework outperforms the blur and mosaic techniques in every category of performance metrics at all of the investigated privacy protection levels. These results show that for a given privacy protection level, our method can generate a higher utility compared with the conventional techqniues.

Table 3.1: Quality metrics at identity similarity around 0.1.

	MSE↓	SSIM↑	FID↓
Blur	0.12	0.50	158.23
Mosaic	0.42	0.40	130.71
Ours	**0.04**	**0.63**	**48.51**

Table 3.2: Quality metrics at identity similarity around 0.2.

	MSE↓	SSIM↑	FID↓
Blur	0.05	0.60	98.65
Mosaic	0.10+	0.46	119.99
Ours	**0.03**	**0.66**	**47.52**

Table 3.3: Quality metrics at identity similarity around 0.4.

	MSE↓	SSIM↑	FID↓
Blur	**0.03**	0.65	65.82
Mosaic	0.05	0.54	108.03
Ours	**0.03**	**0.69**	**46.33**

3.4.3.2 Privacy Protection Evaluation

In this subsection, we evaluate the privacy protection performance of our method. Similar to the evaluation of visual quality, we calibrate the interested methods to achieve a similar SSIM value for fair comparison. From the experimental results, we find that our framework can achieve a relatively stable SSIM value at around 0.65, with different sets of parameters (more details will be discussed in the latter part of this section). In the following, we only evaluate the privacy protection level under an SSIM of 0.65.

Table 3.4: Identity Similarity at same SSIM value (0.65).

Methods	**Identity Similarity↓**
Blur	0.38+-0.11
Mosaic	0.87+-0.04
Ours	**0.16+-0.08**

Table 3.5: Detection Rate at same SSIM value (0.65).

Methods	**Detection Rate↑**
Blur	0.4575
Mosaic	0.9818
Ours	**0.9999**

Table 3.4 shows that our method can significantly reduce the identity similarity between the input and processed images comparing with the benchmark techniques, which indicates that our method can provide a higher privacy protection level.

3.4.3.3 Utility Performance Evaluation

Next, we present the evaluation results of utility. We calculate the detection rate of each method when the Identity Similarity reaches [0.1, 0.2, 0.4] and the SSIM is around 0.65, respectively. The results are reported in Fig.3.6 and Table 3.5.

As shown in Fig. 3.6, our anonymized images consistently achieve 100% detection rates under various privacy metric values. This result indicates that the proposed method could perfectly maintain image utility in the face detection task. On the contrary, the mosaiced and blurred images have much lower detection rates, indicating that these anonymization techniques incur heavy utility loss in detection tasks. Table 3.5 shows

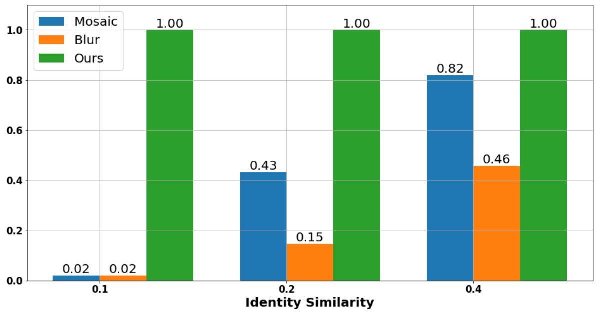


Figure 3.6: Detection Rate at Identity Similarity on [0.1, 0.2, 0.4].

that the mosaic and blur techniques will inevitably cause utility loss even under the same visual quality, while our method shows perfect utility preservation.

3.4.3.4 Qualitative Comparison

Apart from the above quantitative comparison, we also illustrate several original and processed images in Fig. 3.7 with SSIM=0.65 to qualitatively exhibit the privacy protection. Regardless of the relatively high SSIM value, the blur and mosaic technique lead to noticeable perturbation on images, which will significantly compromise their applications in practice. In contrast, our method semantically modifies the ROIs of the images, while maintain the fidelity in the processed images.

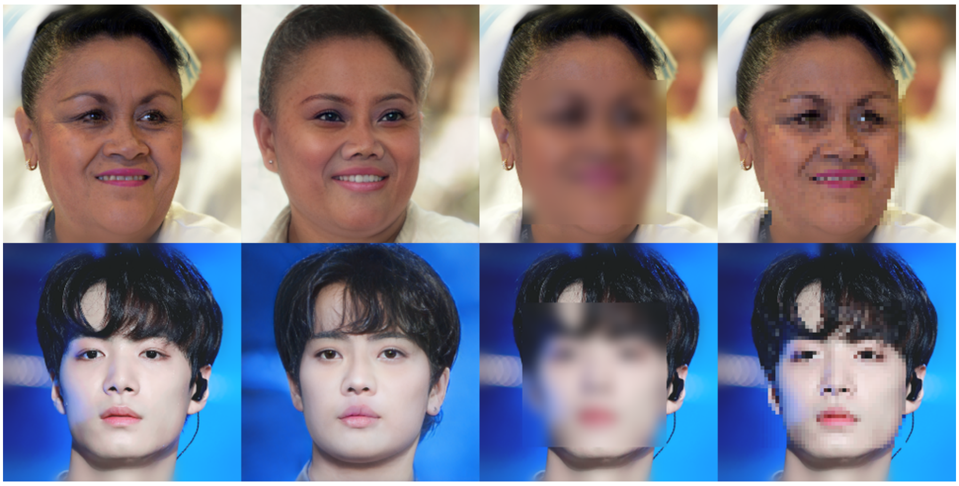


Figure 3.7: Qualitative comparison between our method and classic anonymization techniques under SSIM values: 0.65, from left to right: **Original**, **Ours**, **Blur** and **Mosaic**.

.

3.4.4 Comparison with the state-of-the-art Methods

This section compares our method with the state-of-the-art de-identification methods, i.e., DeepPrivacy [63] and CIAGAN [97], which are both trained and tested on the CelebA dataset. Thus, we also apply our framework to the CelebA dataset for fairness comparison.

Both the reference works cannot adjust the privacy protection levels. Hence, we calculate their outputs' Identity Similarity to be 0.1 and 0.2, respectively. Then we fine-tune our model to achieve the same Identity Similarity and conduct a comparative experiments. Table 3.6 lists the quantitative comparison results. In terms of the visual quality metrics, CIAGAN achieves an impressive performance on FID by obtaining a score of 12.72. Our method is slightly inferior to CIAGAN by achieving an FID score of 31.11. Although FID is usually employed as an important metric to evaluate the output quality of GANs, it is calculated based on the distribution of generated images, which cannot fully capture the quality of a single image. Besides, our method outperforms CIAGAN and DeepPrivacy in the other Visual Quality metrics of MSE and SSIM. It shows that our network could generate anonymous images with comparable visual quality.

Fig. 3.8 illustrates more perceptual comparison results. From this figure, we can see that our model produces more visually-realistic anonymous faces that preserve more characteristics of the original identity. In contrast, the process images from CIAGAN look different to the source images, because of the direct change of the original ID. However, when the fake Identification does not share the same gender, age or makeup, CIAGAN tends to produce extremely unrealistic images (e.g., row 3, column 5 in Fig. 3.8). Besides, distortions and artifacts often occur on their processed images. The processed images from DeepPrivacy could relatively well keep the facial pose and outline, nevertheless it adds fuzziness on the face area. In addition, both CIAGAN and DeepPrivacy share another significant flaw, i.e., these two techniques rely on facial landmark detection to provide pre-annotation and require to feed their networks with face-removing images, making it difficult to deploy them in real-world applications. On the contrary, our approach does not have these issues and can provide adjustable privacy protection.

3.4.5 Discussions

In summary, the experimental results demonstrate that our method could provide adjustable privacy protection, while generating sufficiently high-quality images. This

Table 3.6: Quality metrics at Identity Similarity around 0.2.

	MSE↓	SSIM↑	FID↓	DR↑
CIAGAN	0.07+-0.02	0.65+-0.07	**12.72**	0.9939
Ours(Id=0.2)	**0.02+-0.00**	**0.73+-0.08**	31.11	0.9989
DeepPrivacy	0.09+-0.04	0.61+-0.09	25.94	0.9976
Ours(Id=0.1)	**0.02+-0.00**	0.72+-0.08	33.41	0.9976

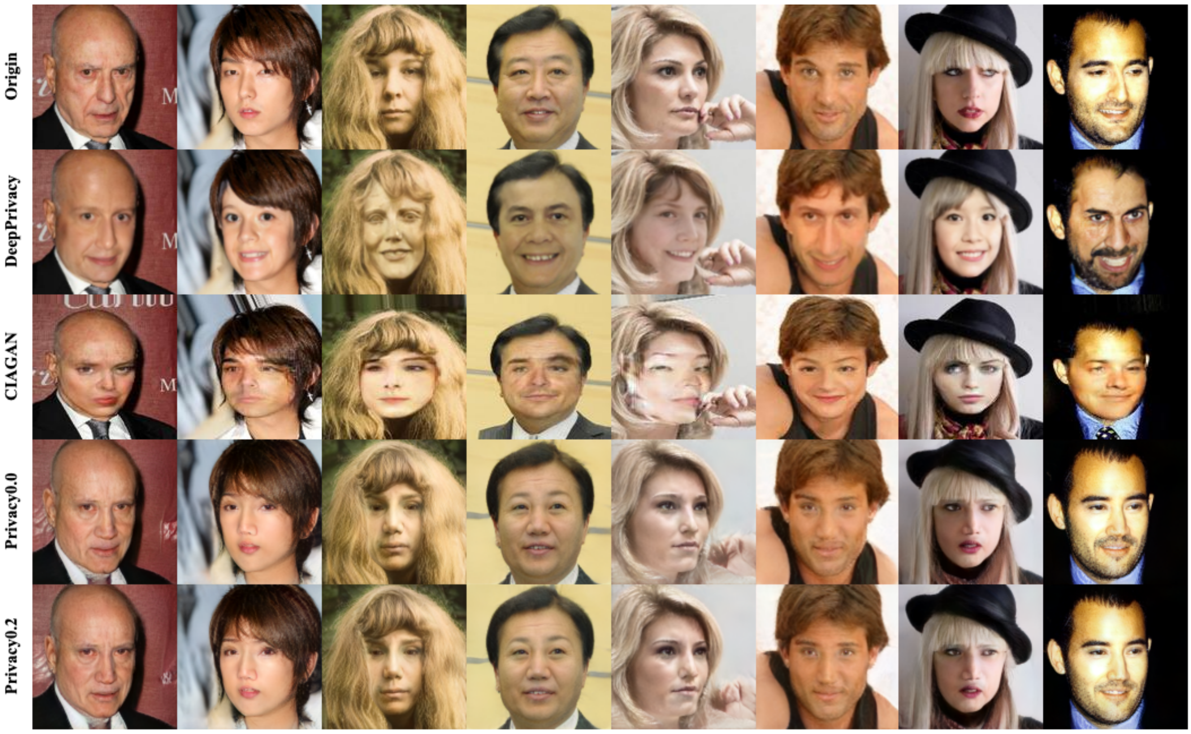


Figure 3.8: Qualitative comparison between our method and SOTA anonymization techniques DeepPrivacy and CIAGAN. From top to bottom show the outputs of: **Original**, **DeepPrivacy**, **CIAGAN**, **Ours** ($\beta = 0.0$), and **Ours** ($\beta = 0.2$).

makes our method capable of satisfying different application requirements in practice. From the presented results, it is obvious that our approach significantly outperforms the classic obfuscated-based methods in anonymization task to achieve a balanced trade-off among visual quality, privacy protection and utility preservation. Compared with the deep learning based methods, i.e., CIAGAN and DeepPrivacy, our method can provide more semantic and accurate anonymization. The qualitative results show that the generated images from the proposed method can retain more original characteristics. In contrast, both CIAGAN and DeepPrivacy fail to preserve enough original features.

However, according to our extensive experiments, we find several weaknesses of the current deep learning based de-identification methods. First, these methods rely on face detection. Any face that is not detected by the deep learning based methods cannot be anonymized. Our method suffers from a similar issue as it depends on the pre-trained StyleGAN. Thus, it is challenging to anonymize face images that are not facing forward because such examples are not available during the StyleGAN training process. In addition, faces covered with objects, such as earrings, are extremely hard to process. To generate a face with such features requires a more careful design, which is still an open problem for the deep learning based methods.

3.5 Conclusion

This chapter presents a novel image privacy protection framework that could protect the image's privacy in the latent space and achieve a balanced trade-off between the image's privacy, utility and quality. The proposed framework consists of an Encoder and a Generator. Input images are translated by Encoder into the latent space and then subject to semantic manipulations to protect privacy of faces. Using the Encoder's output, the Generator is built upon a pre-trained unconditional GAN to reconstruct a high-fidelity and anonymous image. The advantages of our framework are two-fold: i) it can remove the identity information in the target image while retaining the other information that has nothing to do with identity (such as image structures), thereby providing a visually realistic image, and ii) the degree of de-identification can be controlled via a parameter to provide adjustable protection so that users can flexibly tune their requirements of privacy and utility. Our experimental results demonstrate the effectiveness of our framework in real-world image datasets, thanks to its ability to generate comparable performance metrics with the classic techniques as well as the state-of-the-art methods. In the future, we will further explore the disentanglement of sensitive and non-sensitive attributes in images as well as videos.

As Chapter 3 concludes with innovative approaches to sensitive information sanitization in image data, the next chapter will seamlessly transition into the realm of forgery media detection. Building upon the foundational understanding of protecting sensitive data, Chapter 4 will introduce the challenges and advancements in detecting and preventing digital forgeries, especially in the rapidly evolving landscape of Deepfake technology. The focus will shift from safeguarding information to ensuring media authenticity in an era where digital manipulation is becoming increasingly sophisticated. Specifically, in the next chapter, we design a proactive framework to combat malicious forgery by watermarking face identity features to counter the increasing prevalence of visual data manipulation or forgery. This novel framework embeds watermarks in face identity features, differing from traditional artifact-based detection methods. It includes a new method for Deepfake detection, a simple yet effective encoder-decoder network, and extensive evaluations demonstrating its effectiveness, robustness, utility, and security.

CHAPTER 4

FORGERY MEDIA DETECTION

4.1 Preface

Chapter 4 delves into the increasingly critical issue of malicious forgery and tampering in image data, a concern heightened by the explosive progress of Deepfake techniques. These emerging techniques, capable of creating realistic yet false visual content, pose unprecedented privacy and security risks to our society.

Traditional methods for forgery detection often stumble upon limitations, as they primarily rely on capturing artifacts left by a Deepfake synthesis process. However, these clues can be easily removed through various distortions (e.g., blurring) or even more advanced Deepfake techniques, making the detection process challenging. We outline these limitations before introducing an innovative solution to this rapidly evolving problem.

Our proposed framework hinges on an anti-counterfeit labelling mechanism to protect face images from malicious Deepfake tampering, moving beyond merely identifying the artifacts. Using a neural network with an encoder-decoder structure, we embed watermarks as anti-Deepfake labels into the facial identity features. These injected labels are entangled with the facial identity features, which makes them sensitive to face swap translations (i.e., Deepfake) but robust against conventional image modifications (e.g., resize and compress).

In the following comprehensive discussion, we present the process of embedding watermarks as anti-Deepfake labels into facial identity features. We also extensively analyse our method's performance, drawn from extensive experimental results. Our experiments indicate that our method can achieve an average detection accuracy of over 80%. Through this chapter, we aim to contribute a significant step forward in the ongoing fight against the rampant spread of Deepfake and its resulting threats.

4.2 Introduction

The advancement of deep generative approaches has led to various powerful Deepfake methods, which can synthesize visually authentic images/videos. However, abusing Deepfake techniques poses a pressing threat to the integrity of multimedia information and personal privacy, such as fake news or rumours. To counterbalance the aggressiveness of Deepfake, a new research branch known as Deepfake Detection arises, which aims to utilize traditional media forensics methods or deep learning technology to differentiate the fake images/videos from the real ones.

Existing Deepfake detection approaches mainly focus on passively capturing the artifacts introduced during the Deepfake synthesis as clues to identify the fake images/videos, which suffer from two fundamental issues: **(1) Generalization: artifact-based detection methods are difficult to generalize to unknown scenarios.** These methods depend highly on the artifacts learned during the training process, so they exhibit poor performance in dealing with unknown and strange artifacts [93]. Besides, Deepfake techniques are developed with an alarming speed, leaving fewer detectable artifacts in their synthesized results [23, 72]. These methods are thus struggling to keep up with the development of the Deepfake techniques. **(2) Robustness: artifact-based detection methods are not robust against real-world distortions.** Conventional image manipulations (e.g. cropping, compression) might destroy the artifacts in Deepfake results. These effects would further make the artifact-based detection methods less reliable in such scenarios [61, 115]. Besides, the carefully crafted imperceptible adversarial noise in Deepfake images/videos can also significantly reduce the effectiveness of the artifact-based detection [19, 46].

To overcome the above problems, we propose a novel framework to proactively watermarks the identity feature of face images and then determine whether these images are Deepfake or not according to the existence of the watermark. The mechanism of our method is similar to anti-counterfeit. Before sharing personal images online, the user can use our method to embed his/her watermark into these images. The watermark acts as the anti-Deepfake label to protect the user's authenticity of these images. Once images with similar identities to the watermarked images appear online, the owner of watermarked images can verify these suspect images' authenticity according to the existence of his/her watermark. More details about the proposed method's real-world application scenario are in Section 4.4.

As shown in Fig. 4.1, our proposed framework consists of two major steps: watermark injection and watermark verification. In the watermark injection step, the input face image is first disentangled via two dedicated networks into an identity representation and multi-level attributes representation. Then we embed a pseudo-random sequence into the identity representation to generate the watermarked identity. The embedded sequence has no dependence on the face image, so it can be randomly selected but will be preserved and used for the watermark verification step. Another generative network integrates the watermarked identity with the original attributes to synthesize the watermarked image. This watermarked image is perceptually similar to the original one, excluding the negative impact on the image's normal use from the watermark.

The watermark verification step aims to verify the existence of the watermark in the image to determine whether Deepfake has manipulated it. We employ the same network used in watermark injection to extract the image's identity representation and then calculate its correlation with the preserved watermark to inspect whether the watermark exists. According to the feature of the pseudo-random sequence, if a peak appears in the correlation result, it indicates the watermark is still in the corresponding image, so it has not tampered with Deepfake. Otherwise, it will be determined as a fake one. More details about the proposed method will be explained in the following sections.

In summary, our main contributions are summarized as follows: (1) We propose a novel proactive Deepfake detection method by embedding an anti-counterfeiting watermark into images' identity vectors. (2) We design a simple but effective encoder-decoder network to implement invisible anti-Deepfake watermarking, which requires neither pre-annotation nor pre-detection information. (3) We conduct extensive experiments to evaluate the performance of our method in terms of effectiveness, robustness, utility and security.

4.3 Preliminary

Pseudorandom Noise (PN) sequence is widely used in signal processing, which is usually a binary sequence with a spectrum similar to a random sequence but generated by a de-

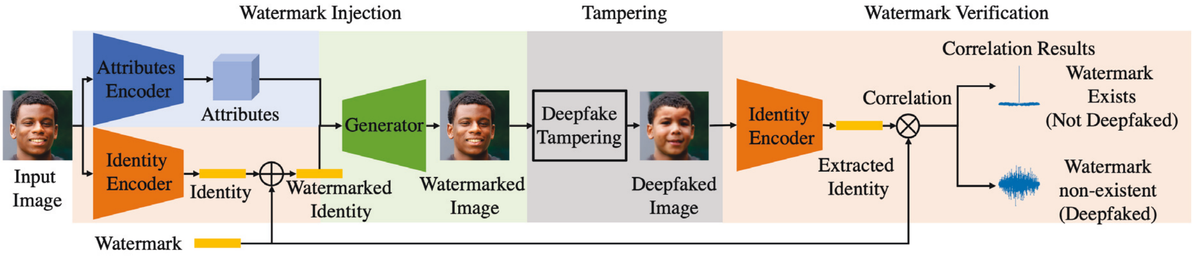


Figure 4.1: Our method's overall framework consists of watermark injection and watermark verification. The watermark injection step aims to generate watermarked images that are perceptually similar to the original and contain the anti-Deepfake watermark. The watermark verification step aims to verify the existence of the watermark in the image's identity representation to determine whether it is counterfeited or not. The tampering part in the middle represents potential Deepfake manipulations on our watermarked image, which is not our framework's component but is the objective we aim to detect.

terministic algorithm. Linear feedback shifter register (LFSR) is one of the simplest ways to generate a PN sequence. In an LFSR, any bit is determined by a linear combination of the previous n bits, which can be formulated as:

$$B_n = A_0B_0 \oplus A_1B_1 \oplus A_2B_2 \oplus ... \oplus A_{n-1}B_{n-1}$$

Since any bit is a function of the previous n bits, every LFSR can produce a sequence of bits that appears random and has a period of $2^n - 1$. There are several commonly used PN sequences which are: maximum length sequence (MLS), Gold sequences, Kasami sequences and JPL sequences.

MLS sequence is the most representative PN sequence which is generated using maximal LFSR. They are periodic and reproduce every binary sequence (except the zero vector) that can be represented by the shift registers, i.e., for m length registers they produce a sequence of with length $2m - 1$. The auto-correlation of an MLS is approached to unit impulse function as MLS length increases. This property makes the MLS suitable for synchronization and in the detection of information in single-user Direct Sequence Spread Spectrum systems.

The Gold sequence is another well-known PN sequence that belongs to the category of product codes for they are produced by XOR two same length MLS. The two MLS must maintain the same phase relationship till all the additions are performed. A slight change of phase even in one of the MLS produces a different Gold sequence altogether. Gold codes are non-maximal and therefore they have poor auto-correlation property when compared to MLS. However, it possesses good cross-correlation properties which

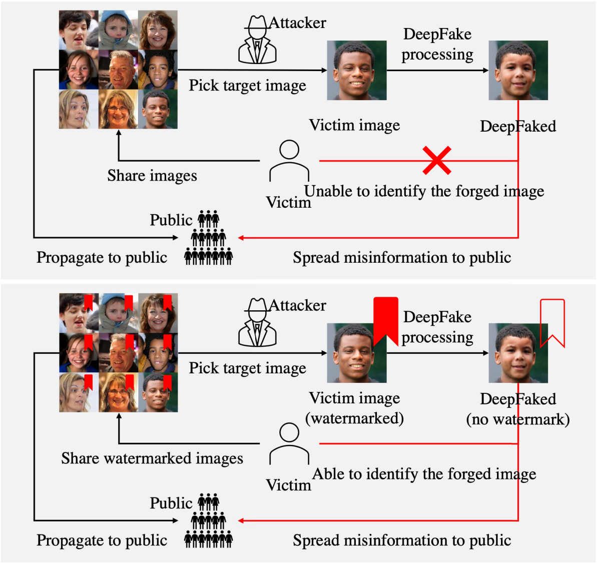


Figure 4.2: Comparison between the unprotected social network (top panel) and the social network protected by our method (bottom panel) in handling misinformation spreading by Deepfake attacker.

are useful when multiple devices are broadcasting in the same frequency range, such as the applications like CDMA and satellite navigation.

4.4 Application Scenario

Fig. 4.2 demonstrates how to deploy our method in the real world. In the scenario without applying our method (Fig. 4.2 top panel), a user shared his/her images to a social network, such as Facebook or Instagram. Once the image is uploaded online, the user cannot verify its authenticity anymore. Therefore, malicious users can not only pick victims' photos and manipulate them to create Deepfakes but also release the synthesized results while falsely claiming these images are authentic. The misinformation will cause severe reputation loss for the victim and raise security and privacy concerns.

On the contrary, in the scenario where the user employed our method (Fig. 4.2 bottom panel), the images shared online will be embedded with the user's personalized watermark. The embedded watermark is invisible to humans and robust to conventional manipulations. Thus there is negligible impact on the image's visual quality and application utility. Nevertheless, unlike other visible or invisible watermarking techniques, our watermark is sensitive to Deepfake manipulations. Once the malicious users apply Deepfake techniques on watermarked images, the corresponding embedded message will be destroyed. According to the existence of the watermark, the real and fake images can be effectively differentiated. Therefore, the user can utilize our method to reduce the negative impact of Deepfake by identifying the forged information and authenticating the real information.

4.5 Methodology

Our proposed framework includes two steps: watermark injection and watermark verification. We will introduce the details in this section.

4.5.1 Watermark Injection

The watermark injection step aims to insert a sequence into a face image to entangle with its identity feature while keeping the watermarked image perceptually similar to the original. The rationale behind this step is that slightly disturbing the identity feature while preserving residual attributes will not significantly distort the face image. Moreover, the conventional image modifications, e.g., cropping, resizing and compression, usually do not impact the identity feature of the facial images. Hence, the embedded watermark can avoid being modified and remain robust against conventional image manipulations. In contrast, Deepfake methods, whose objectives are editing or swapping the image's identity, will inevitably alter the inserted watermark. Therefore, we can utilize this mechanism to detect whether Deepfake modifies a watermarked image or not.

To this end, the watermark injection step consists of three processes: (1) Feature disentanglement, which disentangles the face image as two independent representations, namely identity and attributes; (2) Identity watermarking, which embeds a watermark into the extracted identity representation (vector); (3) Image reconstruction, which integrates the watermarked identity and original attribute to synthesize the corresponding watermarked image. The overview of the watermark injection is illustrated on the left part of Fig. 4.1, and architecture details are in Section 4.5.5.

Feature Disentanglement: Given an input face image, we employ two dedicated networks, namely identity encoder and attributes encoder, to respectively extract the independent representations, $z_{id}(X)$ and $z_{att}(X)$, from the image.

Identity Encoder: The identity representation is the high-level human biometric feature for characterizing a specific person with lesser intra-personal variations and larger inter-personal differences. Similar to most research for disentangling representations of identity and attributes [106, 142], the identity encoder in our work employs the pre-trained face recognition network [31] as the backbone to extract the input image's last feature vector generated before the final fully-connected layer as identity representation. Specifically, the identity representation is a 512-dimension vector, which is formulated as $z_{id}(X) = Arc(X)$, where X denotes the input image and $Arc(\cdot)$ represents the face recognition network.

Attributes Encoder: The attributes representation of face image is defined as spatial features such as pose, expression, background etc. According to the details of these features, attributes can be divided into different levels, from coarse (e.g., overall spatial outline), to fine (e.g., exact shape). Therefore, we adopt multi-level feature maps to preserve such details to represent the attributes. Specifically, we feed the input image into a U-Net style network and then use the feature maps generated from the U-Net decoder as attribute representations. The formal attributes representation is denoted as:

$$z_{att}(X) = \{z_{att}^{1}(X), z_{att}^{2}(X), ..., z_{att}^{n}(X)\}_2,$$

where $z_{att}^{n}(X)$ represents the n-th level feature map from the U-Net decoder, and n is the number of feature levels. The attributes encoder in this work does not require extra annotations as it extracts the attributes using self-supervised training, which is trained to keep the original image X and generated watermarked image $\hat{X}$ have the representation of the same attribute.

Identity Watermarking: After feature disentanglement, we add a bit-wise binary sequence z_{seq} to the identity representation $z_{id}(X)$ to generate the corresponding watermarked identity. The binary sequence can be user-defined or random-generated, which will serve as a signature for future verification, so the user of our method should preserve his/her embedded sequence and keep it secret from adversaries. Besides, to reduce the watermark's perturbation on the identity representation, we regulate it with a constant weight α. Unless otherwise stated, α will be set to 0.1 in our experiments. Therefore, the final watermark sequence values are minimal compared with the original identity sequence. The identity watermarking is formulated as:

$$z_{id}^{w}(X) = z_{id}(X) + \alpha z_{seq},$$

where $z_{id}^{w}(X)$ represents the watermarked identity vector.

Image Reconstruction: The subsequent process is to integrate watermarked identity and the original attributes to synthesize the watermarked image. Previous studies [10, 102] revealed that simply concatenating identity and attributes to synthesize images will incur severe visual quality degradation and distortion. To avoid this problem and generate the high-fidelity watermarked image, we employ a novel *Adaptively Attentional Denormalization* (AAD) [77] mechanism to accomplish feature integration.

The image reconstruction network adopts multiple cascaded AAD Residual Blocks (ResBlk) to integrate the identity and attributes. Each AAD ResBlk consists of multiple AAD layers, which employ an attention mechanism with denormalization to adaptively adjust the participation of identity representation and attribute representation for synthesizing different regions. For instance, the identity will provide more importance on generating the facial area, which is most discriminative for distinguishing identities, while the attributes will focus more on the regions related to spatial features, such as skin colour and background.

We formally define the reconstruction procedure as:

$$\hat{X} = Gen(z_{id}^{w}(X), z_{att}(X)) = z_{id}^{w}(X) \oplus z_{att}(X),$$

where the $\oplus$ denote the ADD ResBlk's integration and $Gen(\cdot)$ denote the reconstruction network.

4.5.2 Watermark Verification

Different from aiming to accurately recover the inserted watermark like the traditional watermark techniques [73, 113, 167], the objective of our watermark verification is to detect whether the watermark still exists in the watermarked image's identity feature and, in turn, determine whether Deepfake modifies this image or not. Since the difficulty of watermark detection is much easier than watermark recovery, our method can thus provide more reliable verification results.

In more detail, our watermark verification step consists of two processes: (1) Extraction, which extracts the input image's identity representation; (2) Verification, which calculates the correlation between extracted identity and pre-defined watermark to verify the existence of watermark in the input image.

Extraction: We re-use the identity encoder adopted in feature disentanglement to extract the identity representation from the watermarked image, which is formulated as $z_{id}(\hat{X}) = Arc(\hat{X})$. The rationale behind this process is that the watermarked identity integrated by our reconstruction process is believed to preserve in the watermarked image's identity feature, so we can use the same identity encoder to extract the corresponding watermarked identity representation. For the watermarked image not modified by Deepfake, the extraction process is defined as:

$$\begin{aligned} z_{id}(\hat{X}) &= Arc(\hat{X}) \\ &= Arc(z_{id}^{w}(X) \oplus z_{att}(X)) \\ &= Arc((z_{id}(X) + \alpha z_{seq}) \oplus z_{att}(X)) \\ &\approx z_{id}(X) + \alpha z_{seq}. \end{aligned}$$

The attributes $z_{att}(X)$ is omitted because $Arc(\cdot)$ only extract identity features.

Verification: After obtaining the identity representation $z_{id}(\hat{X})$, we calculate its correlation with the watermark sequence z_{seq} to verify whether the watermark is present in the input image. Since the $z_{id}(\hat{X})$ and z_{seq} can be regarded as 1-dimensional real discrete sequences, the function computes the correlation of them is defined as:

$$Corr[l] = \sum_{n=0}^{N-1} z_{id}(\hat{X})[n] * z_{seq}[n - l + N - 1],$$

where $l = 0, 1, ..., 2N - 2$ is the index for correlation result,n denotes the index for discrete sequences, N represents the their length, and $z_{seq}[m]$ is 0 when m is outside of the range of z_{seq}.

As demonstrated in Eq. 4.5.1, for the real watermarked image, the correlation function in Eq. 4.5.2 equals to:

$$\begin{aligned} Corr[l] &= \sum_{n=0}^{N-1} z_{id}(\hat{X})_{rec}[n] * z_{seq}[n - l + N - 1] \\ &\approx \sum_{n=0}^{N-1} (z_{id}(X)[n] + \alpha z_{seq}[n]) * z_{seq}[k] \\ &= \sum_{n=0}^{N-1} z_{id}(X)[n] * z_{seq}[k] + \alpha z_{seq}[n] * z_{seq}[k], \end{aligned}$$

where we set $k = n - l + N - 1$ for simplicity. Therefore, the correlation between extracted identity and the pre-defined watermark can be assumed as the sum of two independent calculations: cross-correlation of original identity representation with watermark sequence, and auto-correlation of watermark sequence itself. In contrast, if the watermarked image is modified by Deepfake, its identity representation and entangled watermark sequence will be distorted, so the correlation between its extracted identity and the pre-defined watermark cannot factorize like Eq. 4.5.2 but can only be assumed as two different sequences' cross-correlation like Eq. 4.5.2.

According to the auto-correlation's property, the maximum correlation value will appear at the index of $(N-1)th$. While for the cross-correlation, there is no such property.

Hence, we can detect if there is a distinct peak value at the $(N-1)th$ index to determine whether the watermarked image's identity feature is tampered with by Deepfake methods.

4.5.3 Training Procedure

No extra annotations are required in our training procedure, and except for the identity encoder, all other networks are trainable.

Adversarial Loss: To make the reconstructed image more realistic, we employ a multi-scale discriminator $Dis(\cdot)$ from [66] with hinge loss functions to train our model in an adversarial way:

$$\mathscr{L}_{Adv} = \log Dis(X_m) + \log(1 - Dis(\hat{X}_m)),$$

where X_m and $\hat{X}_m$ indicate the low-resolution original and corresponding reconstructed image after m-th down-sampling.

Attributes Preservation Loss: We also calculate the attributes representations' $\mathscr{L}_2$ distance between original and reconstructed image to enforce attributes preservation:

$$\mathscr{L}_{Att} = \frac{1}{2}\sum_{k=1}^{n}\left\| z_{att}^{k}(X) - z_{att}^{k}(\hat{X})\right\|_2^2,$$

where the n denotes the level of attributes.

Reconstruction Loss: In addition, to keep the reconstructed image resemble the original and mitigate the conflict with watermark injection at pixel-level, we define a perceptual similarities loss LPIPS [159] between the original and reconstructed image rather than the common pixel-level reconstruction loss:

$$\mathscr{L}_R = \left\| L(X) - L(\hat{X})\right\|_2,$$

where $L(\cdot)$ represents the perceptual features extractor.

Watermark Preservation Loss: To minimise the distortion of embedded watermark sequence in the reconstructed image, a watermark preservation loss function is used to measure the cosine similarity between the watermarked identity vector and extracted identity vector:

$$\mathscr{L}_W = 1 - CosineSimilarity(\hat{z}_{id}(X), Arc(\hat{X})),$$

where $Cos(\cdot)$ denotes the operation of cosine similarity.

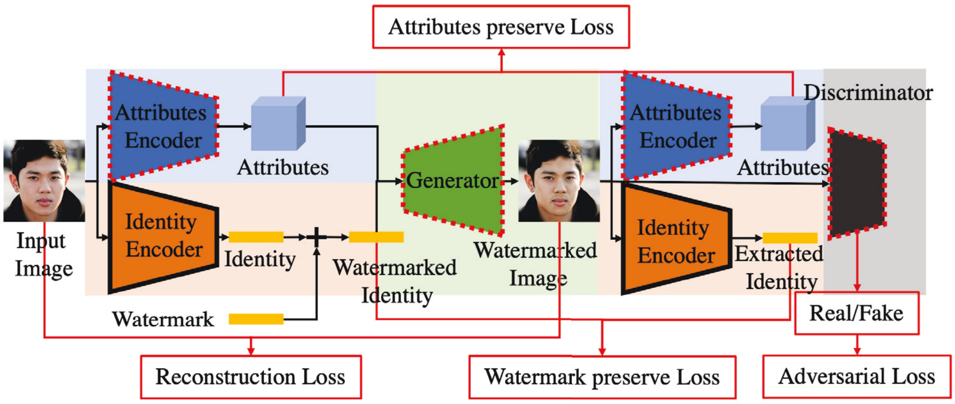


Figure 4.3: Training procedure

Our framework is finally trained with a weighted sum of the above losses, which is defined as:

$$\mathscr{L}(X) = \lambda_R \mathscr{L}_R + \lambda_{Adv} \mathscr{L}_{Adv} + \lambda_{Att} \mathscr{L}_{Att} + \lambda_W \mathscr{L}_W,$$

where $\lambda_R, \lambda_{Adv}, \lambda_{Att}, \lambda_W$ are tunable constant weighting corresponded loss. Unless stated otherwise, the λ values are set as $\lambda_R = 10, \lambda_{Adv} = 0.1, \lambda_{Att} = 10$ and $\lambda_W = 1$.

4.5.4 Training Details

The concept diagram of our framework's training procedure is illustrated in Fig. 4.3. In Fig. 4.3, black arrows refer to data flows from the input image to the watermarked image and Discriminator, while the red lines indicate data flow for each loss function. Besides, the trapezoids with a red dash represent a trainable network, while black sold line trapezoids represent fixed networks. We utilize adversarial training for our framework where the Discriminator adopts a multi-scale network [110].

Given an input face image X, the identity encoder and attributes encoder respectively extract the 1D 1x512 identity representation $z_{id}(X)$ and multi-level attributes representation $z_{att}(X)$ from the image. Then we bit-wise add watermark sequence z_{seq} to the identity representation $z_{id}(X)$ to produce watermarked identity $z^w_{id}(X)$. The Generator finally integrates the watermarked identity $z_{id}(X)$ and original attributes representation $z_{att}(X)$ to produce the watermarked facial image $\hat{X}$. After obtained $\hat{X}$, we calculating different losses according to E.q. 4.5.3,4.5.3,4.5.3,4.5.3 and update all trainable networks. As we can see in the scheme, non-extra annotations are required in our training process.

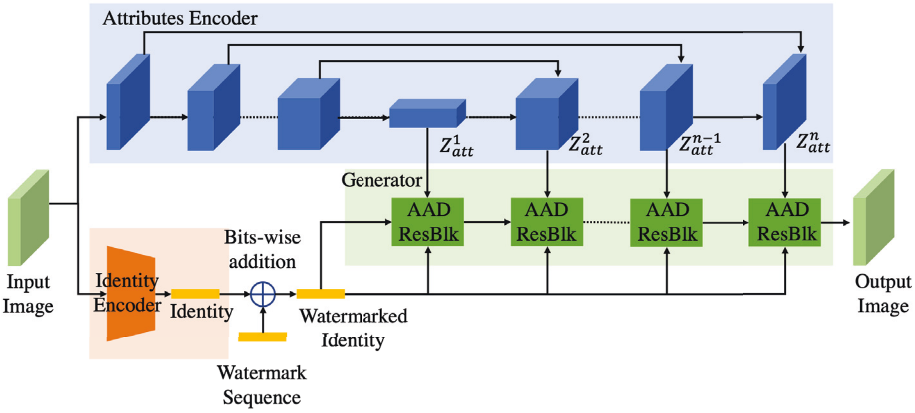


Figure 4.4: Detailed architecture of Watermark Injection.

4.5.5 Network Architecture

Our framework is built upon FaceShifter's AEI-Net [77] but modified by us to implement face identity watermarking rather than face-swapping. The entire network, illustrated in Fig. 4.4, consists of an Identity Encoder, a U-Net style Attributes Encoder, and a Generator composed of 8 cascaded AAD Residual Blocks. Detailed structures of each component are in Fig 4.5.

4.6 Experiment

We conduct extensive experiments to evaluate our method from the following aspects: i) the different pseudo-random sequences' impacts on model performance, ii) effectiveness in Deepfake detection, iii) visual quality of watermarked images, and iv) security in potential attack scenarios. The experiment results demonstrate that our method can achieve the best performance regarding various qualitative and quantitative evaluation metrics. Note that the security analysis results are in Section 4.6.6.

4.6.1 Experiment Setup

Datasets: We train our method on the **Flickr-Faces-HQ (FFHQ) [71]** dataset, and conduct experiments on **CelebA-HQ [70]** and **CelebA [88]** datasets to reveal its generalizability. Unless stated otherwise, all images in the experiment have been aligned and cropped to the size of 256×256.

Baselines: We select the work whose authors released the source codes and pre-trained models in our comparison experiment for results reproducible.

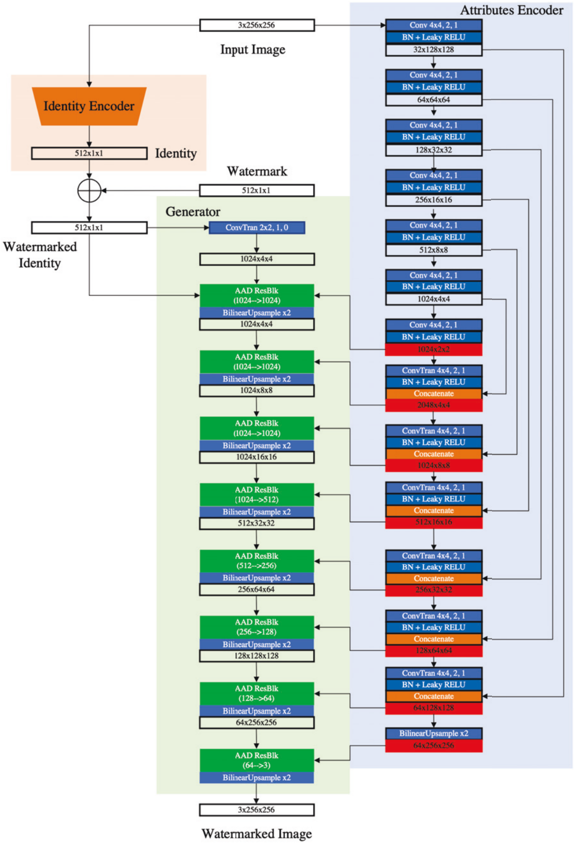


Figure 4.5: Network structures. Following the structures of FaceShifter [77]: *Conv k,s,p* represents a Convolutional Layer with kernel size k, stride s and padding p. *ConvTran k,s,p* represents a Transposed Convolutional Layer with kernel size k, stride s and padding p. All *Leaky ReLUs* have $\alpha = 0.1$. *AAD ResBlk*(c_{in}, c_{out}) represents an AAD ResBlk with input and output channels of c_{in} and c_{out}.

Deepfake Detection: Passive methods [17, 21, 53, 136] and proactive method [153] are selected because they represent the latest reproducible Deepfake detection methods.

Digital Watermarking: We chose StegaStamp [129] and UDH [155] as the baseline because they achieve the SOTA performance in embedding information and exhibit appealing visual quality results.

We use the official codes and pre-trained models for all the above-mentioned methods.

Evaluation Metrics: We evaluate the performance using three different categories of metrics: (1) For both Deepfake detection, to measure the miss detection rate and false alarm rate, we compute image-level **Accuracy(ACC)** and **F1-Score**. **AUC** and related **ROC** curve, which are decision-threshold-free metrics, are also reported to select optimal models; (2) Regarding robustness evaluation, the proportion of correctly detected watermarked images after various post-processing is calculated and denoted as **Detection Ratio (DR)**; (3) **Peak-Signal-to-Noise Ratio (PSNR)** and **Structural Similarity Index Measure (SSIM)** are used to calculate the similarity between the watermarked images and original images to show the visual quality of watermarked images.

The above metrics are higher when the associated methods show better performance, except stated otherwise.

4.6.2 Impacts of the Types of Sequences

According to Eq. 4.5.2, the correlation property of the embedded sequence plays a significant role in the watermark verification. Thus, we first analyse the impact of the different watermark sequences on our method. Two representative Pseudorandom Noise (PN) sequences: Maximum-Length Sequence (MLS) and Gold Code (Gold), and two most common random sequences: Gaussian noise and Laplace noise, are selected for comparison. All embedded sequences are set to a length of 512, the same as the identity representation. For a fair comparison, the network is trained FFHQ images by randomly selecting the above watermarks in each iteration. Then, we apply four different sequences to the 10k randomly chosen Celeba-HQ images to generate watermarked testing images, resulting in 40k testing images and calculate the defined metrics over these images to evaluate the impact.

Correlation Results. In Table 4.1, we present the averaged correlation results, where ***Peak*** denotes the correlation value appear at the zero-lag, ***Average*** represents the mean of the residual correlation results, and ***Peak-to-Average Ratio (PAR)*** outputs a ratio of ***Peak*** over ***Average***, which indicates how significant the ***Peak*** stands out in the correlation results. Except for the results from watermarked images, we also report the correlation of original images in Table 4.1's first row, which acts as a reference. As shown in Table 4.1, the watermarked images' Peak and PAR are significantly higher than the reference, where the Gold sequence achieves the highest values. The apparent difference between watermarked and non-watermarked images' correlation results demonstrates that our method can effectively embed and extract the watermark in images.

Visual Quality. Afterwards, we evaluate the visual quality of images after watermarking different sequences. The best SSIM and PSNR values are also reported in Table 4.1's Original row for reference. According to the quantitative and qualitative results exhibited in Table 4.1 and Fig. 4.6, no matter what types of sequence are embedded, the watermarked images can maintain high visual quality and look perceptually

Table 4.1: Different sequences' correlation results and corresponding watermarked images' visual quality.

Sequences types	**Correlation results**			**Visual Quality**	
	Peak	**Average**	**PAR ↑**	**SSIM ↑**	**PSNR ↑**
Original	0.77	0.54	1.43	1.0	48.0
Gaussian	0.96	0.54	1.79	0.95	34.65
Gold	5.61	0.52	**10.83**	0.94	33.32
Laplace	1.82	0.74	2.45	0.95	34.84
MLS	4.82	0.53	9.17	0.95	33.5

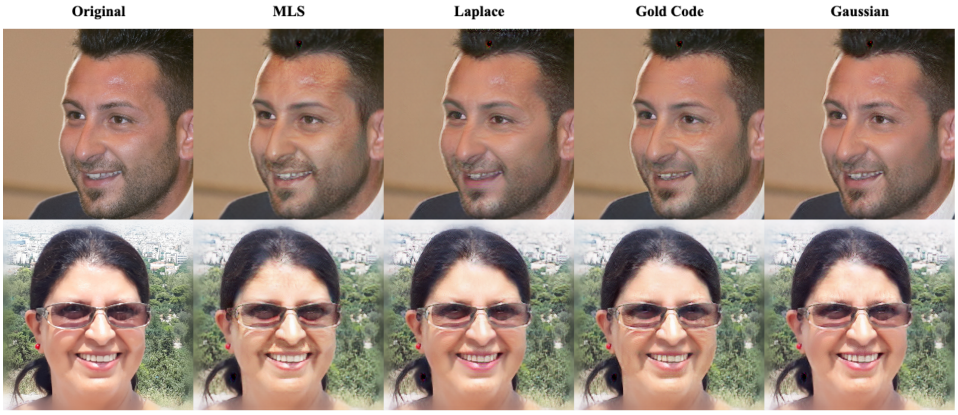


Figure 4.6: Qualitative comparison of different sequences' watermarked images. Despite injecting different sequences, all watermarked images are perceptually identical to the original images.

identical to the original, which demonstrates that watermarking images via our method would not affect its utility.

Effectiveness. In this section, we explore the effectiveness of our method in identifying the watermarked or non-watermarked images. Our method discriminates 10k watermarked images and 10k randomly chosen Celeba-HQ non-watermarked images for different types of watermark sequences. Besides, according to the correlation results summarized in Table 4.1, different types of watermark sequences have different PARs. Here, we consider adopting different PAR values ranging from 1 to 10 with a step size of 1 as the threshold to decide whether a watermark exists in the corresponding image. More specifically, an image with PAR higher than the threshold in its correlation results will be regarded as watermarked. We calculate related ACC and F1 Scores further to analyze our method's discriminability under different PAR thresholds, and also plot the ROC curve and compute corresponding AUC to provide more convincing results.

Fig. 4.7 displays the experiment results. According to the trend of ACC and F1 Score curves, the optimal thresholds for different sequences are 2 for Gaussian and Laplace, 5 for Gold and 4 for MLS. We will adopt these thresholds in the subsequent

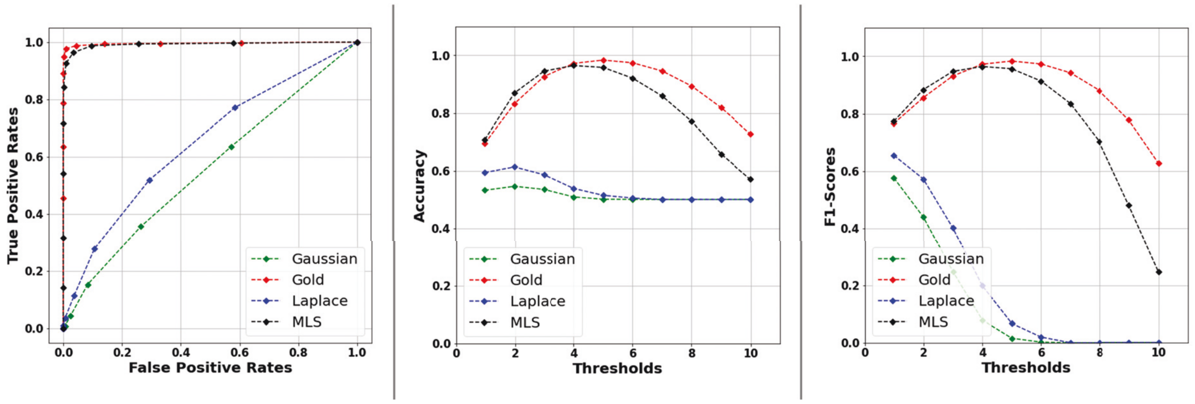


Figure 4.7: **ROC** ↑, **Accuracy** ↑ and **F1 Score** ↑ of different sequences under different PAR thresholds. Gold and MLS sequences' performance to discriminate between watermarked or non-watermarked images are superior to Gaussian and Laplace.

Table 4.2: **Detection ratio** ↑ of different sequences against color adjustment and horizontal flipping.

Image Manipulations	**Sequences types**			
	Gaussian	Gold	Laplace	MLS
ColorJitter	63.9%	**94.8%**	45.4%	91.2%
Flip	61.9%	**97.2%**	52.4%	96.7%

robustness evaluations. Besides, their AUC values are 0.5552, 0.9952, 0.6466 and 0.9917 for Gaussian, Gold, Laplace and MLS, respectively. The ROC curve of Gold and MLS are much closer to the top left than Gaussian and Laplace. These results indicate that adopting Gold and MLS as watermark sequences would perform our method better than Gaussian and Laplace.

Robustness. We test the impact of different sequences on our method's robustness. Five common post-processing operations are adopted in the experiment, i.e., Gaussian blurring, colour adjustment, JPEG, horizontal flipping, resizing and cropping. For Gaussian blurring, we consider kernel standard deviation ranging from 0.5 to 1.0 with a step size of 0.1. For JPEG, we consider quality factors ranging from 50 to 100 with a step size of 10. For resizing and cropping, we consider first cropping the image's peripheral sizes ranging from 50% to 100% with a step size of 10% and then resizing it to 256×256. For Horizontal flipping and colour adjustment, we employ the PyTorch torchvision.transformations' functions RandomHorizontalFlip with the probability of the image being flipped set as 1.0 and ColorJitter with the default setting to achieve all image's horizontal flipping and randomly brightness, contrast, saturation and hue change. Examples of the modification are visualized in Fig. 4.8.

For each sequence's watermarked images, we apply the above operations to generate

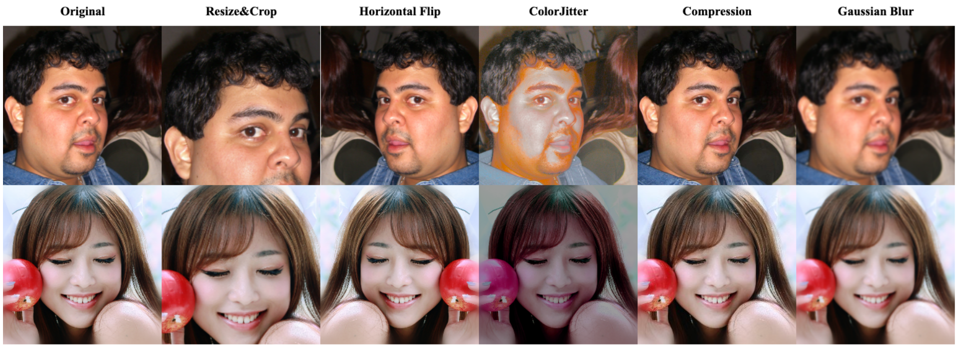


Figure 4.8: Samples of each post-processing operation results adopted in our experiment.

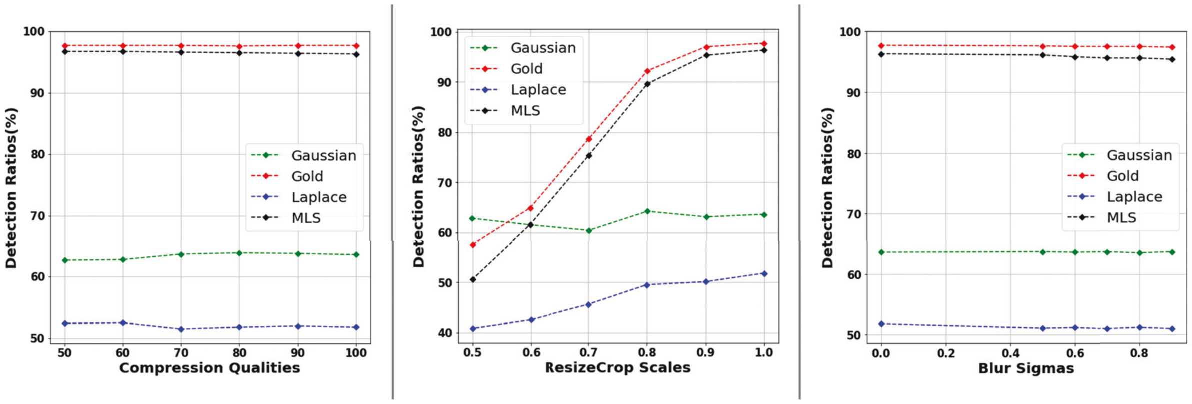


Figure 4.9: **Detection ratio** ↑ of different sequences against JPEG, Resize-Crop and Gaussian Blur.

corresponding images and then employ our method to detect watermarks from these processed images and compute the detection ratio **DR**. Table 4.2 and Fig. 4.9 present each sequence's robustness performance. Our method shows a minor performance degradation when dealing with compression and blur but is susceptible to resizing and cropping. As illustrated in Row 1 Column 2 of Fig. 4.8, the main reason is that a crop size smaller than 80% would cut off partial facial regions, damaging the corresponding identity feature. However, this problem is not severe because a cropped face image is unlikely to be used in practice.

The results of Gold and MLS watermark sequences reflect our method's robustness against these image post-processing, where the Gold sequence achieves the best performance, slightly superior to MLS but much better than Gaussian and Laplace. Therefore, we will adopt the Gold sequence as the watermark to compare our method with other works in Deepfake detection.

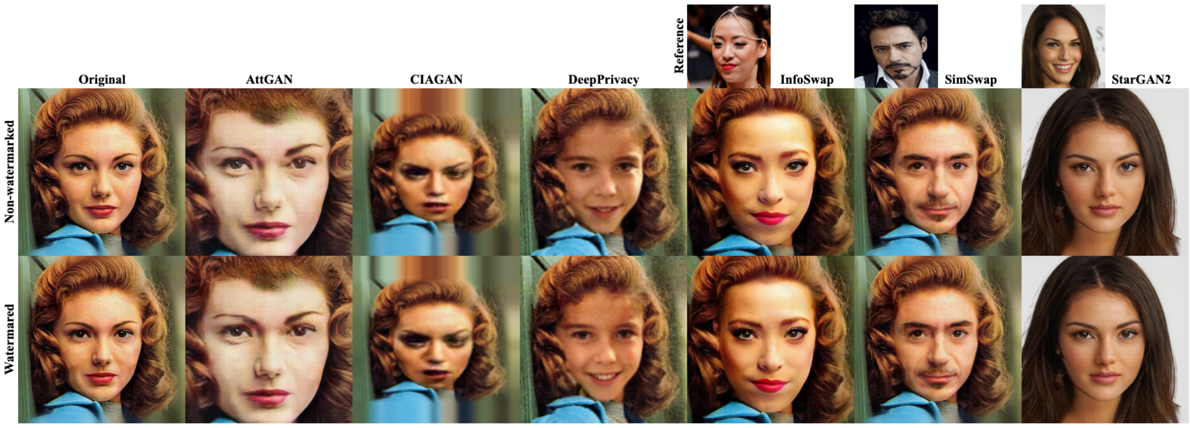


Figure 4.10: Samples of non-watermarked and our watermarked images' Deepfake results. The watermarked image's forgery result is perceptually identical to the non-watermark image's.

4.6.3 Deepfake Detection

We compare our method with other Deepfake detection approaches in image-level real or fake classification. Two attributes manipulation methods, i.e., AttGAN [54] and StarGAN2 [23], two identity swap methods, i.e., InfoSwap [47] and SimSwap [22], and two face anonymization approaches, namely CIAGAN [97] and DeepPrivacy [63] are employed in this experiment. We adopt the official codes and pre-trained models of these works, so our experiment results are reliable and reproducible, which thus can refer for future comparison.

Celeba and CelebaHQ images are employed in this experiment to represent low- and high-resolution Deepfake cases. We apply Deepfake methods to watermarked and non-watermarked images to generate corresponding fake outputs. The watermarked real and fake images are utilised to evaluate our method's discriminability, while the non-watermarked real and fake images are adopted to evaluate other detection methods' performance. According to the analysis of different sequences' performance, our method adopts the Gold sequence as the embedded watermark in the comparison experiment and sets the PAR threshold to 5.

We first illustrated Fig. 4.10 to have a qualitative comparison of Deepfaked non-watermarked and watermarked outputs where we can see that the non-watermarked and our watermarked images' Deepfake results are perceptually identical to each other, demonstrating that our method well maintains the utility of the image. Moreover, it also makes it hard for the Deepfake adversaries to distinguish the protected and non-protected images, increasing our method's secrecy.

Table 4.3 summarizes the comparison results between ours with passive methods.

Table 4.3: **Accuracy** ↑ and **F1 Scores** ↑ of different methods' DeepFake detection results.

Detection methods	**Low Resolutions(Celeba)**						**High Resolutions(Celeba-HQ)**					
	AttGAN	**CIAGAN**	**DeepPrivacy**	**InfoSwap**	**SimSwap**	**StarGAN2**	**AttGAN**	**CIAGAN**	**DeepPrivacy**	**InfoSwap**	**SimSwap**	**StarGAN2**
BTS [53]	0.86/0.87	0.51/0.66	0.5/0.66	0.49/0.66	0.51/0.67	0.53/0.67	0.86/0.87	0.5/0.66	0.5/0.66	0.49/0.65	0.5/0.66	0.55/0.68
CD [139]	0.88/0.86	0.51/0.03	0.51/0.01	0.54/0.17	0.51/0.01	0.78/0.71	0.81/0.77	0.51/0.04	0.52/0.07	0.52/0.07	0.52/0.06	0.84/0.81
ICPR [17]	0.59/0.69	0.62/0.62	0.58/0.68	0.49/0.64	0.6/0.71	0.46/0.63	0.53/0.68	0.65/0.62	0.56/0.69	0.51/0.66	0.55/0.69	0.49/0.65
PF [21]	0.76/0.79	0.51/0.66	0.52/0.65	0.57/0.68	0.54/0.67	**0.99/0.98**	0.75/0.79	0.51/0.66	0.55/0.68	0.56/0.69	0.54/0.68	**0.98/0.97**
RFM [136]	0.5/0.67	0.51/0.67	0.51/0.67	0.5/0.67	0.51/0.67	0.5/0.67	0.5/0.67	0.5/0.67	0.51/0.67	0.5/0.67	0.5/0.67	0.5/0.67
SBI [123]	0.79/0.82	0.77/0.8	0.78/0.82	0.77/0.81	0.78/0.81	0.72/0.75	0.8/0.8	0.72/0.78	0.78/0.8	0.76/0.77	0.83/0.84	0.69/0.7
Ours	**0.94/0.94**	**0.87/0.86**	**0.98/0.98**	**0.98/0.98**	**0.97/0.98**	0.82/0.84	**0.94/0.94**	**0.85/0.82**	**0.99/0.98**	**0.99/0.99**	**0.98/0.98**	0.85/0.87

Our method achieves more than 0.8 ACC and F1 Scores on detecting all Deepfake methods' outputs, revealing its superior effectiveness and generalization. Except PF perform better than ours on StarGAN2, our method outperforms all other baselines with a clear margin. Our method performs poorly on StarGAN2 (still achieves second-rank performance) because StarGAN2 does not modify face identity-related features. To verify this, we employ AttGAN to manipulate identity-related attributes, e.g., gender and skin colour. The result in Table 4.3 shows that our method can accurately detect these manipulated images. On the contrary, other passive detection methods only perform well in detecting limited Deepfake methods, deteriorating to random guesses (50% accuracy) in detecting other Deepfake methods.

Then, we compare our method with the latest proactive detect method, i.e., AGF [153], to detect FaceShifter's Deepfake outputs on Celeba-HQ images and present the detection results in Table 4.4. According to the requirement of AGF, we employ AGF to fingerprint 15k Celeba images and train FaceShifter model A on these fingerprinted images. Then, we use trained FaceShifter model A to generate 5k Deepfake results on original Celeba images without AGF fingerprint. We employ the AGF to conduct detection on the set, which mixes the 5k Deepfaked images with another randomly selected 5k original Celeba images and summarized results in Table 4.4.

We train another FaceShifter model B for our method on the same 15k Celeba images but without AGF fingerprints. We employ our method to inject watermarks into 5k Celeba images, which are the same image in the AGF process. Then, we use trained FaceShifter model B to generate Deepfake results based on our watermarked images and mix another 5k original Celeba image to form a test set. We employ our method to conduct detection on this test set and report the results in Table 4.4.

Although AGF achieves impressive detection performance, with 0.99 Recall and 0.91 F1 Score, our method still beats it on almost all metrics (only 0.005 lower Recall which is negligible). In particular, our method has a significant advantage in detection accuracy due to AGF producing more false alarms on authentic images (only 0.84 Precision).

We also report the AUC of our method on different Deepfakes and datasets in Table 4.5

Table 4.4: DeepFake detection performance of proactive methods.

	Acc ↑	**F1 Scores ↑**	**Precision ↑**	**Recall ↑**
AGF [153]	0.7179	0.9148	0.8444	0.9980
Ours	0.9955	0.9955	0.9980	0.9930

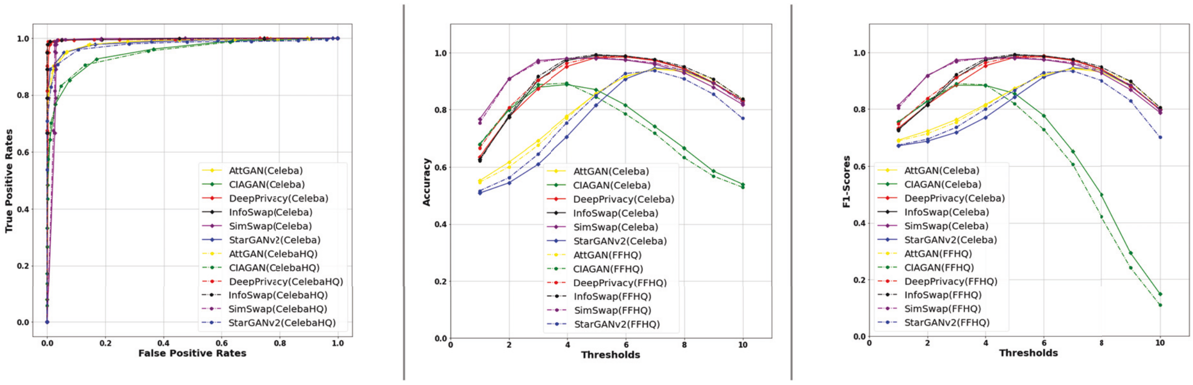


Figure 4.11: Our method's **ROC** ↑, **Accuracy** ↑ and **F1 Score** ↑ on different Deepfake and datasets images. The ROC curve indicates our method has excellent discriminability on both low- and high-resolution Deepfake results. Besides, according to the trend of Accuracy and F1 Scores, our method can achieve different performances under different PAR thresholds.

Table 4.5: **AUC** ↑ of our method on different Deepfake and datasets.

Datasets	**DeepFake Methods**					
	AttGAN	CIAGAN	DeepPrivacy	InfoSwap	SimSwap	StarGAN22
Celeba	0.98	0.95	0.99	0.99	0.98	0.98
CelebaHQ	0.98	0.95	0.99	0.99	0.98	0.98

and plot the ROC, accuracy and F1 Scores curves when adopting different thresholds in Fig. 4.11. We plot our method's ROC, accuracy, and F1-Score curves when adopting decision thresholds ranging from 1 to 10 with a step size of 1. As illustrated in Fig. 4.11, our method can achieve excellent classification capability when facing different Deepfake methods. Besides, the F1 Score and ACC curves indicate that our method has better detection performance on CIAGAN and StarGAN2 when adopting a threshold of 3 and 7, respectively. These results reveal our method's exceptional image level real or fake classification capability when facing different Deepfake methods. In general, the experiment results demonstrate that our method has superior Deepfake detection performance to existing methods.

Table 4.6: Quality of different watermarking methods' outputs.

Quality metrics	Proactive Detection		Deep Hiding	
	Ours	AGF	UDH	StegaStamp
SSIM ↑	**0.94**	0.91	0.69	0.89
PSNR ↑	**33.32**	30.69	20.39	29.77

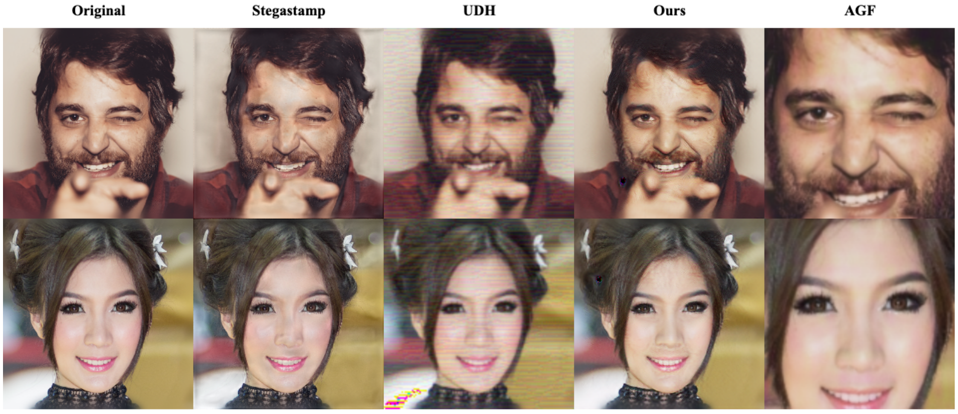


Figure 4.12: Qualitative comparison between our method, AGF and SOTA watermarking techniques StegaStamp and UDH.

4.6.4 Digital Watermarking

We compare our method with the AGF and SOTA digital watermarking techniques, namely Stegastamp and UDH, in the visual quality of watermarked images to show that our method does not sacrifice the normal utility of images. Here, we adopt the widely used metrics PSNR and SSIM to quantitatively reflect comparison results in Table 4.6. The results demonstrate that the outputs of our method have much better visual quality than others. Fig. 4.12 also illustrates perceptual comparison, where our watermarked images are more visual-realistic which accurately preserves the hue and light of the original images. In contrast, UDH introduces apparent artifacts in its watermarked images. StegaStamp's outputs have noticeable colour distortion in the facial area. Therefore, qualitative and quantitative results indicate that our method can generate high-quality images with a robust watermark.

4.6.5 Computational Overhead

According to Table 4.7, the primary computational cost of our method is the watermark injection, while the verification(detection) stage's overheads are close to other detection approaches. Considering the detection performance of our method, we believe it is a balanced trade-off between computational costs and detection accuracy.

Table 4.7: Parameters and FLOPs of different detection methods.

	Parameters	FLOPs
BTS	45.907M	95.1557G
CD	23.5101M	5.3965G
ICPR	0.1271M	0.8869G
PF	23.51M	2.1G
RFM	20.8111M	6.0116G
SBI	0.1288M	0.1981G
Ours		
Injection	396.0907M	83.8398G
Verification	43.7977M	6.3236G

4.6.6 Security Analysis

To verify the security of our method of confronting worst-case threaten, we simulated attacks to the application scenarios of our method. The objective of the adversaries is to utilize the knowledge about our watermarking mechanism to forge the watermark. Here, we consider three strong attack models:

Attack Model 1: The adversary can obtain the victim's entire watermarking framework and its corresponding pre-trained models but knows nothing about the watermark sequence. Thus, the adversary tries to embed different sequences on Deepfaked images via the obtained network to deceive the watermark verification step.

Attack Model 2: The adversary knows the victim's watermark sequence but cannot obtain the corresponding framework. Thus, the adversary utilizes the knowledge about our method trying to build and train a similar watermarking network to embed the victim's watermark on Deepfaked images to deceive the watermark verification step.

Attack Model 3: The adversary stealthily collects the victim's watermarked images and employs these images as dataset to train his/her Deepfake method to generate fake images to deceive the watermark verification step.

Attack Model 1 For Attack Model 1, we simulate the forging process by utilizing one fixed watermarking framework to synthesize four groups of watermarked images according to different types of sequences. Then, we randomly generate another sequence to act as the victim's watermark and use the same framework to implement watermark verification on these watermarked images. As can be seen from Table 4.8, the correlation results of different watermarked images are close to non-watermarked images. Therefore, even if the victim's watermarking framework leaked, the forged watermarked image

Table 4.8: Correlation results of Attack Model 1.

Sequences types	**Auto-correlation results**			**DR**
	Peak	**Average**	**PAR**	
Gaussian	0.78	0.53	1.47	0.6%
Gold	0.95	0.53	1.79	2.98%
Laplace	1.01	0.53	1.89	0.13%
MLS	0.77	0.53	1.46	1.6%

Table 4.9: Correlation results of Attack Model 2.

Networks	**Auto-correlation results**			**DR**
	Peak	**Average**	**PAR**	
$Model_{Arc}$	0.71	0.53	1.33	1.28%
$Model_{Cir}$	0.73	0.53	1.37	1.63%

cannot pass the corresponding watermark verification step.

Attack Model 2 For Attack Model 2, we employ another face recognition network, namely circularface, as the identity encoder to constitute a new watermark framework then utilize both arcface $Model_{Arc}$ and circularface $Model_{Cir}$ watermark networks to generate watermarked images by embedded same sequence respectively. To simulate the deceive process, we use $Model_{Arc}$'s network to verify $Model_{Cir}$'s outputs and $Model_{Cir}$'s network to verify $Model_{Arc}$'s outputs. The correlation results reported in Table 4.9 demonstrate that different frameworks cannot generate the same watermarked results even embedded in the same sequence.

Attack Model 3 For Attack Model 3, we train Faceshifter [77] by 40k Gold sequence watermarked images to imitate the attack scenario and test the specific performance. The Faceshifter is one of the most representative Deepfake methods which generates Deepfake image by utilizing the source image's identity and target image's attributes. Hence, the fake image from Faceshifter would preserve the source's identity feature and the target's attributes feature.

After training the Faceshifter model to coverage, we subdivide the Deepfake generate process into six different cases: 1) Using the non-watermarked image as source and non-watermarked image as the target, denoted as $Id_{ori}Att_{ori}$; 2) Using the non-watermarked image as source and watermarked image as the target, denoted as $Id_{ori}Att_{wat}$; 3) Using the watermarked image as source and non-watermarked image as the target, denoted as $Id_{wat}Att_{ori}$; 4) Using the watermarked image as source and watermarked image as the target, denoted as $Id_{wat}Att_{wat}$; 5) Using the non-watermarked image as source and non-watermarked image as the target but verification with different Gold sequence,

Table 4.10: Correlation results of Attack Model 3.

Networks	Auto-correlation results			DR
	Peak	Average	PAR	
$Id_{ori}Att_{ori}$	1.13	0.52	2.15	3.6%
$Id_{ori}Att_{wat}$	0.58	0.52	1.11	0.1%
$Id_{wat}Att_{ori}$	3.02	0.53	5.68	92.9%
$Id_{wat}Att_{wat}$	3.6	0.53	6.77	94.67%
$Id_{wat}Att_{ori}DS$	0.58	0.53	1.1	1.33%
$Id_{wat}Att_{wat}DS$	0.64	0.52	2.62	1.2%

denoted as $Id_{wat}Att_{ori}DS$; 6) Using the non-watermarked image as source and non-watermarked image as the target but verification with different Gold sequence, denoted as $Id_{wat}Att_{wat}DS$.

The experiment results are summarized in Table 4.10. We can see even the Faceshifter is trained with the watermarked images, but unless the fake image is generated from the source image which has the same watermarked sequence with the verification model, otherwise the correlation results and DR are the same with the non-watermarked image.

4.6.7 Limitations

First, our method requires pre-processing, which will introduce computational overhead (in Section 4.6.5) and cannot perform detection of already synthesized images. Therefore, we think the best application case of our method is to protect critical images and employ our method before spreading them over the social network. Second, current watermarking is embedded in the identity features of face images, as we believe it presents the most severe threat if a person's identity is faked. Our method needs further improvement to adapt to other types of Deepfake content.

4.7 Conclusion

This work poses a proactive method to protect face images from malicious Deepfake. By embedding an invisible watermark into the face image's identity, our method provides users with a reliable approach to verifying their image's authenticity, reducing the negative impact of Deepfake forgery. The experiment results have demonstrated our method's superior performance in identifying Deepfake, preserving reconstructed images' visual quality, retaining watermarked sequence robustness, and resilience to potential malicious attacks.

Upon exploring the intricacies of forgery media detection in Chapter 4, we will venture into Chapter 5, where the emphasis shifts to media authenticity protection. Chapter 5 builds upon the previous discussions on detecting forgeries by proposing proactive measures to embed verification mechanisms directly into media files. The chapter delves into innovative watermarking techniques, demonstrating how they can be used to establish the authenticity of digital images, thereby adding a crucial layer of security against tampering and unauthorized alterations. Specifically, the next chapter proposes a proactive approach to safeguarding media content before malicious actions. This method embeds an invisible watermark pixel-by-pixel into an image to locate tampered regions effectively. It features a novel deep learning-based semi-fragile image watermarking framework, achieving a balance between detection performance and imperceptibility and comprehensive evaluations of its performance against various tampering types.

CHAPTER 5

MEDIA AUTHENTICITY PROTECTION

5.1 Preface

In Chapter 5, we tackle a critical issue in digital media: ensuring the authenticity of image data to safeguard privacy and security amidst the increasing prevalence of malicious image tampering. This emerging threat, where images are intentionally manipulated to harm owners or users, poses severe challenges to image authenticity. Despite the evolution of image manipulation techniques that leave fewer detectable traces, traditional methods have persistently tried to detect tampering by identifying visual artifacts and distortions with limited success.

Given the limitations of conventional methods, we propose a proactive solution. This chapter details the proposed deep learning-based semi-fragile watermarking scheme for media authentication. This watermarking scheme embeds an invisible watermark into a target image, entangled pixel-by-pixel. This watermark acts as an indicator of tampering trials and exhibits changes once the watermarked image is counterfeited. By comparing retrieved and original watermarks, we can locate the tampered regions.

This proactive authentication mechanism makes our method effective against various image tamper techniques, including image copy&move, splicing, and in-painting. Our watermark is designed to be fragile to malicious tampering operations, but it remains robust to benign image-processing operations such as JPEG compression, scaling, saturation, and contrast adjustments. This robustness enables our watermark to retain effectiveness even when images are shared online.

This chapter further discusses the implementation of our semi-fragile watermarking scheme, presenting comprehensive results to illustrate its effectiveness against various tampering techniques. Our extensive experiments demonstrate that our method achieves state-of-the-art forgery detection with superior robustness, imperceptibility, and security performance. In doing so, we hope to pave the way for robust defence mechanisms for image data, contributing significantly to the broader fight for media authenticity.

5.2 Introduction

Digital images have become an essential medium for information transmission in our society. However, technical advancement makes tampering images imperceptibly, which can be exploited for malicious intentions, e.g., creating fake news and Internet rumours. Therefore, detecting the tampered regions in an image is essential to protect image authenticity.

State-of-the-art detection methods leverage deep learning techniques by distinguishing feature distribution inconsistency [60, 75, 145] or boundary discrepancy [121, 164] in an image to identify the forgery or any manipulation. Those methods assume that image manipulation techniques may inevitably produce detectable artifacts in their outputs. For example, [145] detect forgery pixels by identifying local anomalous features in suspicious images. However, this prerequisite might lead to several inherent drawbacks. First, as image manipulation techniques progressively evolve, fake images exhibit less noticeable artifacts. As a result, detection methods developed to detect certain artifacts would be failed with a high chance. Moreover, existing methods trained on seen tampering types might fail to detect unseen counterfeits.

Besides, some methods [5, 138, 153] detect malicious tampering in a proactive style. Those methods embed invisible tags into images. Then, according to the extracted tag, they determine whether a suspicious image has been forged. However, these methods cannot pinpoint the tampered region in a forgery image.

To overcome these issues, we propose a proactive image authentication method based on deep learning semi-fragile watermarks. Our method can provide accurate and generalized tampering detection performance, not limited to a specific forgery or manipulation type. It can pinpoint the tampered pixels rather than only identify whether an image is a forgery.

The pipeline of our method involves converting a secret image into an invisible watermark, which we embed pixel-by-pixel into a cover image to create a watermarked image (known as the container image). The container image remains perceptually identical to the original cover image, allowing us to replace the original cover and use the container image in scenarios with a risk of malicious tampering. Any manipulation of the container image will inevitably affect the same region of the embedded watermark, causing damage to the secret image in that area. Consequently, we can precisely locate the tampered region by comparing the decoded secret image from the container image to the original secret image.

The designed framework consists of three modules to achieve the above function: a hiding network, an attack module and a revealing network. We adopt a cover-agnostic style hiding network that generates watermarks according to different secret images, independent of cover images. This design allows us to directly add the watermark to an arbitrary cover image to construct the container image, significantly improving our method's generalization. Besides, by minimizing the perceptual differences between cover and container images, the hiding network learns to encode secret images as invisible watermarks with remarkable imperceptibility.

In addition, to make our watermark fragile to malicious tampering approaches but robust to conventional image post-processing operations, we introduce an attack module in the training process. It consists of horizontally combined distortion and tamper layers that simulate tampering and post-processing manipulations. By applying these manipulations to container images, the attack module can strengthen the semi-fragility of our watermark. The attack module is only employed for training with hiding and revealing networks and is not included in our method's inference step.

The revealing network aims to recover the secret image from the container image to locate the tampered regions within it. We thus train the revealing network using the masked secret image with the same tampered regions in the processed container image as the label. Consequently, when tampering occurs within the container image, the revealing net focuses on restoring only the areas that remain untouched by tampering operations rather than attempting to reconstruct the entire secret image. The term "untouched area" refers to regions of the container image that have not been altered by tampering. This mechanism allows us to locate the tampered region accurately by comparing the original and recovered secret images.

Experiment results demonstrate that our designed scheme can achieve an average detection AUC of nearly 0.95 across a wide range of image manipulations. It turns the open-world image manipulation detection problem into a trivial watermark retrieval

Table 5.1: Summary of notations in this paper.

Notation	Description
x_{secret}	Secret image: the image to be hidden, serving as the source of a watermark.
x_{cover}	Cover image: the image to hide the secret image, also the image we want to protect from tampering.
$x_{container}$	Container image: the image with x_{secret} embedded, which is exposed to potential tampering methods.
$x_{retrieved}$	Retrieved secret: recovered secret image from container, comparing it with x_{secret} can identify the tampered area.

task, allowing for greater tamper detection accuracy. Our contributions are summarized as follows:

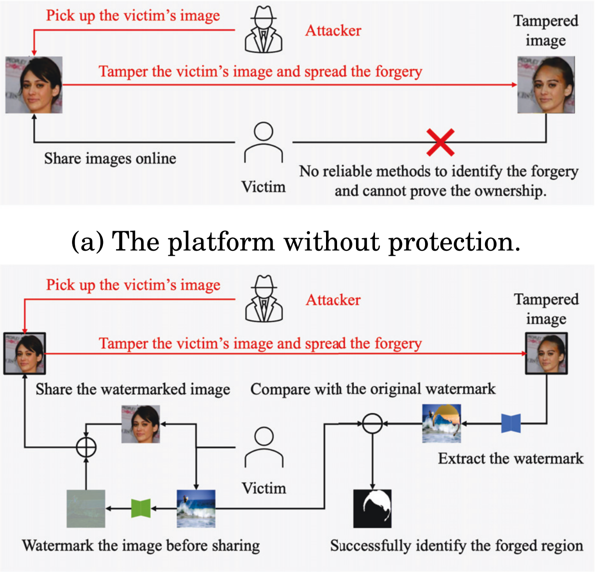
- We develop a novel deep learning-based semi-fragile image watermarking framework, which can serve as a proactive defence against malicious tampering.
- Our watermark achieves a balanced trade-off between detection performance and imperceptibility, so there is no influence on watermarked images' real-world usage.
- We have comprehensively evaluated the proposed method, analysing its performance across various aspects. We not only compared our method with SOTA detection methods on multiple datasets to detect various types of tampering to reflect its detection capability, but we also conducted a deep analysis of its watermark and assessed its performance in terms of robustness and security. Our experiment can serve as a template for similar research in the future.

5.3 Methodology

This section explains how to implement the proposed method. Given an image of size $W \times H$, the goal of our method is not only to determine if the image has been tampered with but also to locate the altered regions. The main notations used in the rest of the paper are listed in Table 5.1.

5.3.1 Motivation and Threat Model

We begin by elucidating the motivation and threat model of our method. Fig. 5.1 presents the common threat model to image-sharing platforms like Facebook or Instagram. At-


Pick up the victim's image
Attacker
Tampered image
Tamper the victim's image and spread the forgery
Share images online
Victim
No reliable methods to identify the forgery and cannot prove the ownership.
Pick up the victim's image
Attacker
Tampered image
Tamper the victim's image and spread the forgery
Share the watermarked image
Compare with the original watermark
Extract the watermark
Victim
Watermark the image before sharing
Successfully identify the forged region

(a) The platform without protection.

(b) The platform with our method's protection.

Figure 5.1: One potential application scenario of our method is protecting users' images on some public platforms, such as Instagram. Our method can give users a reliable approach to verifying the pixel-level authenticity of their images.

tackers in this model aim to tamper images and spread the forgery to produce reputation losses for the victim or obtain benefits from the forgery. We assume they have the same access rights as the victim, enabling them to obtain the victim's posted images and share the tampered images on the same platform.

In this scenario, the victim shares their images without any reliable precautionary measure. Once the image is uploaded online, there is no mechanism to ensure its authenticity and integrity, rendering it vulnerable to malicious tampering attacks. The attackers can easily pick the victim's photos, manipulate them, and release the tampered results while falsely claiming them to be authentic. Such misinformation can lead to severe reputational damage for the victim and raise security and privacy concerns. Worse still, it is hard for the victim to disclose the forgery since there are no reliable third part identification methods.

To address the above problems, we design a solution for proactively protecting the authenticity of images. We transform the secret image into a semi-fragile and invisible watermark and then embed it into the target image. This watermark is designed to be fragile to malicious manipulations or tampering, while simultaneously remaining robust to benign image-processing operations such as compression, scaling and colour adjustment. In this way, our methodology enables the identification of tampered regions

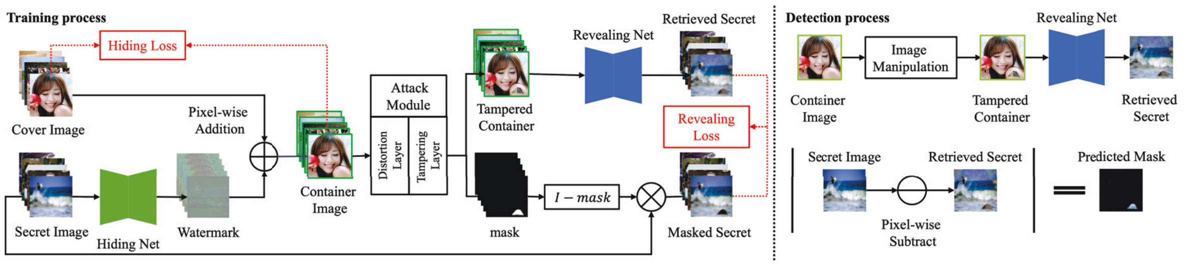


Figure 5.2: Overview of the proposed framework. The black arrows refer to data flows, and the red dashed lines show the loss flows. Our framework comprises three modules: hiding network, attack module and revealing network. The hiding network generates a watermark from the secret image, which is embedded in the cover image to produce the container image. The attack module manipulates the container image to create a processed image and corresponding mask. Finally, the revealing network retrieves the secret image from the processed container and determines the tampered region by comparing it with the original secret image. As illustrated in the detection process (right side), we can thus pinpoint the tampered region (predicted mask) by pixel-to-pixel comparing the retrieved and original secret images.

or pixels by comparing the recovered secret image from the container against the original secret image. This mechanism equips image owners with a reliable means of proactively protecting their images' authenticity and integrity before sharing them online.

Additionally, our approach distinguishes itself from passive detection methods by utilizing the embedded watermark as detection clues instead of artifacts left by tampering operations. This feature makes our method agnostic to the evolution of image manipulation techniques. So, it has reliable performance when detecting unknown and novel tampering methods.

The bottom panel of Fig. 5.1b illustrates how to deploy our method in the real world. Prior to sharing images online, users can use our method to embed the personalized, human-invisible watermark into their images. These watermarked images are virtually indistinguishable from the originals, thereby having a negligible impact on their visual quality and utility. Once the watermarked image has been tampered with maliciously, users can employ our method to verify the tampered region and declare the forgery.

5.3.2 Network architecture

As outlined in the motivation section, the objective of our networks is to convert a secret image as semi-fragile and invisible watermark, which is fragile to malicious tampering but withstand conventional image processing.

To this end, our networks comprise three distinct modules, as depicted in Fig. 5.2: (1) Hiding network: This module transfers the secret image as the human-invisible watermark. (2) Attack module: The attack module perturbs the container image that co-adapts with the training of hiding and revealing networks to strengthen our watermark's robustness and improve our method's tamper detection accuracy. (3) Revealing networks: This module recovers the secret image from the container and allows it to be compared with the original secret for identifying the tampered regions.

Hiding net. Previous learning-based image hiding techniques [155, 167] utilize the frequency discrepancy between the cover image and watermark to achieve effective hiding that is invisible to humans. Drawing inspiration from these methods, we adopt the same U-Net style architecture with full-convolution layers as the backbone for our hiding network. This architecture has performed excellently in extracting the secret image's representative features and encoding them into the high-frequency (HF) domain to generate the corresponding watermark. Unlike nature images, which mainly consist of low-frequency (LF) information, the HF watermark is invisible to human observers.

Consequently, embedding it in the cover image will not produce virtually noticeable alternations in the corresponding result, i.e., the container image in our context. Besides, the pixel-wise addition establishes the simple but effective one-to-one corresponding relationship between the watermark and the container image pixels. As a result, pixel changes in the container image directly affect the corresponding pixel in the embedded watermark, which will be further exhibited in the recovered secret image. The full-convolutional architecture also enables us to adjust the number of layers of the hiding network to produce the watermark with the same resolution size and the number of channels to fit different cover images.

Moreover, ensuring that the embedded watermark is semantically independent of the container image is crucial in achieving practical pixel-level tamper detection. An adversary may exploit the semantic information in the container image to hide tampering traces by modifying the container image according to its content. However, it is not possible to hide the variation caused by the tampering process in the embedded watermark when the watermark is semantically independent of the container image. Therefore, we adopt the cover-agnostic framework proposed by [155], which enables us to generate the watermark only with secret image input without any information from the cover image. This approach allows our hiding network to embed any secret image into any cover image without re-training. As a result, even when advanced tampering operations can generate visually realistic outputs, the tampered region remains detectable from the secret image, enabling practical pixel-level tampering detection.

Attack module. Our watermark should be robust against various image distortions and be sensitive to tampering operations. The former refers to the damage to images produced in their usage scenarios, such as Gaussian blurring or JPEG compression, affecting the whole image. In contrast, the latter involves intentional manipulation that only affects parts of the image. Based on this difference, we design an attack module consisting of a horizontally combined distortion layer and tampering layer, which applies global and local manipulation on container images.

The distortion layer is inspired by previous works such as [155, 167] and is designed to apply various distortions on container images. By co-adapting the training of our networks with relevant distortions, we can improve their robustness against these distortions.

To improve the tamper detection accuracy, we also design a tamper layer that imitates tampering operations on the container images. Before inputting the container image into the revealing net, we randomly select a region of the container image and modify the pixel values in this region to simulate pixel variation caused by tampering operations. In subsequent training, we mark the modified region as the mask and use it as the ground-truth label.

The rationale of the tampering layer is that it can generate a tampered container image and corresponding mask, allowing us to introduce simple tampering operations into our network training procedures. By enforcing the retrieved secret image with the same tampered region as the container image, the hiding network learns to embed the secret image sensitive to the container image's pixel variation. Similarly, the revealing network learns to extract the secret image with the same variation region based on changes in the container image rather than simply reconstructing the entire secret image. We can thus pinpoint the tampered region in the container image by comparing the original secret image with the retrieved secret image.

The distortion and tampering layers compose our attack module. Like other deep hiding methods, we add it between hiding and revealing networks. This attack module is only employed during training and is not included in our method's inference step. A detailed analysis of the attack module can be found in Sec.IV.B.

Revealing net. As explained in the preceding section, the hidden watermarks and cover images typically occupy different frequency domains within container images, with the watermark mainly residing in the HF and the original cover image mainly in the LF. From the perspective of the embedded watermark, the information of the cover image can be perceived as a frequency disturbance. The revealing network thus aims to retrieve the watermark under this disturbance and recover it as the secret image.

To this end, we employ a convolutional framework with residual connect as the backbone of our revealing network, which functions well when the inputs and outputs are distinct [38]. We jointly train the revealing network with the hiding network to pay more attention to the high-frequency spectrum of the embedded watermark. Such a design can significantly limit the impact of the disturbance of the cover image on the watermark retrieval and secret image recovery, resulting in superior performance in both concealing and revealing.

In addition, different from the conventional deep hiding methods aiming to reconstruct the secret image from the container image as high-fidelity as possible, the objective of our revealing process is to recover the secret image with the same tampered region as in the container image. Therefore, we use the tampering layer masks to mask the secret image during training as the recovery label for the revealing network instead of the original secret image.

5.4 Network Architecture

We illustrate the detailed structure of our networks in Fig. 5.3. Our method consists of two U-Net style pure convolutional networks, the Hiding network and the Revealing network, respectively.

The left side of Fig. 5.3 depicts the architecture of the hiding network, which is symmetric with down- and up-sampling blocks. The features extracted by the down-sampling blocks are passed on to the up-sampling blocks for further processing. Specifically, the down-sampling block consists of a 4 × 4 *Conv2d* layer with a stride of 2, a *LeakyReLU* layer with a negative sloop of 0.1, and a *BatchNorm* layer. The cover image is firstly down-sampled to a tensor with a size of 1024 × 2 × 2, which is then up-sampled to obtain the watermark. The up-sampling block includes a 4 × 4 *ConvTransposed2d* with a stride of 2, a *ReLU* layer, and a *BatchNorm* layer. At the end of the up-sampling process, the sigmoid function projects the outputs pixel value to -1 to 1 as the subsequent embedding process's watermark. This architecture enables the up-sampling blocks to share the features extracted by the down-sampling blocks, thereby preserving the input information for generating the output.

The revealing network's architecture illustrated on the right side of Fig 5.3 comprises three parts: the down-sampling, the residue, and the up-sampling. In the down-sampling

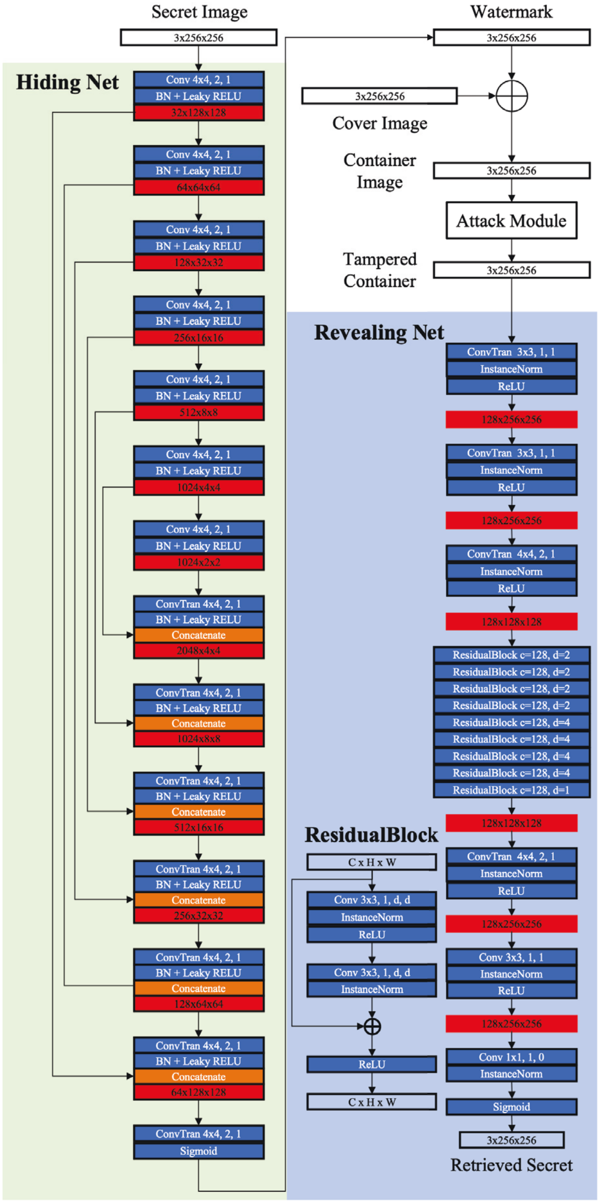


Figure 5.3: Network Architecture. *Conv k,s,p* represents a Convolutional Layer with kernel size k, stride s and padding p. *ConvTran k,s,p* represents a Transposed Convolutional Layer with kernel size k, stride s and padding p. All *Leaky ReLUs* have $\alpha = 0.1$.

part, the unit block consists of a 3 × 3 *Conv2d* layer, a *BatchNorm* layer, and a *ReLU* layer. Specifically, the stride is set to two in the last block of the down-sampling part to enlarge the receptive field. In the residue part, nine residue blocks, comprising 18 convolution layers, generate residual features. In the up-sampling part, the residual features are up-sampled to recover the secret image. The unit block here is similar to the down-sampling part, with only differences in the number of input and output channels.

For up-sampling, the first block's *Conv2d* layer has a stride and kernel size of 2 and 4×4, respectively. Finally, the sigmoid function is applied to output the final recovered secret image. This architecture is based on CEILNet, which performs well when its output differs significantly from its input, making it suitable for retrieving the secret image from the container image.

5.4.1 Loss functions

To minimize the difference between the cover and container image, and enforce the retrieved secret to reflect the container's tampered pixels accurately, we adopt the following losses:

Hiding secret loss. We define a simple but effective pixel similarity loss function between the cover and container image to optimize the hiding network to achieve indistinguishable image hiding:

$$\mathscr{L}_{Hiding} = \|x_{container} - x_{cover}\|_2,$$

where the $\mathscr{L}_{Hiding}$ adopts l_2-norm.

Revealing secret loss. Given the tampered container image from the attack module, we train the revealing network using the same loss function as the Hiding loss. Nevertheless, to ensure that the retrieved secret image accurately reflects the tampered area in the container image, we do not simply reconstruct the entire secret image. Instead, we use the masked secret image as the ground truth label to train the revealing network to recover the secret image with the same tampered region applied to the container image. This function is defined as follows:

$$\mathscr{L}_{Revealing} = \|x_{secret} \times M - x_{retrieved}\|_2,$$

where $\mathscr{L}_{Revealing}$ also adopts l_2 norm, and M is the absolute residual between the mask from the tampering layers and identity matrix, i.e., $|I - mask|$.

Total loss function. The total loss function $\mathscr{L}_{Total}$ is a weighted sum of $\mathscr{L}_{Hiding}$ and $\mathscr{L}_{Revealing}$, as follows:

$$\mathscr{L}_{Total} = \lambda_H \mathscr{L}_{Hiding} + \lambda_R \mathscr{L}_{Revealing},$$

where λ_H and λ_R are weights used to balance different loss terms.

It should be noted that all losses can adopt the l_1 norm or a combination of different norms. However, our validation results have shown that the choice of the norm does not significantly impact our method's performance. Therefore, we have adopted the l_2 norm for uniformity and convenience.

5.5 Experiments

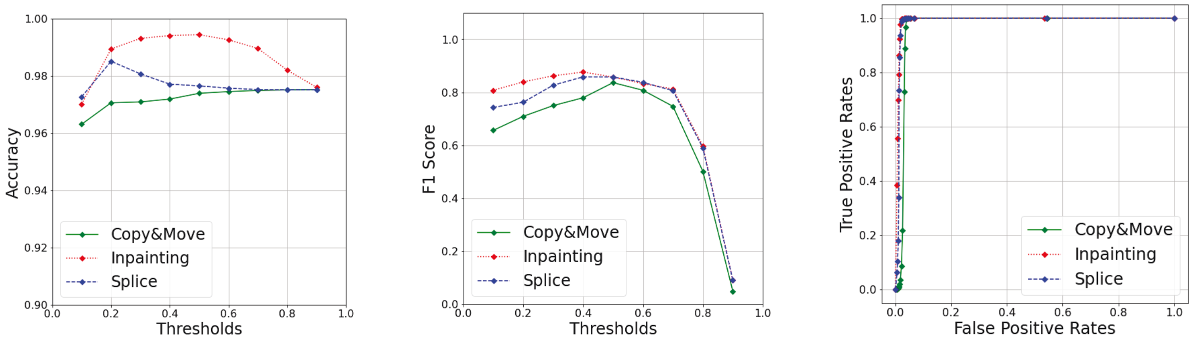
We conduct extensive experiments to evaluate our method's performance from the following aspects: (1) impact of different components and decision thresholds; (2) effectiveness in image tamper detection; (3) robustness against conventional image post-processing; (4) imperceptibility of the embedded watermark; (5) security under threats and countermeasures. Furthermore, we implemented detailed analyses of the watermark that we embedded.

5.5.1 Experimental Setup

Dataset. We train our networks on the widely used face image dataset FFHQ [71] and test its performance on other datasets. The gap between training and test datasets can validate our method's generalization. Note that choosing the FFHQ dataset as the training set is not based on considering performance differences.

Evaluation Metrics. To evaluate the performance of our method, we employed several evaluation metrics for each aspect of our experiments. Peak-Signal-to-Noise Ratio **(PSNR)** and Structural Similarity Index Measure **(SSIM)** are used to measure the similarity between the watermarked (container) images and original (cover) images to reflect the imperceptibility of the watermark and fidelity of the watermarked (container) images. We also calculate the Area Under the receiver operating characteristic curve **(AUC)** as the primary evaluation metric to reflect the image tamper detection performance.

Implementation Details. Our method is implemented in PyTorch and trained on an NVIDIA Tesla K80 GPU. The image size in the experiment is set to 256x256. We use an Adam optimizer whose learning rate periodically decays from 10e-4 to 10e-7. We set the two weights in the combined loss as $\lambda_H = 1$ and $\lambda_R = 0.75$, according to the model performance on a held-out validation set from FFHQ.



(a) Accuracy (b) F1 score (c) ROC

Figure 5.4: The tampering detection Accuracy, F1 Score and ROC of our method under different decision thresholds.

5.5.2 Ablation Study

We first perform ablation studies to evaluate the impact of distinct components and decision thresholds on the performance of our method.

Impact of decision thresholds. Our method classifies the genuine and forged pixels based on the disparity between the recovered and original secret images. Thus, the decision threshold for our method is the threshold value that determines whether two pixels are dissimilar. To assess the impact of different decision thresholds on our method, we embed the same secret image into 1k Celeba-HQ [70] images using our method. Next, we apply three typical tampering operations, i.e., copy-move, inpainting and splice, to generate the corresponding test sets. Then, we employ our method to detect the tampered regions in each set while varying the decision threshold from 0.1 to 0.9 with a step size of 0.1. We calculate the corresponding accuracy and F1 score according to the detection results and plot the ROC. We do not calculate AUC values here because it is a threshold-irrelevant metric that cannot reflect the fluctuations in detection performance with decision thresholds varying. The results are presented in Fig. 5.4.

The results in Fig. 5.4a show high detection accuracy across different sets, even under extremely low or high thresholds. This abnormal phenomenon is due to the imbalanced data problem between forgery and authentic pixels in each image, which rendering accuracy cannot distinguish the performance of our method under varying decision thresholds. Nevertheless, according to the trend of F1 Score curves, we can find that our method exhibits superior detection ability when the threshold is below 0.7, but beyond this threshold, the performance decreases significantly. This phenomenon occurs because a high threshold ignores correctly identified tampered pixels, leading to poor detection

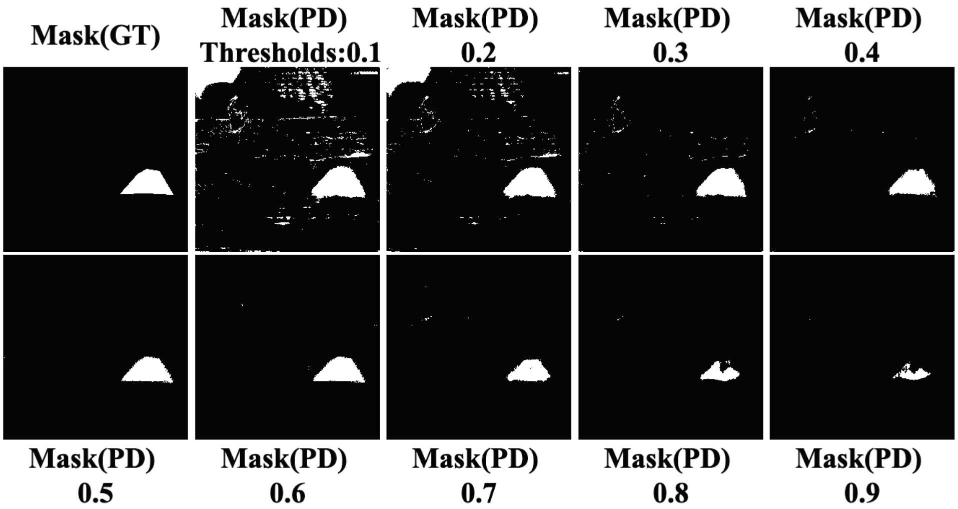


Figure 5.5: Samples of pixel-level detection results of our method in varied decision thresholds. Mask (GT) represents the ground truth of tampered masks. Mask (PD) represents the predicted tampered results, and the values under Mask (PD) are the corresponding decision threshold.

results. The ROC presented in Fig. 5.4c proves the outstanding detection capability of our method. All three curves are close to the top left, and the corresponding AUCs are higher than 0.95.

Visualized predicted masks in Fig. 5.5 also verify the inference in the above quantitative evaluation. The predicted tampered areas are smaller than the ground truth when decision thresholds are higher than 0.6, while predicted tampered areas with lower thresholds tend to produce false alarms on authentic pixels. We can safely conclude that our method can effectively classify genuine and forgery pixels under the appropriate decision thresholds. Unless otherwise specified, we will set the decision threshold to 0.5 in the following experiments.

Impact of different components. To assess the influence of individual components of our method, we evaluate the designed scheme's performance under various configurations. All configurations are trained on the complete FFHQ dataset and tested on a random sample of 1k CelebaHQ images. For the sake of simplicity, we compute the average pixel-level detection AUC values of Copy-Move, Inpainting and Splice. Additionally, we employ three commonly used image post-processing methods, namely Blur (with kernel sigma 1.0), JPEG (with compression factor 70) and Crop (random crop 70% image), to attack the container images of each configuration. We calculate the detection AUC on these images to gauge each configuration's robustness against post-processing.

Table 5.2: Ablation studies for different model designs.

Secret	Attack module			Effectiveness & Robustness				Stealthiness	
Image	None	Distortion	Tampering	AUC ↑	AUC(Blur1.0) ↑	AUC(JPEG70) ↑	AUC(RC0.7) ↑	SSIM ↑	PSNR ↑
RGB	✓			0.9368	0.5658	0.6111	0.9360	**0.95**	**39.65**
		✓		0.9397	0.7410	0.9095	0.9285	0.93	38.01
			✓	0.9755	0.6092	0.6992	0.9745	0.93	37.79
		✓	✓	**0.9793**	**0.7340**	**0.9650**	**0.9750**	0.94	38.05
Gray		✓	✓	0.9689	0.7213	0.9342	0.9815	0.94	38.01
QR Code		✓	✓	0.9745	0.7460	0.9676	0.9745	0.93	37.89
Noise		✓	✓	0.9780	0.7130	0.9456	0.9667	0.94	38.10

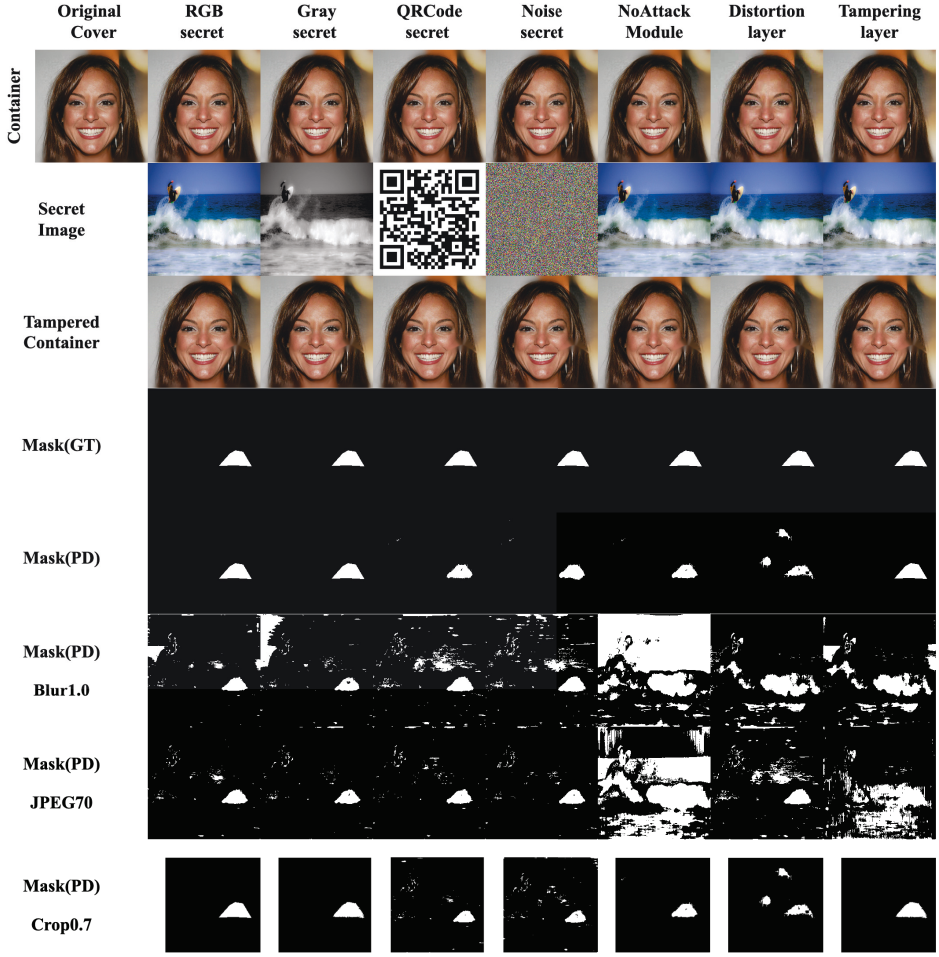


Figure 5.6: Samples of pixel-level manipulation detection results of our method in varied setups. Mask (PD) with the name of post-processing methods or without name means the predicated tampered mask from the corresponding post-processed or original tampered container, respectively.

The results are summarized in Table 5.2, while a more detailed analysis is presented below.

(1) Influence of different secret image formats. As stated in the methodology section, the secret image used in our method is independent of the cover images. Thus, we investigate the impact of different formats of secret images on our method's performance. We select images from four common types of images as secret images, including RGB, grayscale, QR code, and Gaussian noise. Different secret images were embedded into the same 1k CelebaHQ images using the same networks and trained models, resulting in four distinct groups. Next, we evaluated the performance of each group to determine if there were any differences in the method's performance.

The evaluation results for the various secret image formats are presented in the last four rows of Table 5.2 and 2nd to 5th columns in Fig. 5.6. The metrics values for each group are similar, with a maximum difference of 10%, while the qualitative results are almost identical. These findings suggest that the choice of different secret image formats does not significantly impact the performance of our proposed method. This characteristic is particularly practical because users of our method can freely choose any image as their secret image making it challenging for potential adversaries to obtain users' secret images falsely, thus increasing the security of our method. In the following experiment, we will use RGB images as the secret image for simplicity and uniformity.

(2) Influence of attack module. Finally, we use different combinations of components in the attack module to reveal their effect (1st to 3rd rows in Table 5.2 and last three columns in Fig. 5.6).

Without equipping the distortion layers, networks achieve high detection AUCs on no-distorted container images in the first and fourth rows. However, when applying image post-processing methods to the container images, the networks experience a significant performance drop. In contrast, networks with distortion layers in their attack module achieve higher detection AUCs on distorted container images, as shown in the second and third rows. It can also be observed in Fig. 5.6 that the networks without the distortion layer will produce more errors in the predicted mask when the container images are distorted. These performance fluctuations indicate that the distortion layers can improve our method's robustness against image distortion attacks.

Furthermore, by comparing the models equipping the tampering layers (third and fourth rows) to those without tampering layers (first and second rows), we can observe that tampering layers can further improve the method's tampering detection accuracy, especially on the non-distorted container images. As illustrated in Fig. 5.6, networks with tampering layers produce more accurate predictions of tampered regions, with fewer false negatives than networks without attack modules. Conversely, networks without

Table 5.3: The averaged pixel-level tamper detection **AUC** ↑ results from different methods.

Detection methods	**CASIA 1**			**CASIA 2**			**Columbia**			**MS-COCO**		
	CM	**IP**	**SP**	**CM**	**IP**	**SP**	**CM**	**IP**	**SP**	**CM**	**IP**	**SP**
MVSS-Net	0.9118	0.9376	0.9063	0.9118	0.9243	0.9168	0.9350	0.9104	0.9261	0.9050	0.9199	0.9108
HP-FCN	0.5995	0.6837	0.5906	0.5885	0.6734	0.5909	0.5580	0.6816	0.5845	0.4085	0.6912	0.4050
ManTraNet	0.9064	0.8987	0.8862	0.8774	0.8966	0.8788	0.9161	0.9205	0.8851	0.9411	0.9049	0.8890
Ours	**0.9793**	**0.9930**	**0.9959**	**0.9778**	**0.9924**	**0.9959**	**0.9719**	**0.9936**	**0.9906**	**0.9663**	**0.9991**	**0.9915**

*Note: the abbreviation **CM** represents CopyMove, **IP** represents Inpainting, **SP** represents Splice and **DF** represents DeepFake.

attack modules tend to underestimate the extent of tampered areas, resulting in reduced detection performance. Networks equipped only with distortion layers exhibit a high rate of false alarms, likely due to the perturbations introduced by the distortion layers.

Overall, the full framework with both distortion and tampering layers in the attack module achieved the best performance on almost all detection metrics, justifying the necessity of all components in the attack module.

5.5.3 Effectiveness

In this section, we compare our method with other tamper detection methods to reflect our method's advantages and disadvantages in tamper detection. Three published methods are selected for comparison with our method: HP-CNN [75], MVSS-Net [32], and ManTraNet [145], for they all have publicly available pre-trained models and source codes.

Pixel-Level comparison. We first detect pixel-level tampering on CASIA 1 [33], CASIA 2 [33], Columbia [103], and 10k randomly selected MS-COCO [82] datasets. We embedded the watermark into these images and applied different tampering operations. Then, we used our method to detect tampering in these images, while other methods were directly employed for detecting forgery on non-watermarked images. As shown in Table 5.3, our method outperforms other methods on all datasets by a significant margin. It exhibits nearly perfect detection performance on some sets, e.g., IP in CASIA 1 or SP in MS-COCO. Additionally, all set results indicate that our method has a more stable detection capability across all different datasets, while others have extremely higher performance fluctuations.

Image-Level comparison. We then conduct the image-level detection comparison. Our method can achieve image-level tamper detection by setting a threshold that the image will be identified as a forgery when the number of pixels within this image

Table 5.4: The averaged image-level tamper detection **AUC** ↑ results from different methods.

Detection methods	**CASIA 1**			**CASIA 2**			**Columbia**			**MS-COCO**		
	CM	**IP**	**SP**	**CM**	**IP**	**SP**	**CM**	**IP**	**SP**	**CM**	**IP**	**SP**
MVSS-Net	0.8010	0.7945	0.8134	0.8201	0.7847	0.8130	0.8210	0.8014	0.8203	0.7900	0.7887	0.8108
HP-FCN	0.5159	0.5837	0.5100	0.5534	0.5974	0.6100	0.5050	0.5860	0.5145	0.5085	0.5901	0.4950
ManTraNet	0.7850	0.7780	0.7915	0.8074	0.8090	0.7880	0.8105	0.8130	0.7905	0.8330	0.8158	0.7930
Ours	**0.9139**	**0.9353**	**0.9291**	**0.9210**	**0.9213**	**0.9312**	**0.9152**	**0.9301**	**0.9445**	**0.9236**	**0.9080**	**0.9210**

*Note: the abbreviation **CM** represents CopyMove, **IP** represents Inpainting, **SP** represents Splice and **DF** represents DeepFake.

Table 5.5: The averaged pixel-level tamper detection **FPR** ↓ from different methods.

Detection methods	**CopyMove**	**Inpainting**	**Splice**
MVSS-Net	0.1921	0.0526	0.077
HP-FCN	0.0419	0.0413	0.0388
ManTraNet	0.1444	0.0419	0.0988
Ours	**0.0144**	**0.0176**	**0.0124**

recognized as forgeries is over this threshold. Specifically, we fix to tamper the 30% area of each image, and an image will be classified as tampered when 10% of its pixels are recognized as forgery pixels. We increase this decision threshold from 10% to 30% with step 2% to calculate corresponding detection AUC values. Besides, the same number of authentic images is mixed with tampered images in each set to test different methods' real/fake classification ability.

Table 5.4 summarizes the performance of distinct models. Despite the degradation of AUC scores across all methods compared to pixel-level detection results, our method again emerges as the top performer. These findings demonstrate our method's superior detection performance at pixel and image levels. Notably, the MVSS-Net and ManTraNet exhibit more severe degradation in detection performance than others. We attribute this phenomenon to false-alarm problems of these methods on the authentic pixels/images. To verify this hypothesis, we calculate each method's pixel-level detection results' averaged False-Positive Rate (**FPR**). According to Table 5.5, the baseline methods' FPR is significantly higher than ours. These results suggest that other methods produce more false alarms on the authentic pixels than ours, leading to unreliable detection results that limit their practical reliability. On the contrary, our method can provide more reliable detection effects on both tampered and authentic pixels/images.

Computation efficiency comparison. We also measure the detection efficiency in terms of Frames Per Second (FPS). Tested on NVIDIA Tesla K80 GPU, HP-FCN, MVSS-

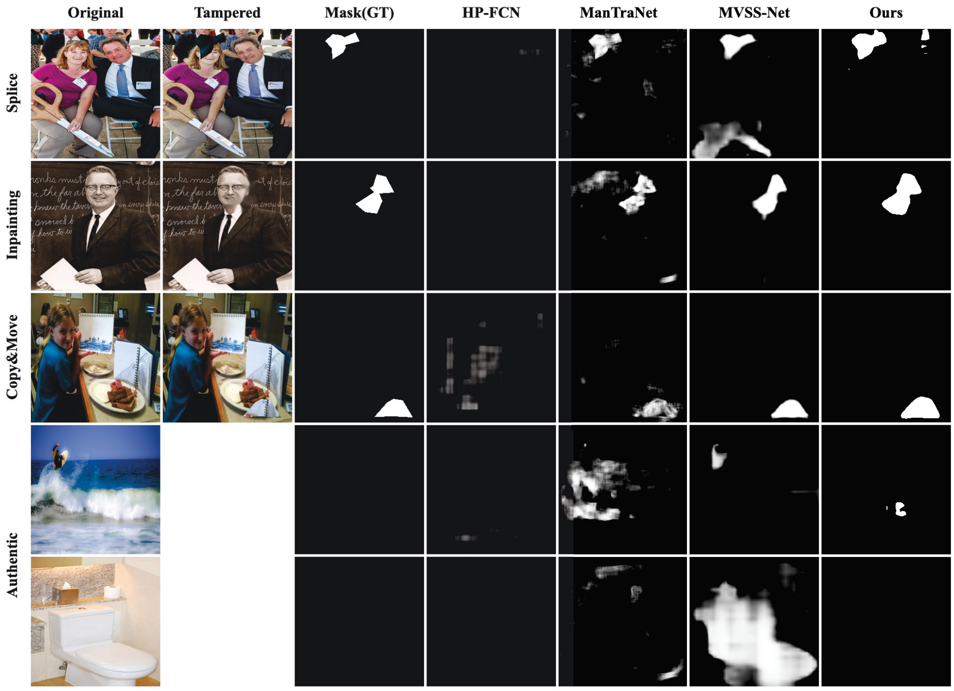


Figure 5.7: Examples of mask prediction from different detection methods. Compared with other methods, ours can accurately pinpoint the tampered pixels and produce significantly fewer false alarms on the authentic pixels.

Net, and ManTra-Net run at FPS of 3.91, 19.84 and 2.73, respectively. The watermarking stage of our method runs at 9.88 FPS, while the detection stage runs at 36.71 FPS. Our method's primary time consumption is mainly associated with embedding the watermark into images. In practice, this stage can be executed before tamper detection. Therefore, only considering the detection stage, the FPS of our method is sufficient for real-time application.

Qualitative comparison. Finally, we visualize each method's detection results in Fig. 5.7. Consistent with qualitative results, the figure illustrates that our method can accurately locate the tampered regions with significantly fewer false positive pixels, especially when detecting authentic images. While other methods can also pinpoint the tampered regions, they are accompanied by some false alarm results in both tampered and authentic images. Therefore, our method's detection results are more reliable.

5.5.4 Robustness

In real life, one may disguise a tampered image with additional post-processing to evade detection or apply various post-processing on the container image for different

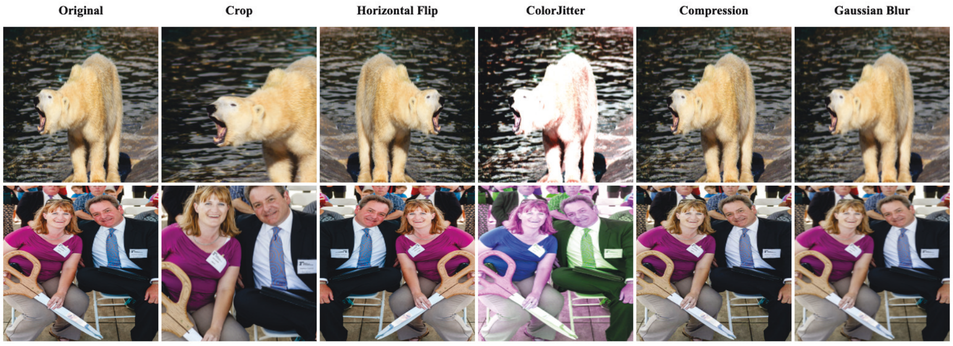


Figure 5.8: Samples of each post-processing method are adopted in the robustness evaluation. Note that the Crop's samples are resized in this figure for better visualization.

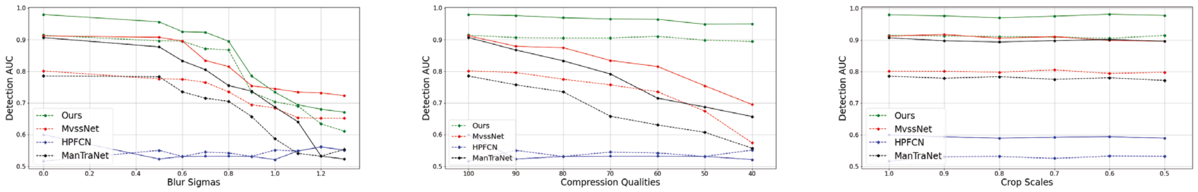
applications. The detection methods should remain effective against these processing techniques. Here, we consider five typical post-processing techniques and also present how we apply them in the following experiment:

- **Gaussian blur** the images with kernel standard deviation ranging from 0.5 to 1.3 with a step size of 0.1.
- **JPEG compress** all images with quality factors ranging from 40 to 100 with a step size of 10.
- **Crop** the image to a smaller size ranging from 100% to 50% with a step size of 10%.
- **Horizontally flip** all images.
- **Colour adjusts** all images' brightness, contrast, saturation and hue randomly.

We present samples of different post-processing operations' results in Fig. 5.8. These samples provide a visual representation of the impact of various post-processing techniques on the images. We apply the above post-processing to the tampered images and then employ different methods to detect the tampered regions. As shown in Table 5.6 and Fig. 5.9, our method is immune to crop, JPEG compression and horizontal flip, whereas susceptible to Gaussian blur and colour adjustment. Especially for the Gaussian blur, the overall performance almost drops linearly. The main reason is that these processing methods would modify the images' pixel value, further distorting our embedded watermark so that they will significantly impact our method's detection capability. However, as the robustness evaluation results revealed, our method still provides comparatively

Table 5.6: The averaged image/pixel-level detection **AUC** ↑ results from different methods under colour adjustment and horizontal flipping.

Image Manipulations	Detection methods			
	MVss-Net	HP-FCN	ManTraNet	Ours
ColorAdjust	**0.77/0.88**	0.53/0.55	0.73/0.86	0.78/0.84
HorizontallyFlip	0.79/0.91	0.54/0.58	0.78/0.89	**0.92/0.98**



(a) GaussianBlur

(b) JPEGCompress

(c) Crop

Figure 5.9: The averaged image/pixel-level tamper detection **AUC** ↑ from different methods against Gaussian blur, JPEG compress and crop. The solid line indicates the pixel-level detection results, while the dashed line represents the image-level detection results. Our method is robust against crop and JPEG compress but sensitive to Gaussian blur. However, our method still achieves better detection performance than other methods under these post-processing distortions.

better detection performance against other baselines, demonstrating that it can provide reliable detection performance under real-world distortions.

5.5.5 Imperceptibility

In this section, we evaluate the visual quality of the container image to reflect the stealthiness and imperceptibility of our watermark, thereby ensuring that it does not compromise the utility of the watermarked image. As there is no available deep watermark-based tamper detection method for comparison, we compare our method with SOTA invisible watermarking methods. To ensure a fair comparison, we only select methods with a payload capacity higher than 24 bits per pixel (bpp), which can hide an entire image into a cover image. As a result, HiNet [67], DDH [143], and UDH [155] are selected as the learning-based baselines, and 4bit-LSB [128] which is a traditional watermarking technique, is also selected as a reference in our comparison.

To assess the imperceptibility of our method and all baselines, we use them to embed the same secret image into 1k randomly selected MS-COCO images to generate corresponding container image sets. We then calculate PSNR and SSIM between each method's container and cover images. The numerical results are summarized in Table 5.7.

Table 5.7: The averaged similarity between original and watermarked images from different image hiding methods.

Quality metrics	**Deep Hiding Techniques**			**LSB**	**Ours**
	HiNet	**DDH**	**UDH**		
SSIM ↑	**0.97**	0.71	0.84	0.78	0.94
PSNR ↑	**40.17**	26.24	29.77	28.05	38.05

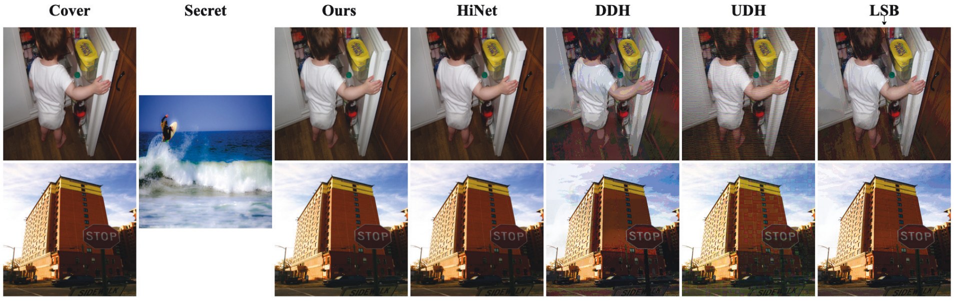


Figure 5.10: Qualitative comparison between our method and SOTA hiding techniques' watermarked images. Our outputs are perceptually identical to the original, while the results of LSB have slight colour distortion, and DDH and UDH introduce noticeable artifacts in their outputs.

From the table, we observe that HiNet achieves exceptionally high values on both metrics, while our method ranks second with slightly inferior performance (SSIM 0.04 lower and PSNR 2.0 dB lower), which still outperforms other methods by a large margin.

While our method may not have the best quantitative imperceptibility performance, the qualitative results in Fig. 5.10 demonstrate that the container images of our method are sufficiently naturalistic for human observers. Both HiNet and our method generate high visual quality container images with more visual-realistic and accurately preserve the original's hue and light in corresponding watermarked images. In contrast, UDH and DDH introduce apparent artifacts in their watermarked images. LSB outputs have obvious colour distortions. The high visual quality of the container image can also verify our watermark's imperceptibility.

Besides evaluating the similarity between the container and cover images, we employ an open-source steganalysis tool called StegExpose to measure the anti-steganalysis ability of each method's watermark. Fig. 5.11 shows the ROC curve of each method. We can see that the StegExpose detection accuracy on learning-based methods is quite close to the random guess, indicating their watermarked images are highly imperceptible to fool the Statistical steganalysis tool.

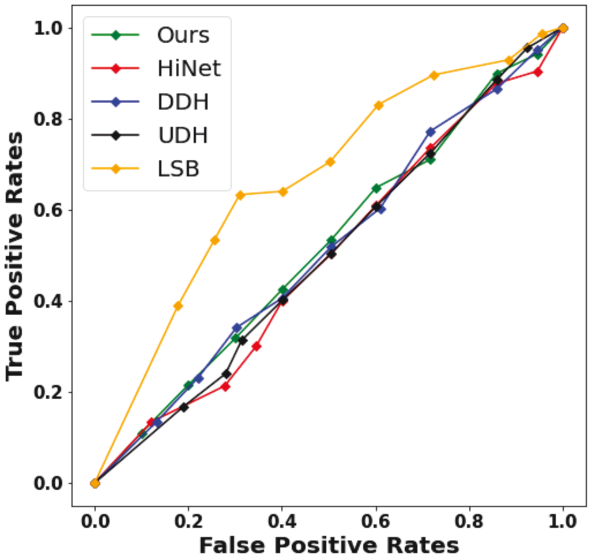


Figure 5.11: ROC curves of StegExpose for detecting different image-hiding methods. The detection accuracy on all learning-based methods is close to random guess, demonstrating the highly imperceptible to fool the Statical steganalysis tool.

In summary, qualitative and quantitative results demonstrate that our method can generate high visual-quality container images and imperceptible watermarks, ensuring that our watermarked image can still be used normally in real-world scenarios.

5.5.6 Security

This section investigates potential security risks that the attackers may exploit to compromise the effectiveness of the proposed method.

The security of the secret image. Our initial investigation focuses on determining whether adversaries can acquire users' secret images by analyzing information from container or cover images.

The first line of defence against this threat is the imperceptibility of our watermark. As demonstrated by the results of our imperceptibility experiments, adversaries cannot distinguish between the container and cover images using quantitative, qualitative or steganalysis-based analyses. Consequently, it would be challenging for adversaries to locate images containing secret information, let alone obtain them.

In our method's application scenario, users share only the container images on platforms with risks of malicious tampering while keeping their cover images private. As a result, adversaries cannot access original cover images without users' permission. To fully validate the security of our method, we investigate whether the secret image

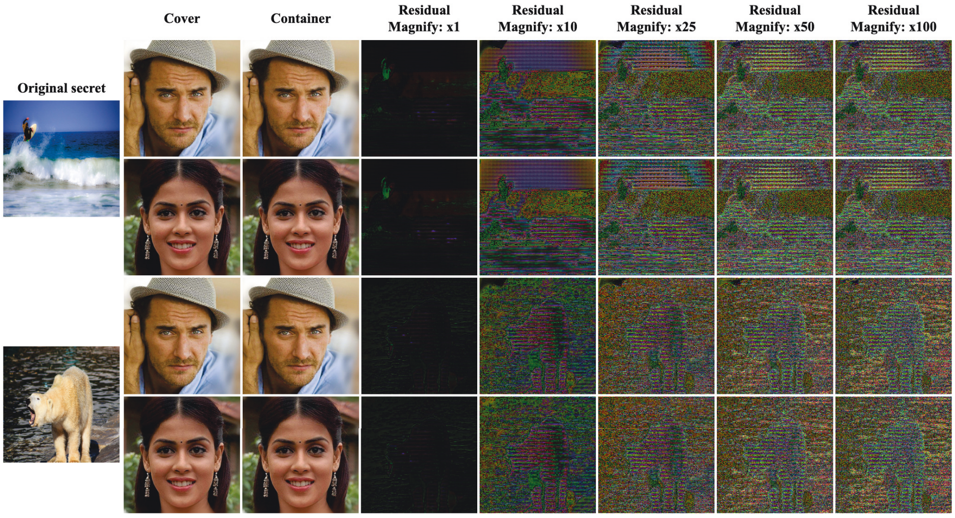


Figure 5.12: Residuals between our method's container and cover images. As expected, the residuals without magnification are nearly equal to blank. Magnifying the residuals may reveal the embedded secret images' outlines, but still very ambiguous.

is safe when the cover images leak. As illustrated in Fig. 5.12, the residuals between the container and cover images are exceedingly thin, and even with 50 or 100 times magnification, they remain ambiguous. Consequently, adversaries can not obtain the secret images according to the container and cover images.

Risk of the secret image leakage. We then consider the scenario in which users' secret images are compromised while the pre-trained models remain secure. Adversaries will use their trained models to embed these secret images into their tampered images to conceal the tampered area and evade our detection.

To simulate this attack, we trained two different groups of models on an identical dataset to represent the user and adversary, respectively. We first employ user networks to embed the secret image into 1k randomly selected CelebaHQ images, followed by tampering operations. We then use adversary networks to embed the same secret image into these tampered images to simulate the adversary's cloaking action. Finally, we utilize the user's networks to detect tampered and cloaked images, testing whether our method is still effective. Table 5.8 summarizes the detection FPR, TPR, Precision and F1 Score, while Fig 5.13 provides visual representations of the results.

Our findings indicate that this attack reduces the detection performance of our method, particularly by generating more false negative tampered pixels. However, the TPR suggests that more than 65% of tampered areas can still be successfully localized, demonstrating that adversaries cannot wholly evade detection. Furthermore, the quali-

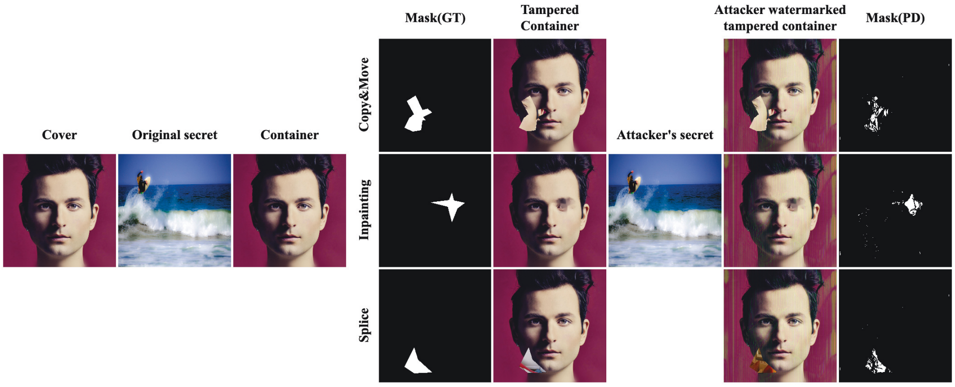


Figure 5.13: Samples of the experiment result when the secret image is leaked. Embedding the same secret image into the tampered container images will lead our method to miss some tampered pixels in detection results, but the major tampered regions are still accurately localized. Additionally, this cloak action will also produce apparent artifacts in the outputs.

Table 5.8: Pixel-level detection results when the secret image leaks. w/o represents without, and w represents with.

	TPR ↑	FPR ↓	Precision ↑	F1 Score ↑
CopyMove				
w/o cloak	0.7035	0.0140	0.9490	0.8080
w cloak	0.6114	0.0162	0.9441	0.7422
Inpainting				
w/o cloak	0.7271	0.0195	0.9273	0.8151
w cloak	0.6114	0.0162	0.9441	0.7422
Splice				
w/o cloak	0.7209	0.0131	0.9518	0.8204
w cloak	0.6114	0.0162	0.9441	0.7422

tative results presented in Fig. 5.13 validate the above inference that, despite reduced detection accuracy, the primary tampered regions in the container images are still accurately identified. We attribute this to the inability of different network groups to replace the watermarked secret images in each container image perfectly.

Additionally, we observe that cloaked images exhibit apparent artifacts, likely due to the overflow of embedding the extra secret image into already watermarked images. This phenomenon renders the attack meaningless again, as cloaked images are easy to identify and forbidden.

Risk of the pre-trained model. Finally, we analyze the situation where users' pre-trained models are compromised while their secret images remain safe. Adversaries

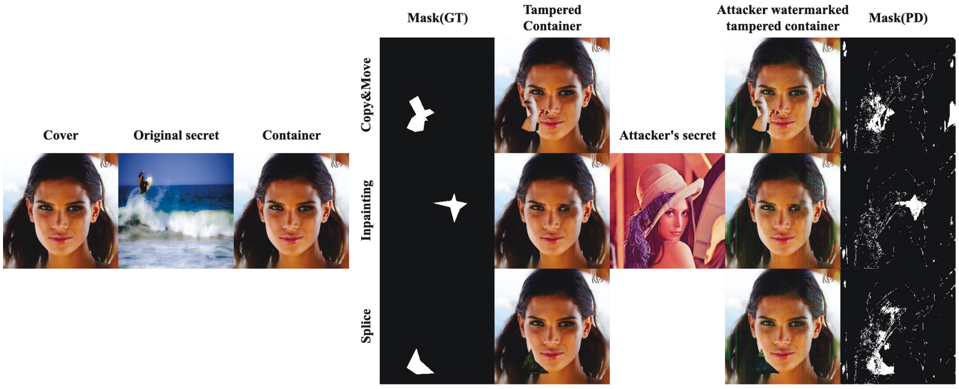


Figure 5.14: Samples of the experiment result when the pre-trained model is leaked. Embedding the different secret images using the same models into the tampered containers will make our method produce more false alarms when detecting authentic pixels. However, the tampered areas are still accurately identified, and apparent artifacts appear in the cloaked images.

Table 5.9: Pixel-level detection results when pre-trained models leaked.

	TPR ↑	FPR ↓	Precision ↑	F1 Score ↑
CopyMove				
w/o cloak	0.7094	0.0144	0.9473	0.8113
w cloak	0.6750	0.0236	0.9141	0.7766
Splice				
w/o cloak	0.7232	0.0182	0.9323	0.8146
w cloak	0.6760	0.0292	0.8924	0.7693
Splice				
w/o cloak	0.7258	0.0138	0.9489	0.8225
w cloak	0.6356	0.0220	0.9218	0.7524

will use the same networks to embed different secret images into their tampered images to conceal the tampered regions and evade our detection.

To simulate this attack, we first employ our networks to embed a secret image (secret a) into the cover images and then perform tamper operations on these images. Next, we use the same networks to embed another secret image (secret b) into these tampered images to imitate the adversaries' concealing process. Finally, we conduct tamper detection on these images based on the secret image a.

The results are provided in Table 5.9 and Fig 5.14. Similar to the previous, this attack also declined our method's detection performance, but as evidenced by FPR values, it will lead to more false alarms on the authentic pixels in detection results rather than falsely negating tampered pixels. The results in Fig 5.14 are consistent with quantitative results, as the tampered regions can be accurately identified, but together with some false alarms on the authentic pixels.

We infer that this phenomenon occurs because the newly embedded secret image destroys the original watermark in the container image. Therefore, when comparing the original secret image with the secret image recovered from this cloaked container, the destroyed area in the watermark will result in false alarms. Nevertheless, as the results show, the tampered regions can still be identified.

In summary, we have validated our method's security, which can keep users' secret images safe in the practical application scenario and sustain detection performance even if some credential information is compromised. However, considering the performance degradation when credentials are leaked, it is still essential for users of our method to keep their secret images and pre-trained models private.

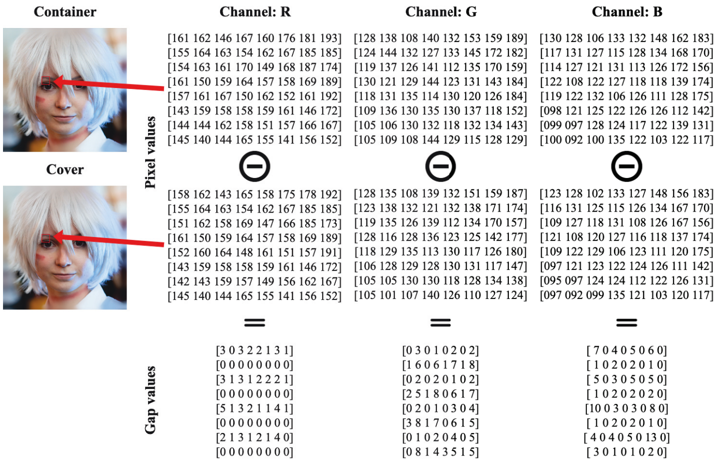
5.5.7 Analysis

According to the above examinations, it is clear that our method is capable of reliable pixel and image-level tamper detection. However, it is still insufficient to understand our watermark and the watermarked image. In this section, we will implement an investigation to analyse how our method works.

Pixel value Analysis. We begin our investigation by analyzing the pixel values of our watermark. Our method incorporates the watermark into the cover image by directly pixel-wise addition to generate the corresponding container image. Consequently, this process inevitably introduces pixel differences between the cover and container images. However, the Hiding secret loss of our method enforces the container to be identical to the cover image. This design ensures that the Hiding network produces the watermark with minimal pixel values to include the secret image's information.

As a result, there is no noticeable distinction between cover and container images, even though their pixel values are slightly different. To support our hypothesis, we provide Fig 5.15 as evidence. We can observe that although the pixel values of the cover and container images within the red box differ in all three RGB channels, this difference is very slight. Therefore, the changes in pixel values resulting from the watermark would not produce distinguishable perceptual alterations in the container image.

Frequency analysis. We then compute the averaged Azimuthal Integral (**AI**) [85] values of different images that appear in our method to explore their frequency properties. In brief, Azimuthal Integral [34] computes the radial integral over the 2D discrete



Channel: R	Channel: G	Channel: B
[161 162 146 167 160 176 181 193]	[128 138 108 140 132 153 159 189]	[130 128 106 133 132 148 162 183]
[155 164 163 154 162 167 185 185]	[124 144 132 127 133 145 172 182]	[117 131 127 115 128 134 168 170]
[154 163 161 170 149 168 187 174]	[119 137 126 141 112 135 170 159]	[114 127 121 131 113 126 172 156]
[161 150 159 164 157 158 169 189]	[130 121 129 144 123 131 143 184]	[122 108 122 127 118 118 139 174]
[157 161 167 150 162 152 161 192]	[118 131 135 114 130 120 126 184]	[119 122 132 106 126 111 128 175]
[143 159 158 158 159 161 146 172]	[109 136 130 135 130 137 118 152]	[098 121 125 122 126 126 112 142]
[144 144 162 158 151 157 166 167]	[105 106 130 132 118 132 134 143]	[099 097 128 124 117 122 139 131]
[145 140 144 165 155 141 156 152]	[105 109 108 144 129 115 128 129]	[100 092 100 135 122 103 122 117]

[158 162 143 165 158 175 178 192]	[128 135 108 139 132 151 159 187]	[123 128 102 133 127 148 156 183]
[155 164 163 154 162 167 185 185]	[123 138 132 121 132 138 171 174]	[116 131 125 115 126 134 167 170]
[151 162 158 169 147 166 185 173]	[119 135 126 139 112 134 170 157]	[109 127 118 131 108 126 167 156]
[161 150 159 164 157 158 169 189]	[128 116 128 136 123 125 142 177]	[121 108 120 127 116 118 137 174]
[152 160 164 148 161 151 157 191]	[118 129 135 113 130 117 126 180]	[109 122 129 106 123 111 120 175]
[143 159 158 158 159 161 146 172]	[106 128 129 128 130 131 117 147]	[097 121 123 122 124 126 111 142]
[142 143 159 157 149 156 162 167]	[105 105 130 130 118 128 134 138]	[095 097 124 124 112 122 126 131]
[145 140 144 165 155 141 156 152]	[105 101 107 140 126 110 127 124]	[097 092 099 135 121 103 120 117]

[3 0 3 2 2 1 3 1]	[0 3 0 1 0 2 0 2]	[7 0 4 0 5 0 6 0]
[0 0 0 0 0 0 0 0]	[1 6 0 6 1 7 1 8]	[1 0 2 0 2 0 1 0]
[3 1 3 1 2 2 2 1]	[0 2 0 2 0 1 0 2]	[5 0 3 0 5 0 5 0]
[0 0 0 0 0 0 0 0]	[2 5 1 8 0 6 1 7]	[1 0 2 0 2 0 2 0]
[5 1 3 2 1 1 4 1]	[0 2 0 1 0 3 0 4]	[10 0 3 0 3 0 8 0]
[0 0 0 0 0 0 0 0]	[3 8 1 7 0 6 1 5]	[1 0 2 0 2 0 1 0]
[2 1 3 1 2 1 4 0]	[0 1 0 2 0 4 0 5]	[4 0 4 0 5 0 13 0]
[0 0 0 0 0 0 0 0]	[0 8 1 4 3 5 1 5]	[3 0 1 0 1 0 2 0]

Figure 5.15: The partial pixel values of cover and container images and the values gap between them. Since adding with watermark in it, the container image's pixel values are different from the cover image. Nevertheless, the values gap is tiny, so there is no perceptual difference between cover and container images, indicating our method's perfect concealing ability.

Fourier Transform spectrum along the spatial frequency. Given a square image I of size $M \times N (M = N)$, the spectral representation is computed from the discrete Fourier Transform (DCT)

$$\begin{aligned} \text{DCT}(I)(k,l) &= \sum_{m=1}^{M} \sum_{n=1}^{N} e^{-2\pi i \cdot \frac{jk}{M}} e^{-2\pi i \cdot \frac{jl}{N}} \cdot I(m,n), \\ \text{for} \quad k &= 1, \ldots, M, \quad l = 1, \ldots, N, \end{aligned} \tag{5.1}$$

via Azimuthal Integration over radial frequencies ϕ

$$\begin{aligned} \text{AI}(\omega_k) &= \int_0^{2\pi} \|\text{DCT}(I)(\omega_k \cdot \cos(\phi), \omega_k \cdot \sin(\phi))\|^2 d\phi \\ \text{for} \quad k &= 1, \ldots, M/2. \end{aligned} \tag{5.2}$$

As depicted in Fig. 5.16, the 1D Azimuthal Integral power spectrum reflects the relative intensity of the 2D spectrum at a certain frequency spatial coordinate, where the intensity begins with the highest value at the lowest frequency and decreases as the frequency increases. The outcomes are plotted in Fig. 5.17 and Fig. 5.18.

From Fig. 5.17, it can be observed that the frequency distribution of the container image in the low-frequency domain almost aligns with that of the cover images. Since the high-frequency contents in images are generally imperceptible to human observers, it explains why container images are perceptually indistinguishable from their corresponding

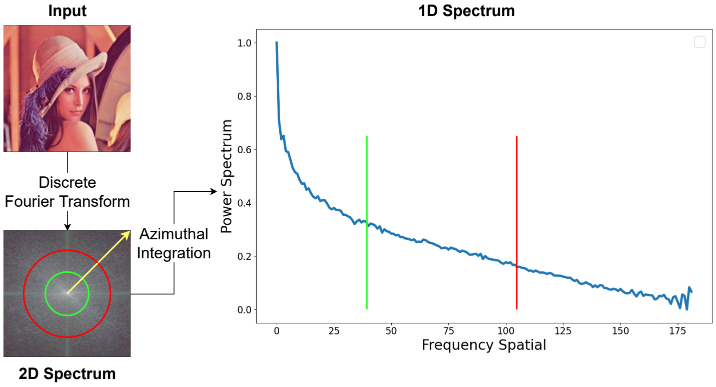


Figure 5.16: Azimuthal Integral. The input image is transformed into a 2D spectrum using discrete Fourier transform. Then, the integration is conducted from the inside of the 2D spectrum to the outside, along the yellow arrow circularly. Eventually, the 1D spectrum is derived, where the red and green lines represent the sum of the pixel values on the red and green circles.

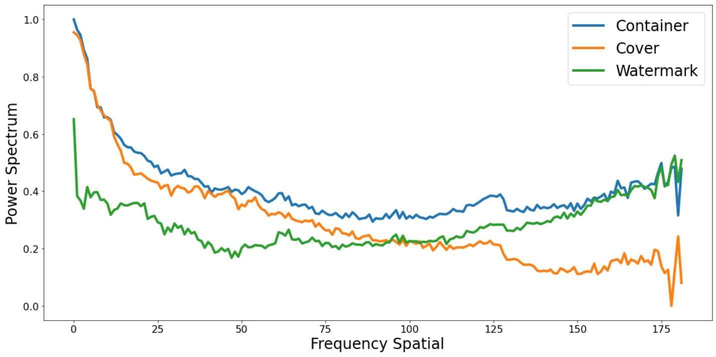


Figure 5.17: The averaged Azimuthal Integral values of different cover, container and watermark images. The watermarks mainly consist of a high-frequency spectrum. The container images' low-frequency distribution is almost aligned with that of the cover images while having a higher high-frequency spectrum. So, it would be straightforward to conclude that this high-frequency difference is due to the embedded watermarks on containers.

cover images. Similarly, the watermark images, which are almost perceptually invisible, mainly consist of high-frequency distribution with a significantly weak low-frequency spectrum. Consequently, the container images embedded with these watermark images will only exhibit some high-frequency distortion compared to the cover images but no noticeable artifacts. The secret images' frequency distributions in Fig. 5.18 are almost perfectly aligned in most frequency domains but only slightly differ in the high-frequency domain. These imperfections could be attributed to the imperfection reconstruction of our method, but based on the previous experimental results, we consider them acceptable.

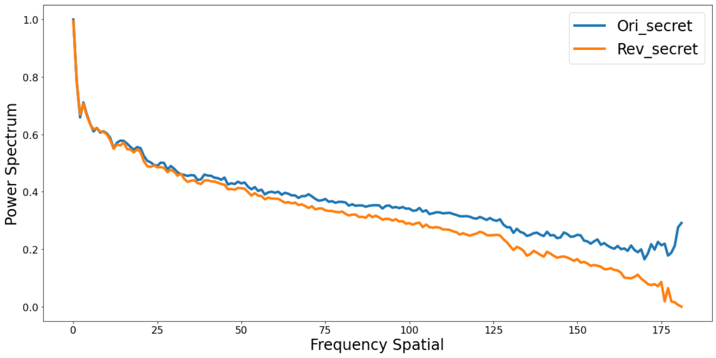


Figure 5.18: The frequency distribution of the original and recovered secret images. They are almost perfectly aligned in most frequency domains but only differ in the high-frequency.

5.6 Conclusion and Discussion

The task of image tamper detection has become increasingly challenging due to the constant evolution of image editing and synthesizing techniques. Confirming the authenticity of images by identifying artifacts left from the manipulation process has become more complex than ever before. To address these challenges, we propose a proactive method to protect the authenticity of images by embedding a semi-fragile and invisible watermark into each target image. This watermark serves as an indicator to verify the authenticity of the image's pixels.

The experiment results have demonstrated that our method performs exceptionally well across all evaluation metrics. However, it should be noted that our method follows a distinct pipeline from other current image tamper detection methods, which requires additional steps. Given the perfect detection performance of our method, we believe this overhead is acceptable. At present, the proposed method is best suited for protecting critical information. For example, public celebrities can use our method to add personalized watermarks to their images to prevent malicious image forgery and tampering. Only images with their watermarks can be considered authoritative, while images without the corresponding watermarks will be assumed to come from unofficial channels. They can also verify the tampered regions and declare forgery to reduce reputation loss.

Overall, this work represents a new direction for proactively fighting against malicious tampering operations on image data.

Chapter 5's in-depth analysis of media authenticity protection sets the stage for Chapter 6, which tackles the critical aspect of media authorship proof. This chapter extends the conversation from ensuring media authenticity to establishing and protecting the rights of content creators in the digital domain. It introduces novel techniques for authorship attribution in image data, highlighting the importance of robust and reliable methods for asserting and defending intellectual property in an increasingly digital and interconnected world. Specifically, we propose a novel method that leverages the semantic information in images to boost the robustness of watermarks, enabling reliable authorship attribution. It includes a novel semantic image-hiding network, explores semantic features for hidden information, and achieves a balanced trade-off between capacity, imperceptibility, and robustness with comprehensive performance evaluations.

CHAPTER 6

MEDIA AUTHORSHIP PROOF

6.1 Preface

Chapter 6 addresses a significant aspect of our work that underpins the fight against plagiarism and copyright infringement: authorship proof in image data. By identifying the limitations of conventional authorship-proof methods, we set the stage for introducing our unique strategy, specifically designed to enhance the robustness of authorship attribution.

Our work contributes to image steganography, the practice of imperceptibly embedding a secret image within a cover image. While recent advancements have concentrated predominantly on refining output quality and payload capacity, our research identifies the often-overlooked aspect of robustness as a crucial focus. Ignoring this dimension exposes these methods to potential distortions during data pre-processing and transmission stages.

To counter this vulnerability, we introduce ROSIN (Robust Semantic Image-hiding Network), a robust semantic image-hiding network beyond merely embedding the secret into a particular domain of the entire cover image. Our novel methodology first semantically disentangles the cover image into attribute and identity features and then hides the secret image in the identity feature. This feature, characterized by its valuable value and geometric invariance properties, is an ideal candidate for secret image concealment.

The results of our extensive experiments, covered in this chapter, validate the effec-

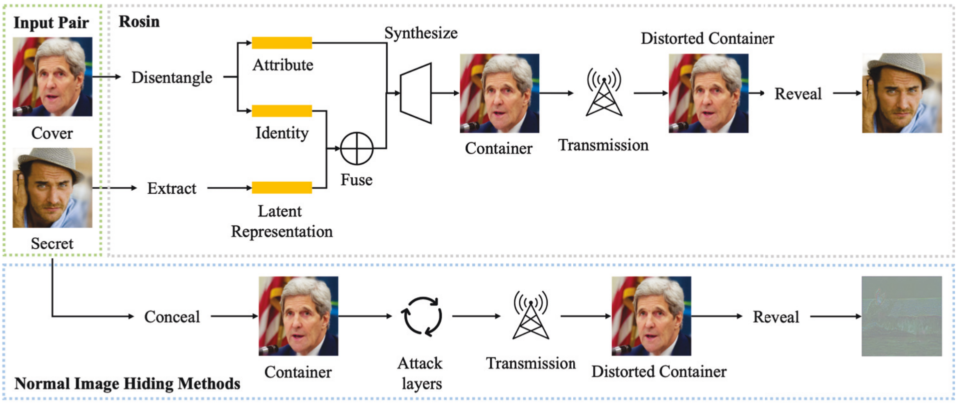


Figure 6.1: The structure of ROSIN vs. conventional learning-based image steganography framework. Previous works enhance their model's robustness by inserting diverse distortions at the training stage. On the contrary, our Rosin semantically disentangles the input cover image and exploits the stability identity feature to embed the secret image to achieve high robustness.

tiveness of our design. When juxtaposed with state-of-the-art techniques, we demonstrate that ROSIN achieves superior robustness while maintaining comparable imperceptibility and capacity. The ability of our proposed solution to leverage an image's semantic features for enhanced robustness represents the first of its kind in the field of image steganography, paving the way for promising real-world applications.

Throughout this chapter, we delve deeper into the implementation of our method, discussing in detail the outcomes of our experiments and the potential implications of our work on authorship attribution. This exploration underscores the critical role of our work in combating plagiarism and copyright infringement, providing new insights into a novel approach to authorship proof in image data.

6.2 Introduction

Image steganography, a well-studied subset of steganography, involves concealing a secret image within a cover image to create a container image. The container image must appear virtually identical to the cover image to facilitate the transmission of the secret image while remaining undetectable to unauthorized viewers. Only the intended recipient possesses the means to extract the embedded secret image from the container image. Unlike other steganography techniques, such as bit-level message hiding [8], image steganography prioritizes high imperceptibility and hiding capacity over perfect decoding of embedded secrets.

Traditional techniques conceal the secret image with different strategies, including the least significant bits (LSB) [15, 68, 117], pixel value differencing (PVD) [161], histogram shifting [116], discrete Fourier transform (DFT) [3], discrete wavelet transform (DWT) [6, 133], etc. While these methods have achieved promising fidelity, their embedding capacity is limited (typically around 0.2-4 bits per pixel (bpp)), and they often lack sufficient robustness. Recently, learning-based image steganography methods have alleviated the above limitations. These novel methods demonstrated impressive performance in visual quality and hiding payloads, compared to traditional approaches. However, some of them neglect the crucial aspect of robustness. Even slight distortions introduced to the container image during data pre-processing and transmission will severely degrade the quality of the revealed secret image at the receiver's end. Given the prevalence of such distortions in the dissemination of container images, image steganography without a robustness guarantee becomes impractical.

Existing learning-based methods attempt to enhance robustness by training their networks to mitigate various distortions that occur between the concealing and revealing processes. These distortions, sometimes referred to as attack layers [156], can involve pre-defined image post-processing manipulations or simulated perturbations generated by neural networks. Typically, these methods employ architectures similar to the one shown in Fig. 6.1.

Albeit effective, this approach faces several challenges. Firstly, the distortions must be differentiable to enable joint training in the concealing and revealing processes. However, many practical distortions, such as JPEG compression, are non-differentiable, making it difficult to incorporate them into neural network training. Secondly, in most cases, real-life distortions often differ from the assumed distortions used in training. Selecting a diverse and well-balanced set of distortions for training becomes nearly impossible, resulting in poor performance when facing novel distortions. Lastly, this design makes it challenging to balance robustness with other metrics. Training with distortions inherently conflicts with the goal of improving imperceptibility, potentially leading to a decrease in image perception quality.

To address these issues, we propose a **Ro**bust **S**emantic **I**mage Hiding **N**etwork, which is called **ROSIN**. As illustrated in Fig. 6.1, ROSIN differs from previous methods by semantically disentangling the input cover image and utilizing the stable identity feature to achieve high robustness while concealing the secret image. Unlike prior methods, our approach does not require prior knowledge of image distortions during training.

In the ROSIN framework, the cover image is disentangled into attributes and identity features. We employ an encoder network to extract the secret image's latent representation. A learnable network combines the identity feature and latent representation to form a fused feature vector. The training process ensures that the vector preserves the essential information from both the identity feature and the latent representation. Following the fusion of the identity feature and the latent representation, a synthesizing network is employed to generate the container image by utilizing the fused feature and the original attribute feature of the cover image. Although visually identical to the original cover image, the container image secretly carries the embedded secret image. This allows for the transmission of the secret image without raising any suspicion. At the receiver end, the same identity encoder used in the cover image disentanglement is utilized to extract the identity feature of the container image. Dedicated networks are then employed to separate the embedded secret representation, which is subsequently used to reconstruct the secret image. Thanks to the stability of the identity feature, the embedded secret image can be revealed even under a wide variety of image distortions that may occur during transmission.

To the best of our knowledge, this work represents the first attempt to leverage semantic feature stability for image steganography. We conduct extensive experiments to evaluate our method, and the results consistently demonstrate the superior performance of ROSIN compared to existing baselines across various aspects. In summary, our main contributions are listed as follows:

- We develop a novel semantic image-hiding network that exhibits high robustness under diverse distortions, enhancing image steganography practicability.

- Our work explores semantic features' redundancy and validates their suitability for carrying hidden information in the context of image steganography.

- We demonstrate the identity feature is invariant to conventional image distortions, further emphasizing its robustness.

- Our method achieves a balanced trade-off between capacity, imperceptibility and robustness, addressing the limitations of previous approaches.

- We conduct comprehensive evaluations of our proposed method, analyzing its performance across multiple dimensions to establish its effectiveness and superiority.

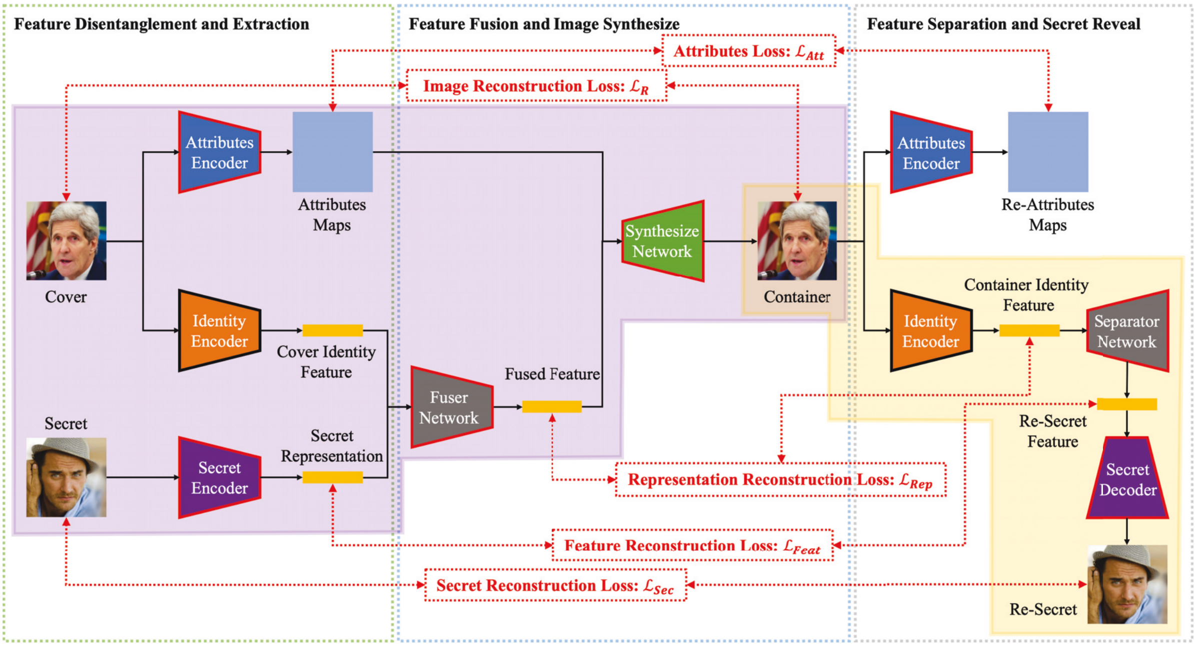

Feature Disentanglement and Extraction
Feature Fusion and Image Synthesize
Feature Separation and Secret Reveal
Attributes Loss: $\mathcal{L}_{Att}$
Image Reconstruction Loss: $\mathcal{L}_R$
Attributes Encoder
Attributes Maps
Cover
Identity Encoder
Cover Identity Feature
Secret
Secret Encoder
Secret Representation
Fuser Network
Fused Feature
Synthesize Network
Container
Attributes Encoder
Re-Attributes Maps
Container Identity Feature
Identity Encoder
Separator Network
Re-Secret Feature
Secret Decoder
Re-Secret
Representation Reconstruction Loss: $\mathcal{L}_{Rep}$
Feature Reconstruction Loss: $\mathcal{L}_{Feat}$
Secret Reconstruction Loss: $\mathcal{L}_{Sec}$

Figure 6.2: Overview of our proposed Rosin's framework. The black arrows refer to data flows, and the red dashed lines show the loss flows. Besides, the trapezoids with red solid lines represent a trainable network, while the black line trapezoid represents a frozen network. The input cover image is first disentangled into attribute and identity features, and the secret image is projected as a latent representation. Then, the identity feature and latent representation are combined to generate the fused feature by the Fuser network, which is believed to contain both the cover image's identity feature and the secret's semantic representation. A synthesis network next takes fused feature and original attribute maps to yield the container image. In the reveal process, the container image is fed to the same identity encoder used in cover image disentanglement to extract its identity feature, which is then input to the separator network to retrieve embedded secret representation and reconstruct the secret image.

6.3 Methodology

The primary target of Rosin is to design a general and robust framework for image steganography under diverse distortion, which is achieved via our novel semantic disentanglement-based image-hiding techniques. Fig. 6.2 gives an overview of Rosin's architecture. Our Rosin network embeds the secret representation to the disentangled cover identity feature, then uses the embedded identity feature with the original cover attributes features to synthesize a container image perceptually identical to the original cover. The secret representation fusion and concealing processes adopt adaptive mechanisms trained to reduce perturbation on the container image. When we need to reveal the hidden secret image, we can extract the container image's identity feature

Table 6.1: Summary of notations in this paper.

Notation	**Description**
x_{sec}	Secret image: the image to be hidden, and wants to be transmitted without notice.
x_{cov}	Cover image: the image to hide the secret image, also the image we used to camouflage the secret.
x_{con}	Container image: the image with x_{sec} embedded, exposed to various distortions in transmission.
$\hat{x}_{sec}$	Revealed secret: recovered secret image from x_{con}, reflecting whether the transmission succeeded.
$z_{id}(X_{con})$	Identity feature of the cover image, serving as the carrier of the secret image.
$z_{att}(X_{cov})$	Attribute feature of the cover image, reserve its spatial information.
$z_{rep}(X_{sec})$	Latent representation of the secret image, preserving the essential information for reconstruction.
$z^{w}_{id}(X_{cov})$	Feature incorporated identity and representation, used to synthesize the container image.
$z_{id}(X_{con})$	Identity feature of the container image, which is believed to contain the secret representation.
$z_{rep}(\hat{X}_{sec})$	Retrieved secret representation, which can be used to reconstruct the secret image.

and separate the embedded secret representation for reconstruction.

In summary, the functionalities of our framework can be divided into three major parts: (1). feature disentanglement and extraction; (2). feature fusion and image synthesis; (3). feature separation and secret reveal. In the following parts of this section, we provide a detailed explanation of the Rosin architecture, including its functionality, loss function and processing pipeline. The main notations used in the rest of the paper are listed in Table 6.1.

6.3.1 Feature Disentanglement and Extraction

Given input cover and secret images, the first step of our method is to disentangle the cover image into two independent representations, which are identity $z_{id}(X_{con})$ and attribute features $z_{att}(X_{cov})$, and extract the secret image's latent representation $z_{rep}(X_{sec})$. We designed three dedicated networks to do that, illustrated in Fig. 6.2, which are Attributes and Identity Encoders for the cover image disentanglement and

Representation Encoder for the secret image extraction.

Cover Images' Features Disentanglement: The identity feature in this context represents the high-level human bio-metric information neural networks use to distinguish one individual from another. It includes facial structure and landmarks, enabling the characterizing of a specific person with lesser intra-personal variations and larger interpersonal differences. Therefore, the identity feature will remain consistent in face image transmission even under image distortions or post-processing manipulations. Embedding secret information within the identity feature can thus leverage this inherent stability to its resilience against distortions.

Similar to most research for feature disentanglement works [106, 142], the identity encoder in our work employs the pre-trained face recognition network [31] as the backbone to extract the input image's last feature vector generated before the final fully-connected layer as identity feature. This network adopts additive angular margins in the well-established softmax loss function to maximize identity class separability, so its identity feature has a clear geometric interpretation and is highly discriminative. Specifically, the identity feature is a 512-dimension vector, which is formulated as $z_{id}(X) = Arc(X)$, where X denotes the input image and $Arc(\cdot)$ represents the face recognition network.

The attribute feature of the face image is defined as spatial information related to various aspects, such as pose, expression, background, etc. Depending on the level of detail, attributes can range from coarse (e.g., overall spatial outline) to fine (e.g., precise shape). Therefore, we adopt multi-level feature maps to preserve such details to represent the attributes. Specifically, we feed the input image into a U-Net style network and then use the feature maps generated from the U-Net decoder as attribute representations. The formal attributes representation is denoted as:

$$z_{att}(X) = \left\{z_{att}^{1}(X), z_{att}^{2}(X), \ldots, z_{att}^{n}(X)\right\},$$

where $z_{att}^{n}(X)$ represents the n-th level feature map from the U-Net decoder, and n is the number of feature levels.

We use the U-Net architecture from [77] as the attributes encoder for its ability to capture both high-level and low-level spatial information, making it suitable for preserving the diverse details associated with the attributes of the face image. It does not require extra annotations, in our work, as it extracts the attributes using self-supervised training, which is trained to keep the original Cover image X_{cov} and Container image X_{con} have the representation of the same attribute.

Secret Images' Representation Extraction: The identity feature of the cover image is a 512-dimensional vector, which is insufficient to directly accommodate the pixels of secret images. Moreover, attempting to embed pixels into this high-level latent space would result in value overflow and significant distortion on the identity feature [119]. To address this issue, we train an encoder to map the input secret image into a latent representation.

The secret encoder adopts Resnet-50 [52] as the backbone, as it performs excellently in extracting input image features. We use the last feature vector generated by the final fully-connected layer as the secret image's representation, which is also a 512-dimensional vector. By training the secret encoder with the decoder to minimize the difference between the input and output images, it learns to encode the secret image's essential feature while discarding the unnecessary details. This process allows the representation to effectively guide high-quality secret reconstruction, even with the reduced dimensionality. This process is denote as $z_{rep}(X_{sec}) = Enc(X_{sec})$.

6.3.2 Feature Fusion and Image Synthesize

The next step in our method involves concealing the secret representation within the identity feature and synthesizing the container image using the fused feature and the original attribute feature from the cover image. Since the container image must be indistinguishable from the cover image, our approach employs learnable mechanisms for more adaptive fusion and synthesis.

We designed a dedicated Fuse Network to conceal the secret representation into the identity feature. This Fuse Network adopts five full-connected layers, where the first four layers are followed by a LeakyReLU layer with a negative sloop of 0.2 and a BatchNorm layer. It inputs two 512-dimension vectors (the identity feature and secret representation) and then outputs a single 512-dimensional fused feature. Through training to synthesize the container image and reconstruct the secret image, the Fuse Network learns to adaptively combine the different representations, effectively generating a fused feature that preserves essential information from the identity feature and secret representation. This use of learnable fusion reduces distortion in the container image, as denoted by $z^{w}_{id}(X_{cov}) = Fuse(z_{id}(X_{cov}), z_{rep}(X_{sec}))$.

Subsequently, we integrate the fused feature $z^{w}_{id}(X_{cov})$ and the original attributes $z_{att}(X)$ to synthesize the container image. Previous studies [10, 102] revealed that simply concatenating identity and attributes to synthesize images will incur severe visual quality degradation and distortion. To overcome this problem and generate the high-fidelity container image, we employ a novel *Adaptively Attentional Denormalization* (AAD) [77] mechanism to accomplish feature integration.

The synthesize network incorporates multiple cascaded AAD Residual Blocks (ResBlk) to integrate the identity and attributes. Each AAD ResBlk comprises multiple AAD layers, which employ an attention mechanism with denormalization to dynamically adjust the participation of identity representation and attribute representation for synthesizing different regions. For instance, the identity will provide more importance on generating the facial area, which is most discriminative for distinguishing identities, while the attributes will focus more on the regions related to spatial features, such as skin colour and background.

We formally define the synthesize procedure as:

$$X_{con} = Gen\left(z_{id}^{w}(X_{cov}), z_{att}(X_{cov})\right),$$

where $Gen(\cdot)$ denote the synthesize network. This adaptive attentional mechanism allows for fine-grained control over the contribution of each feature representation, resulting in a more accurate and visually pleasing synthesis of the container image. More importantly, it can help perverse the secret image's representation into the container image's identity feature.

6.3.3 Feature separation and secret reveal

The final step of Rosin is to recover the embedded secret image from the container image. Since the secret representation is concealed in the identity feature, we initially adopt the same identity encoder to extract the container's identity feature, denoted as $z_{id}(X_{con}) = Arc(X_{con})$. Then, we use a Separator Network to extract the secret representation from the container's identity feature. This network also uses multiple full-connected blocks, which consist of a full-connected layer, a ReLu layer and a BatchNorm layer, as the backbone and outputs a 512-dimension revealed secret representation $z_{rep}(\hat{X}_{sec}) = Sep(z_{id}(X_{con}))$. After that, we input the revealed secret representation into the secret Decoder to reconstruct the secret image $\hat{X}_{sec} = Dec(z_{rep}(\hat{X}_{sec}))$.

The Decoder adopts the CEILNet-structure framework [38] that is believed to function well in image reconstruction. This network mainly comprises several upsampling blocks followed by a residue block to generate the output. Specifically, each up-sampling block consists of a 3x3 Conv2dtrans layer, an InstanceNorm layer and a LeakReLu layer. The final output image is obtained by convolving the last activation with a Tahn function.

During the training, we extract the attribute feature from the container image and compute the difference between this feature and the cover attribute feature. This difference is used to optimize the attribute encoder.

Additionally, to mitigate the issue of over-fitting, we employ a straightforward strategy at the training. Specifically, we ensure that the revealing process of Rosin can distinguish between the container image from the clean image, rather than simply producing a secret image regardless of the input. To achieve this, we divide the input to the revealing stage during training into two parts. One half consists of the clean images generated using the original cover identity and attribute features, without any embedded secret representation. The remaining inputs are container images containing the secret image. By doing so, the Separator and Decoder are forced to reconstruct the secret image when presented with a container image, while producing a null image, i.e., a pure black image when given a clean image. This simple yet effective strategy helps address the over-fitting problem in Rosin.

6.3.4 Loss Functions

We describe the loss functions for training Rosin using the same notations as in Fig. 6.2. No extra losses are required in our training procedure, and except for the identity encoder, all other networks are trainable.

Image Reconstruction Loss: To keep the container image resemble the original cover image and also mitigate the conflict with the secret image concealing, we define a perceptual similarities loss LPIPS [159] between the cover and container images rather than the common pixel-level $\mathscr{L}_2$ similarities:

$$\mathscr{L}_R = \left\| L(X) - L(\hat{X}) \right\|_2 ,$$

where $L(\cdot)$ represents the perceptual features extractor.

Attributes Loss: We also calculate the attributes features' $\mathscr{L}_2$ distance between the cover and container images to enforce attributes preservation:

$$\mathscr{L}_{Att} = \frac{1}{2} \sum_{k=1}^{n} \left\| z_{att}^{k}(X) - z_{att}^{k}(\hat{X}) \right\|_2^2 ,$$

where the n denotes the level of attributes.

Feature Preservation Loss: This loss function measures the cosine similarity between the fused feature and the extracted container identity. It expects to train Rosin to integrate the fused feature in the container image as its identity feature.

$$\mathscr{L}_{Feat} = 1 - CosineSimilarity(\hat{z}_{id}(X), Arc(\hat{X})),$$

where $Cos(\cdot)$ denotes the operation of cosine similarity.

Representation Reconstruction Loss: Additionally, we adopt a simple but effective $\mathscr{L}_2$ similarity loss between the original and revealed secret representations to regulate the secret representation reconstruction.

$$\mathscr{L}_{Rep} = \frac{1}{2}\sum_{k=1}^{n}\left\|z^k_{rep}(X_{sec}) - z^k_{rep}(\hat{X}_{sec})\right\|_2^2,$$

For the inputs from the clean image, we set their original secret representations as zeros vectors, which can help Separator to differentiate between the container and clean images.

Secret Reconstruction Loss: Finally, an $\mathscr{L}_2$ loss is utilized to measure the similarity between the original and the revealed secret image to minimize the average distance between each pair of them.

$$\mathscr{L}_{Sec} = \frac{1}{2}\left\|X_{sec} - \hat{X}_{sec}\right\|_2^2,$$

Like Eq. (6.3.4), when the inputs are clean images, the original secret image is set as the null images to guide the Decoder output pure black results to avoid the over-fitting problem.

In summary, the total objective function for training Rosin is a weighted sum of the above losses, which is defined as:

$$\mathscr{L} = \lambda_R\mathscr{L}_R + \lambda_A\mathscr{L}_{Att} + \lambda_F\mathscr{L}_{Feat} + \lambda_R\mathscr{L}_{Rep} + \lambda_S\mathscr{L}_{Sec},$$

where $\lambda_R, \lambda_A, \lambda_F, \lambda_R, \lambda_S$ are tunable constant weighting corresponded loss. Unless stated otherwise, the λ values are set as $\lambda_R = 10, \lambda_A = 0.1, \lambda_F = 1, \lambda_R = 1$ and $\lambda_S = 1$.

6.3.5 Rosin Processing Pipeline

In summary, the Rosin framework can be divided into two main stages: concealing and revealing. In the concealing phase (light purple area in Fig. 6.2), the sender first maps a secret image to a secret representation. This secret representation is then embedded into the disentangled cover identity feature using the fuser network. The fused feature, combined with the original cover attributes, is used by the synthesis network to generate the container image, which can be transmitted to the receiver.

In the revealing phase (light yellow area in Fig. 6.2), the receiver uses the same identity encoder to extract the identity feature from the container image, which conceals the secret representation. The Separator network is then used to separate the secret representation from the identity feature. Finally, the secret Decoder is applied to reconstruct the secret image based on the extracted secret representation.

Compared to existing image steganography methods, our method conceals and transmits the secret image via the identity feature of the cover and container images. This approach provides higher robustness due to the stability and invariance of the identity feature under conventional image distortions. Additionally, the container image synthesized by the AAD-based generator achieves high fidelity, enhancing the imperceptibility of secret transmission. As a result, our method achieves a balanced trade-off between imperceptibility and robustness.

6.4 Experiments

In this section, we present the results from our extensive experiments to evaluate Rosin's performance from the following aspects: (1). Imperceptibility of concealing the secret image in the container and fidelity of the revealed secret image. (2). Robustness of the embedded secret image; (3). Concealing and revealing phases' computational overhead and efficiency; (4). Scalability to hide a larger secret image in the container. Furthermore, we implement detailed discussions of how Rosin works. The experiment results demonstrate that our method can perform best across various qualitative and quantitative evaluation metrics.

6.4.1 Experiments Setting

Datasets: We train our method on the widely used face image dataset **Flickr-Faces-HQ (FFHQ) [71]**, and then conduct experiments and comparison on another famous face dataset **CelebA-HQ [70]**. The gap between training and testing datasets can validate our method's generalisation. Unless stated otherwise, all images in the experiment have been aligned and cropped to the size of 256×256.

Metrics: Peak-Signal-to-Noise Ratio (PSNR) and **Structural Similarity Index Measure (SSIM)** are used to calculate the similarity between the cover/container and original/revealed secret image pairs to assess their quality. Moreover, following the same setting as previous works [91, 147], these two metrics are also employed to evaluate the robustness of revealing the embedded secret image from distorted container images. Overall, in this paper, higher PSNR and SSIM values indicate better fidelity of the container and revealed secret images, and greater robustness in revealing the embedded secret image from distorted container images.

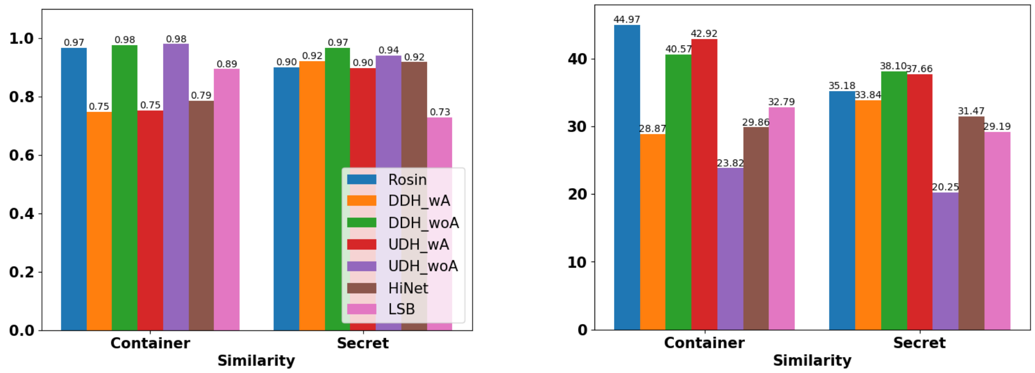
Baselines: We compare Rosin with SOTA invisible image-hiding methods to reflect its advantages and disadvantages in various aspects. To ensure a fair comparison, we only consider methods with a payload capacity higher than 24 bpp, which allows them to hide an entire image within a cover image. Furthermore, we select the method whose authors have made their source codes and pre-trained models publicly available, ensuring the reproducibility of our experimental results. Consequently, three learning-based methods: HiNet [67], DDH [143], and UDH [155], along with one traditional method: 4bit-LSB [128] are selected as baselines in our experiment. We re-train the learning-based methods following the original authors' configuration using Rosin's training dataset. Besides, we note that the original DDH and UDH methodologies can incorporate additional attack layers in their training process to enhance robustness. Hence, we train another group of DDH and UDH models equipped with their respective attack layers. In our experiment, we use the suffix w_A and w/o_A to distinguish DDH or UDH models with/without attack layers.

6.4.2 Imperceptibility and fidelity

We first evaluate each method's imperceptibility of concealing a secret image in the container and the fidelity of revealing the secret image. These two benchmarks are critical in the image steganography task, as they decide whether hiding secret images compromises the utility of the container image and whether the secret image can be accurately transmitted. For a fair comparison, we employ all methods to conceal the same secret image within 1k randomly selected CelebaHQ images to generate container images. Then, we reveal the embedded secret image from these container images producing revealed secret images. These images are used for the following comparisons.

Similarity with Original: We calculate the averaged PSNR and SSIM values of each method's cover/container and original/revealed secret image pairs. The results are illustrated in the histogram of Fig. 6.3.

We observe that Rosin, HiNet, DDH_w/o_A and UDH_w/o_A achieve close performance in the similarity between the cover and container images, while our Rosin slightly falls behind in secret images' similarity. However, our method significantly outperforms the residual of baselines. The relatively lower performance of Rosin in secret image



(a) SSIM

(b) PSNR

Figure 6.3: SSIM and PSNR values of cover/container and original/revealed secret image pairs from different methods. Rosin has comparable performance in both pairs with baselines, but earns greatly enhanced robustness in return.

quality can be attributed to the process of compressing the secret image into a latent representation will discard some less essential details, thereby reducing the similarity to the original image. Nonetheless, the gap is minimal, only 0.04 SSIM and 2.0 PSNR db lower, so it is safe to conclude that our method is comparable with SOTA. In addition, equipping attack layers significantly drop the imperceptibility and fidelity of DDH and UDH, demonstrating that this strategy cannot achieve a balanced trade-off between robustness and other metrics.

In addition to the quantitative comparison, the visualisation of each method's outputs is illustrated in Fig. 6.4 for qualitative assessment. We also show the ten times magnified residual between cover/container and original/revealed secret image pairs. Despite not achieving the best quantitative performance, the qualitative results in Fig. 6.4 demonstrate that our method's container and revealed secret images are sufficiently naturalistic and accurately preserve the original hue and light conditions, making them visually appealing to human observers. When examining the magnified residual maps, we observe that the difference between our cover and container images is nearly imperceptible, indicating that Rosin successfully conceals the secret image within the cover image. The main discrepancies are primarily concentrated in the facial area, highlighting that Rosin predominantly embeds the secret image into the container image's identity feature.

In addition, our method can nearly perfectly recover the secret image, i.e., the residual between the recovered image and the ground-truth secret image is nearly all in black, even after ten times magnification. In contrast, DDH_w_A, UDH_w_A, and LSB

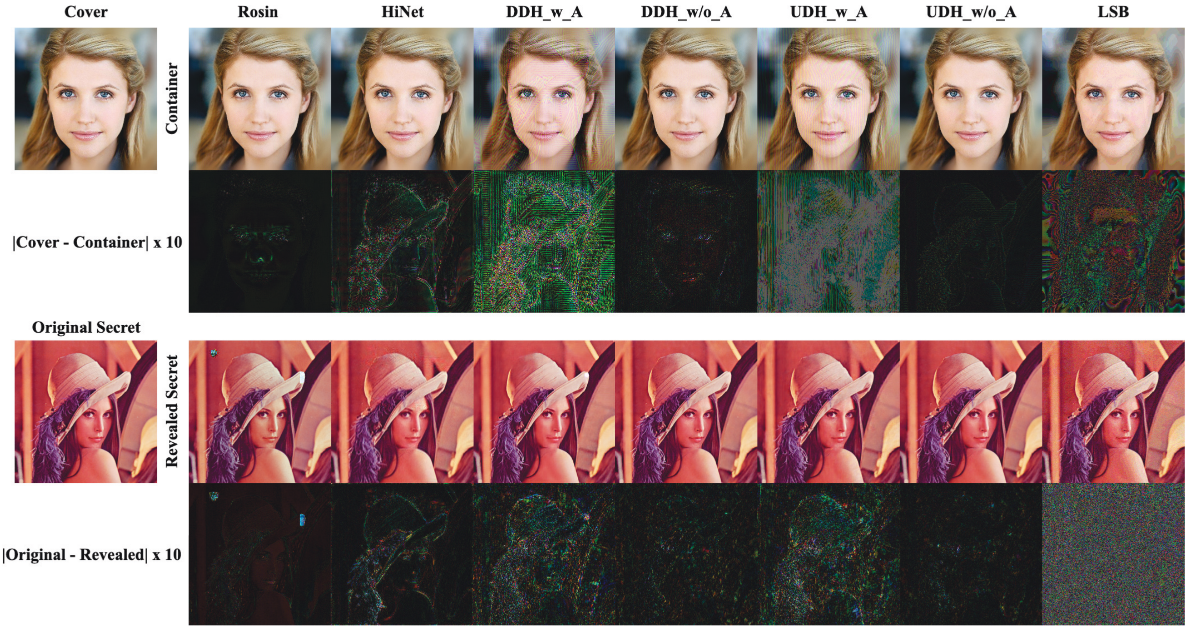


Figure 6.4: Visualization of each method's output and residual maps (Magnified x10). Our outputs are perceptually identical to the original, indicating that embedding a secret image into the container will not compromise its utility, and the secret image can also be precisely transmitted. The magnified residues between the cover and our container show that the majority manipulation area of Rosin is centralized at the face area. Besides, it also shows that our method has relatively fewer residues between the original and revealed secret images.

container images have noticeable texture-copying artifacts, especially in smooth regions. LSB container images also suffer from undesirable colour deviation issues, leading to visible blurring artifacts.

Steganographic analysis: Apart from the similarity between cover and container images, the undetectability from steganalysis tools can also demonstrate the image steganography method's imperceptibility. Therefore, we employ a well-known open-source steganalysis tool called StegExpose [16] to measure the anti-detection ability of each method's container images. Specifically, StegExpose is used to differentiate the mixed cover images and corresponding container images from each method. The results are then used to draw the receiver operating characteristic curve (ROC) in Fig. 6.5 and calculate the corresponding area under the curve (AUC) value.

The computed AUC values for each method are as follows: Rosin (0.5158), HiNet (0.9850), DDH w_A (0.7274), DDH w/o_A (0.5721), UDH w_A (0.8935), UDH w/o_A (0.8084) and LSB (0.6058), where our method's AUC value is the closest to 0.5. Besides, our ROC in Fig 6.5 most closely aligns with that of a random guess. Both numerical

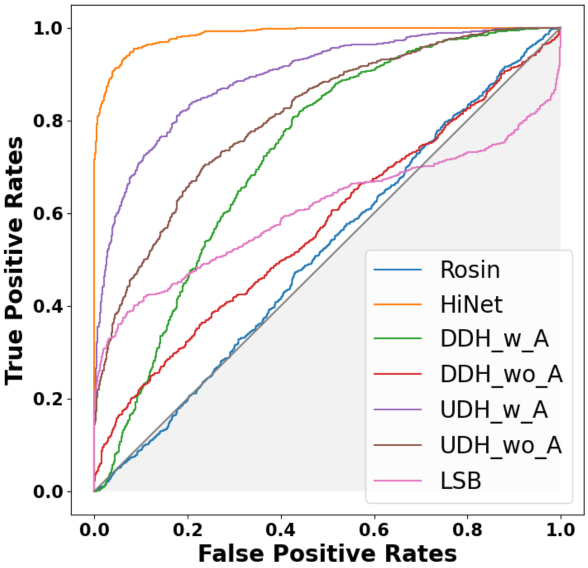


Figure 6.5: The ROC curve produced by StegExpose for different methods' container images. The shadow area represents the random guess's area under the curve. Rosin's ROC curve is closest to the random guess, representing its container images can almost evasion the StegExpose's detection.

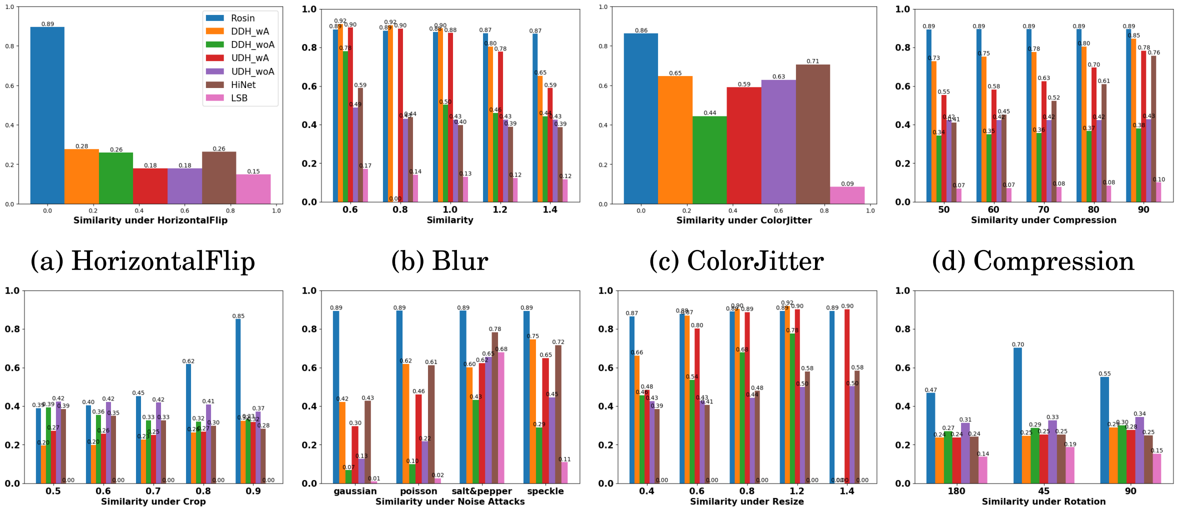
and graphical results indicate that Rosin's container images offer high security, thereby evading detection by the statistical steganalysis tool. We infer this is due to the unique embedding process of Rosin, where the secret image is exclusively embedded within the facial regions of the container images, thereby leading to minimal and restricted alterations that are challenging to detect.

In summary, qualitative and quantitative results demonstrate that our method has high imperceptibility in concealing the secret image in the container, comparable fidelity of the revealed secret image, and can also fool the StegExpose tool with high probability.

6.4.3 Robustness

We proceed to evaluate the robustness of each method in terms of extracting the embedded secret images from distorted container images. The robustness here refers to the distortion tolerance range, a crucial aspect in real-world image steganography tasks. This is because container images often encounter various lossy distortions during their transmission in practice, and the embedded secret image should remain effective and extractable despite these distortions. Ignoring the robustness will render the embedded secret image vulnerable and fragile, limiting its application in practice.

To comprehensively evaluate each method's robustness, we employ various commonly



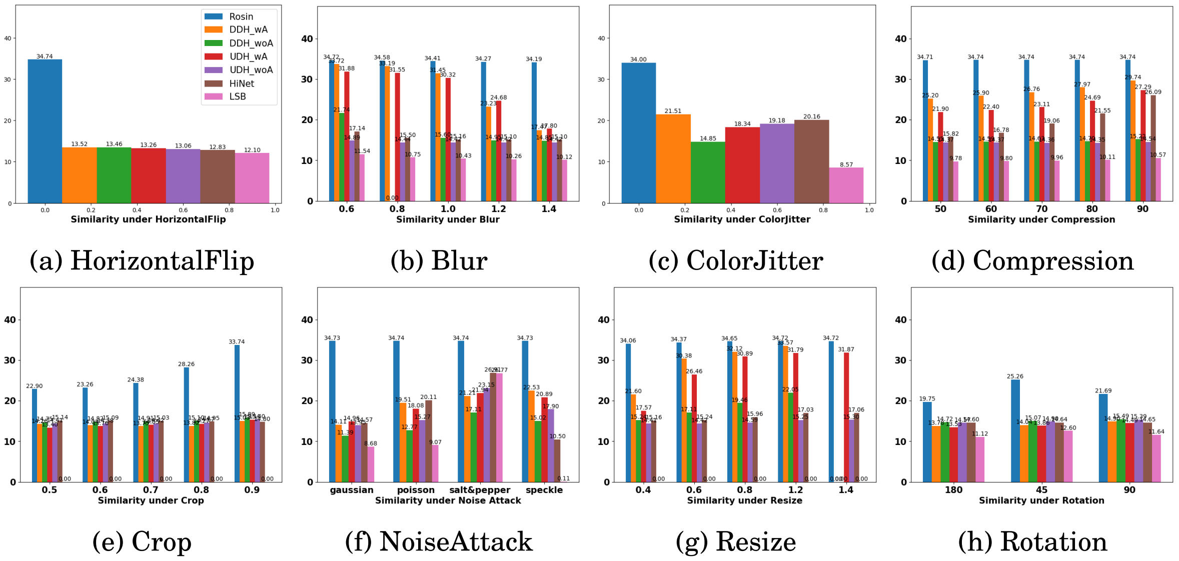
(a) HorizontalFlip (b) Blur (c) ColorJitter (d) Compression

(e) Crop (f) NoiseAttack (g) Resize (h) Rotation

Figure 6.6: Robustness comparison with SOTAs. The histograms show the averaged SSIM values of revealed secret images from different post-processing container images. The plotting scale in each sub-figure is the same, so a higher bar represents higher SSIM values and indicates better robustness. Rosin almost outperforms SOTAs in all sets and is immune to some distortions, such as Blur and Compression.

used image post-processing techniques, including Flip, Blur, Color Adjustment, JPEF Compression, Crop, Noise Attack, Resize and Rotation. Fig. 6.8 displays examples of each post-processing technique's output and corresponding setting. We choose a range of distortions strong enough to differentiate the performance between different methods but in a regime where the distorted image still resembles the original. Each post-processing technique is applied to 1k container images from Rosin and other baselines to generate corresponding sets of distorted container images. Then, we use these image steganography methods to extract the embedded secret images from these distorted container images. Finally, following previous works [91, 147], we calculate the averaged SSIM and PSNR values of the revealed and original secret images to reflect each method's robustness against image distortions.

The evaluation results are reported in Fig. 6.6 and Fig. 6.7, demonstrating that our method almost outperforms all baselines across all distortions. According to the results, despite mild interference, e.g., blur and resize, on the container image, the secret restoration of HiNet and LSB witness a substantial drop in performance. The embedded secret images of DDH and UDH without attack layers are also susceptible to distortions on their container images, with performance lower than HiNet and LSB in some respects, such as noise attack and colour adjustment. These findings confirm



(a) HorizontalFlip

(b) Blur

(c) ColorJitter

(d) Compression

(e) Crop

(f) NoiseAttack

(g) Resize

(h) Rotation

Figure 6.7: Robustness comparison with SOTAs. The histograms show the averaged PSNR values of revealed secret images from different post-processing container images. The plotting scale in each sub-figure is the same, so a higher bar represents higher PSNR values and indicates better robustness.

that overlooking distortions makes methods inapplicable in practice. Furthermore, the results demonstrate that DDH and UDH, when equipped with attack layers, can improve robustness but are only effective for limited types of distortions introduced in their training process. For example, by incorporating a differentiable compression attack in the training process, DDH and UDH can efficiently combat JPEG compression. However, when confronted with unknown distortions, such as rotation, DDH and UDH perform poorly and fail entirely.

In contrast, Rosin does not suffer from the above issues. It delivers superior robust performance across various distortions. The results even suggest that the embedded secret image of Rosin is immune to some image post-processing techniques, such as HorizontalFlip, Blur, Colorjitter, and compression, where the performance remains unimpacted with the distortion strength increase. This is because these manipulations do not distort the container image's identity feature, so it would not affect our embedded secret image's revealing. Nevertheless, our method is relatively susceptible to Crop and Rotation, although it still outperforms baselines with a considerable advantage margin. Specifically, there is a clear performance downtrend with the crop size increasing. We believe this phenomenon is because cropping large container image areas will inevitably damage its identity feature, posing a threat to our identity feature embedding. Similar to the crop, the rotation manipulation will also produce non-negligible changes in the

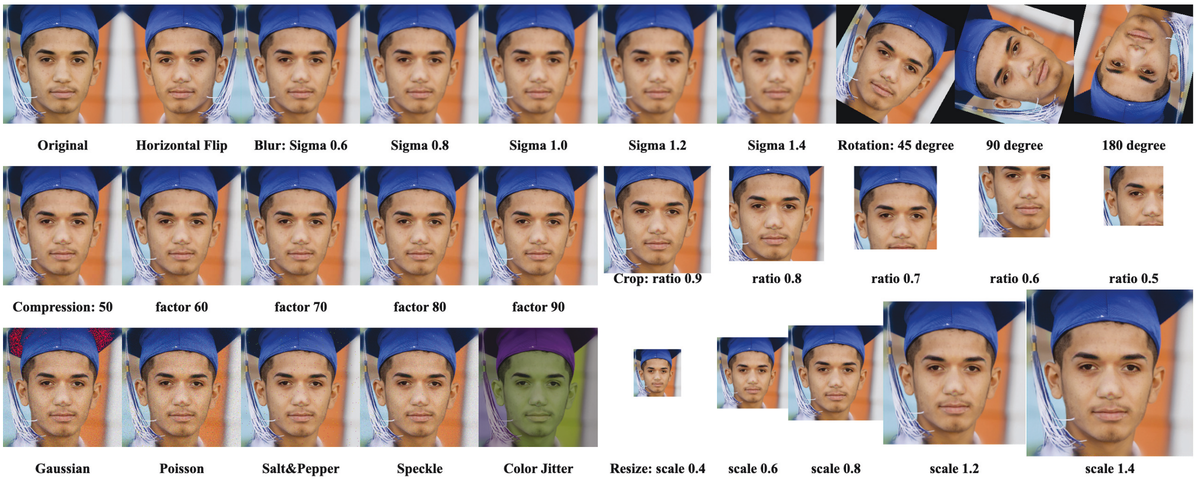


Figure 6.8: This visualization showcases the outputs and corresponding configurations for various image post-processing techniques. Each image displays the changes by a specific distortion, demonstrating the potential challenges encountered during image steganography.

Table 6.2: FPS, Parameters and MACs of Concealing&Revealing.

	FPS	**Parameters (M)**	**MACs (G)**
HiNet	0.21&0.69	4.05	66.46
DDH	48.69&48.98	16.66&0.74	16.35&48.71
UDH	46.78&57.32	16.66&0.74	16.30&48.71
LSB	1.92&1.81	N/A	N/A
Rosin	2.17&6.65	449.87&75.27	116.85&6.89

containers' identity feature, which led our method to fail in some samples. Despite this, it is pretty encouraging to see that Rosin can partially withstand these two powerful attacks while others fail.

In conclusion, the robustness evaluation shows our superior performance against common image distortions. Coupled with the results of the last section, we can confidently affirm that our method strikes a balanced trade-off between robustness and imperceptibility.

6.4.4 Efficiency and Overhead

Next, we assess each method's efficiency and computational overhead. Regarding efficiency, we test both learning-based and traditional methods' concealing and revealing Frame Per Second (FPS) on an Intel i7 13700K/KF CPU environment without any GPU acceleration for a fair comparison. As for computational overhead, we count each

Table 6.3: SSIM and PSNR of different resolution secret images.

	256 x 256	**512 x 512**	**1024 x 1024**
SSIM			
Container	0.9701	0.9557	0.9479
Secret	0.9030	0.8964	0.8749
PSNR			
Container	44.97	42.39	41.51
Secret	35.34	33.49	31.95

method's number of parameters and Multiply Accumulate Operations (MACs). The results are given in Table 6.2.

Not surprisingly, Rosin has the most parameters and MACs due to its complex design. However, its concealing and revealing are faster than HiNet and LSB, as Rosin's subnetworks adopt the simple backbone. Besides, when accelerating with a single NVIDIA GeForce RTX 4090 GPU, the FPS of Rosin can improve to concealing: 347.61 and revealing 1063.48, which is still not the best one but sufficient for real-time application. Considering the robustness of our method, we believe it is worth accepting the computation overhead of our method.

6.4.5 Scalability

We subsequently enlarge the embedded secret image's resolution to examine the hiding scalability of Rosin. We separately retrain three Rosin models using the same cover images but with secret images of three different resolutions, which are 256x256, 512x512, and 1024x1024. To do that, we keep the majority of these models' networks identical, only adjusting the Secret Encoder's input layer and the Decoder's out layer to fit different resolutions. Then, we calculated the averaged PSNR and SSIM values of each model's 1k cover/container and original and revealed secret image pairs.

Table 6.3 presents computed results. With the increase of the secret image's resolution, the PSNR and SSIM slightly drop but still maintain acceptable values. We believe the performance degradation is due to the Decoder network's reconstruction ability limitation. A more sophisticated network might achieve higher similarities. These results also demonstrate that the 512-dimensional secret representation vector can preserve the essential information of a larger secret image. Therefore, it indicates that the identity feature of the cover image also has redundancy to embed a larger secret image representation, which exhibits the scalability of Rosin.

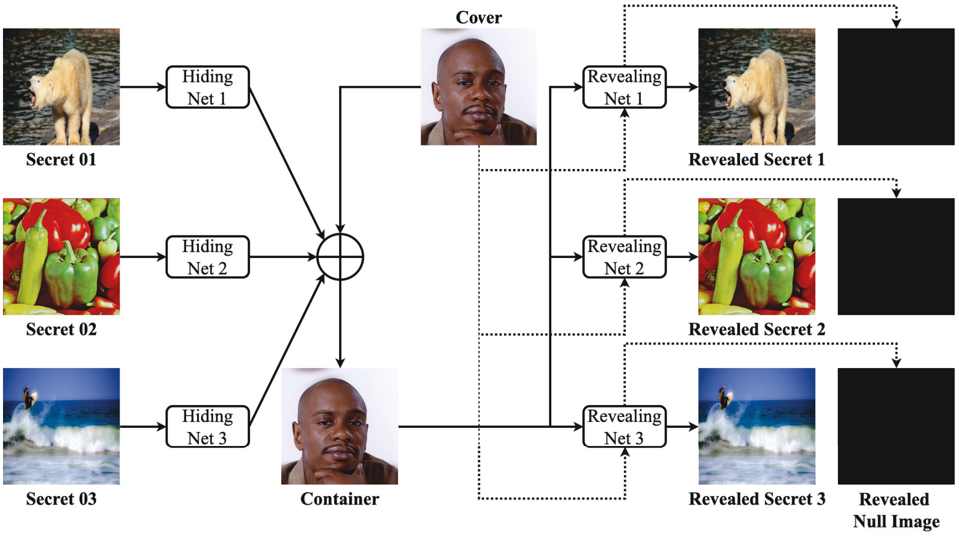


Figure 6.9: Pipeline for specificity analyses. Rosin can successfully differentiate cover and container images, and can also accurately retrieve its corresponding secret images.

Table 6.4: SSIM and PSNR of different secret images.

	Secret 1	**Secret 2**	**Secret 3**
SSIM	0.9807	0.9747	0.9789
PSNR	44.88	43.92	44.51

6.4.6 Discussion

According to the above examinations, it is clear that our method has superior performance in image steganography tasks. However, a deeper understanding of our method's functioning is required. Consequently, this section aims to conduct an investigation to analyze Rosin and its outputs.

Specificity: We begin our investigation by analyzing the specificity of Rosin. The specificity here represents the revealing network's ability to differentiate its corresponding concealing network's container image from the clean image or other concealing network's container image. In other words, when the revealing network receives clean images or container images from other embedded networks, it should output a null image, i.e., a pure black image.

To demonstrate the specificity performance of Rosin, we experiment with multiple recipients receiving different secret images from the same container image. We train three pairs of concealing and revealing networks to hide and retrieve their respective secret images but hide the secret images in the same cover image, i.e., container = cover + secret 1 + secret 2 + secret 3. The overall procedure is depicted in Fig. 6.9, and

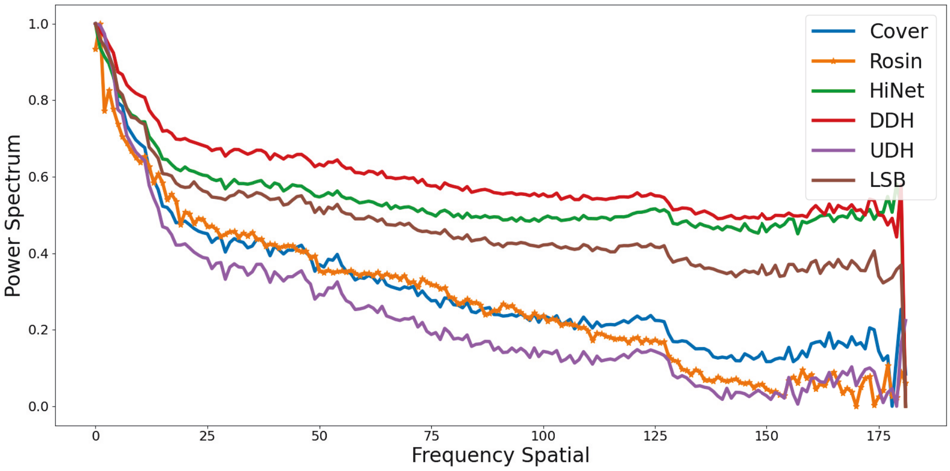


Figure 6.10: Container image frequency analysis results. Rosin's frequency distribution almost aligns with that of cover images, while others have a significantly more high-frequency spectrum.

quantitative results are provided in Table 6.4. From our observations, the retrieval performance is reasonably good for all three recipients without revealing the wrong secrets. Besides, we use these three revealing networks to retrieve secret images in a clean image. All three networks yield null images, demonstrating that they can easily distinguish between the container and cover images.

Frequency Distribution: We then compute the averaged Azimuthal Integral (**AI**) [85] values of different images involved in our method to explore their frequency properties. In brief, Azimuthal Integral returns the relative frequency intensity distribution spectrum, where the intensity begins with the highest value at the lowest frequency and decreases as the frequency increases. The outcomes are plotted in Fig. 6.10 and Fig. 6.11. We also present other methods' AI values for comparison.

From Fig. 6.10, we observed that the frequency distribution of our container images almost aligns with that of the cover images, only slightly rising in the distribution's tails. On the contrary, the frequency distribution of HiNet, DDH and UDH container images obviously has a more high-frequency spectrum. It is because the convolutional neural network (CNNs) used in learning-based methods tend to have an abnormal distribution in their high-frequency domain [34, 85]. Besides, high-frequency contents in images are generally invisible to human observers [124], making learning-based methods tend to embed the secret image into the high-frequency components of the container image to achieve high imperceptibility. However, high-frequency hiding leads the embedded secret image less robust [29, 37] and susceptible to image distortions aimed at the frequency domain, such as JPEG compression. Our method relies less on high-frequency, as it

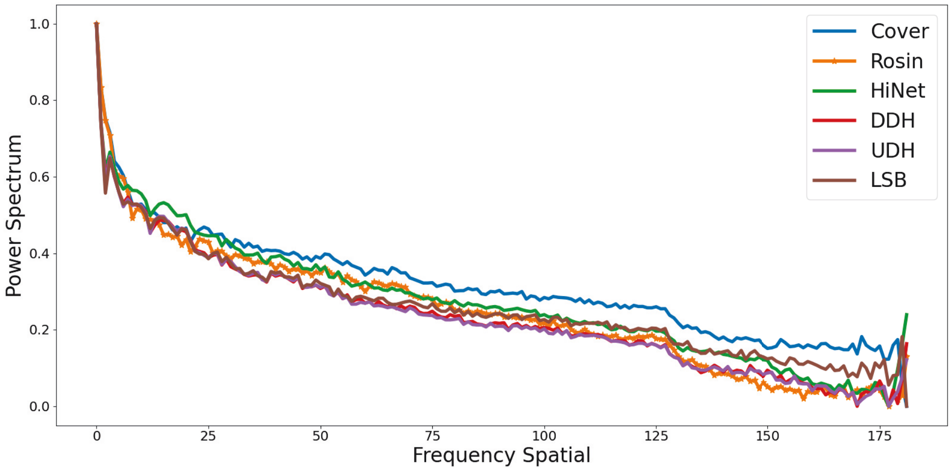


Figure 6.11: Secret image frequency analysis results. The original and Rosin's revealed secret image's frequency distributions are almost perfectly aligned in most frequency domains but slightly differ in the high-frequency.

embeds the secret representation rather than the entire secret image and only embeds it into the identity feature. This advantage makes our method more robust.

The original and revealed secret images' frequency distributions in Fig. 6.11 are almost perfectly aligned in most frequency domains but still slightly differ in the high-frequency domain. While this phenomenon is also due to the abnormal high-frequency problem of CNNs, we consider these minor differences acceptable, considering the previous experimental results.

6.5 Conclusion and Discussion

Although imperceptibility and capacity are critical metrics of image steganography, overlooking robustness renders existing image steganography methods inapplicable in the real world. Current solutions to enhance these methods' robustness are often limited to specific distortions and may also compromise other essential metrics. This study presents Rosin, a new learning-based image steganography framework that exploits the stability of identity features to improve robustness against various image distortions. Experimental results confirm that Rosin exhibits exceptional robustness while maintaining comparable imperceptibility and fidelity with SOTA methods. We firmly believe that Rosin strikes a balanced trade-off between robustness and other key metrics, thereby extending the practicality of image steganography in real-world applications. Moreover, our innovative semantic image-hiding method paves the way for a new research direction, promising further advancements in image steganography.

CHAPTER 7

Conclusion and Future Work

7.1 Conclusion

The work presented in this thesis addressed the security and privacy threats arising from applying novel learning-based techniques to image data. Two prominent threats were primarily tackled: Surveillance and Tracking, and Malicious Forgery and Tampering. By developing novel techniques and methodologies, this work effectively solved the above threats from the following aspects: Sensitive information sanitization, Forgery detection, Authenticity protection, and Authorship proof.

We proposed a novel approach for semantically sanitizing sensitive information in the latent space of neural networks to protect against privacy breaches, striking a balance between privacy protection and utility preservation. We moved away from the traditional artifact-based detection methods and introduced a unique framework for proactive forgery media detection, leveraging face identity feature watermarking. To defend against malicious tampering on images, we develop a novel learning-based semi-fragile image watermark for pixel-level authentication. Finally, to establish reliable authorship and combat plagiarism and copyright infringement, we designed a robust image-hiding method that utilized semantic information in images.

Each contribution made in this thesis showcased the potential of learning-based techniques in securing image data against malicious threats. The findings not only expanded our understanding of the challenges posed by learning-based methods but also provided innovative solutions to enhance the security and privacy of image data.

By addressing these threats and introducing novel methodologies, this research has advanced the field's knowledge and contributed to developing practical strategies for protecting image data. The outcomes of this work demonstrate the importance of considering security and privacy implications in applying learning-based techniques to image data, and offer insights into the potential future directions for further research in this area.

7.2 Future work

Looking ahead, several potential directions for future research emerged from this work. Firstly, as learning-based techniques continue to evolve, the solutions proposed in this thesis should be regularly re-evaluated and enhanced to maintain their effectiveness. The continuous development of more sophisticated malicious techniques necessitates an ongoing commitment to updating and improving our methods.

Secondly, the scope of our research primarily revolved around image data. However, similar threats exist for other forms of media, such as video and audio data. Extending the proposed solutions to these domains would be an essential step in broadening the effectiveness of our methods.

Thirdly, this work focused on individual solutions for each threat. There may be value in investigating a more integrated approach that can address multiple threats simultaneously. Such a unified model could potentially offer greater efficiency and more robust protection against learning-based threats.

Lastly, while this thesis has primarily addressed the technical aspects of the threats, future research could also explore the legal and ethical dimensions. For instance, investigating how legal frameworks could support technical solutions, or examining the ethical implications of using such technologies.

By pursuing these directions, we can continue to advance our defence against learning-based threats, ensuring the security and privacy of media content in an increasingly digital world.

Bibliography

[1] R. Abdal, Y. Qin, and P. Wonka, *Image2stylegan: How to embed images into the stylegan latent space?*, in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 4432–4441.

[2] ——, *Image2stylegan++: How to edit the embedded images?*, in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 8296–8305.

[3] F. Alturki and R. Mersereau, *Secure blind image steganographic technique using discrete fourier transformation*, in Proceedings 2001 International Conference on Image Processing (Cat. No. 01CH37205), vol. 2, IEEE, 2001, pp. 542–545.

[4] I. Amerini, T. Uricchio, L. Ballan, and R. Caldelli, *Localization of jpeg double compression through multi-domain convolutional neural networks*, in 2017 IEEE Conference on computer vision and pattern recognition workshops (CVPRW), IEEE, 2017, pp. 1865–1871.

[5] V. Asnani, X. Yin, T. Hassner, S. Liu, and X. Liu, *Proactive image manipulation detection*, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 15386–15395.

[6] D. Baby, J. Thomas, G. Augustine, E. George, and N. R. Michael, *A novel dwt based image securing method using steganography*, Procedia Computer Science, 46 (2015), pp. 612–618.

[7] S. Baluja, *Hiding images in plain sight: Deep steganography*, Advances in neural information processing systems, 30 (2017).

[8] ——, *Hiding images within images*, IEEE transactions on pattern analysis and machine intelligence, 42 (2019), pp. 1685–1697.

[9] Q. Bammey, R. G. v. Gioi, and J.-M. Morel, *An adaptive neural network for unsupervised mosaic consistency analysis in image forensics*, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 14194–14204.

[10] J. Bao, D. Chen, F. Wen, H. Li, and G. Hua, *Towards open-set identity preserving face synthesis*, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 6713–6722.

[11] J. H. Bappy, A. K. Roy-Chowdhury, J. Bunk, L. Nataraj, and B. Manjunath, *Exploiting spatial structure for localizing manipulated image regions*, in Proceedings of the IEEE international conference on computer vision, 2017, pp. 4970–4979.

[12] J. H. Bappy, C. Simons, L. Nataraj, B. Manjunath, and A. K. Roy-Chowdhury, *Hybrid lstm and encoder–decoder architecture for detection of image forgeries*, IEEE Transactions on Image Processing, 28 (2019), pp. 3286–3300.

[13] M. Barni, F. Bartolini, and A. Piva, *Improved wavelet-based watermarking through pixel-wise masking*, IEEE transactions on image processing, 10 (2001), pp. 783–791.

[14] B. Bayar and M. C. Stamm, *A deep learning approach to universal image manipulation detection using a new convolutional layer*, in Proceedings of the 4th ACM workshop on information hiding and multimedia security, 2016, pp. 5–10.

[15] R. Bhardwaj and V. Sharma, *Image steganography based on complemented message and inverted bit lsb substitution*, Procedia Computer Science, 93 (2016), pp. 832–838.

[16] B. Boehm, *Stegexpose-a tool for detecting lsb steganography*, arXiv preprint arXiv:1410.6656, (2014).

[17] N. Bonettini, E. D. Cannas, S. Mandelli, L. Bondi, P. Bestagini, and S. Tubaro, *Video face manipulation detection through ensemble of cnns*, in 2020 25th International Conference on Pattern Recognition (ICPR), 2021, pp. 5012–5019.

[18] J. Cao, B. Liu, Y. Wen, R. Xie, and L. Song, *Personalized and invertible face de-identification by disentangled identity information manipulation*, in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 3334–3342.

[19] N. Carlini and H. Farid, *Evading deepfake-image detectors with white-and black-box attacks*, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2020, pp. 658–659.

[20] T. Carvalho, H. Farid, and E. R. Kee, *Exposing photo manipulation from user-guided 3d lighting analysis*, in Media Watermarking, Security, and Forensics 2015, vol. 9409, SPIE, 2015, p. 940902.

[21] L. Chai, D. Bau, S.-N. Lim, and P. Isola, *What makes fake images detectable? understanding properties that generalize*, in European Conference on Computer Vision, Springer, 2020, pp. 103–120.

[22] R. Chen, X. Chen, B. Ni, and Y. Ge, *Simswap: An efficient framework for high fidelity face swapping*, in Proceedings of the 28th ACM International Conference on Multimedia, 2020, pp. 2003–2011.

[23] Y. Choi, Y. Uh, J. Yoo, and J.-W. Ha, *Stargan v2: Diverse image synthesis for multiple domains*, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 8188–8197.

[24] U. A. Ciftci, I. Demir, and L. Yin, *Fakecatcher: Detection of synthetic portrait videos using biological signals*, IEEE Transactions on Pattern Analysis and Machine Intelligence, (2020).

[25] D. Cozzolino, D. Gragnaniello, and L. Verdoliva, *Image forgery localization through the fusion of camera-based, feature-based and pixel-based techniques*, in 2014 IEEE International Conference on Image Processing (ICIP), IEEE, 2014, pp. 5302–5306.

[26] D. Cozzolino, G. Poggi, and L. Verdoliva, *Splicebuster: A new blind image splicing detector*, in 2015 IEEE International Workshop on Information Forensics and Security (WIFS), IEEE, 2015, pp. 1–6.

[27] N. Dalal and B. Triggs, *Histograms of oriented gradients for human detection*, 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05), 1 (2005), pp. 886–893.

[28] H. Dang, F. Liu, J. Stehouwer, X. Liu, and A. K. Jain, *On the detection of digital face manipulation*, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern recognition, 2020, pp. 5781–5790.

[29] H. Daren, L. Jiufen, H. Jiwu, and L. Hongmei, *A dwt-based image watermarking algorithm*, in IEEE International Conference on Multimedia and Expo, 2001. ICME 2001., IEEE Computer Society, 2001, pp. 80–80.

[30] T. J. De Carvalho, C. Riess, E. Angelopoulou, H. Pedrini, and A. de Rezende Rocha, *Exposing digital image forgeries by illumination color classification*, IEEE Transactions on Information Forensics and Security, 8 (2013), pp. 1182–1194.

[31] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, *Arcface: Additive angular margin loss for deep face recognition*, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 4690–4699.

[32] C. Dong, X. Chen, R. Hu, J. Cao, and X. Li, *Mvss-net: Multi-view multi-scale supervised networks for image manipulation detection*, IEEE Transactions on Pattern Analysis and Machine Intelligence, (2022).

[33] J. Dong, W. Wang, and T. Tan, *Casia image tampering detection evaluation database*, in 2013 IEEE China Summit and International Conference on Signal and Information Processing, IEEE, 2013, pp. 422–426.

[34] R. Durall, M. Keuper, and J. Keuper, *Watch your up-convolution: Cnn based generative deep neural networks are failing to reproduce spectral distributions*, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 7890–7899.

[35] R. Durall, M. Keuper, F.-J. Pfreundt, and J. Keuper, *Unmasking deepfakes with simple features*, arXiv preprint arXiv:1911.00686, (2019).

[36] T. Dzanic, K. Shah, and F. Witherden, *Fourier spectrum discrepancies in deep network generated images*, Advances in neural information processing systems, 33 (2020), pp. 3022–3032.

[37] H.-Y. Fan, Z.-M. Lu, and Y.-L. Liu, *A low-frequency construction watermarking based on histogram*, Multimedia Tools and Applications, 79 (2020), pp. 5693–5717.

[38] Q. Fan, J. Yang, G. Hua, B. Chen, and D. Wipf, *A generic deep architecture for single image reflection removal and image smoothing*, in Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 3238–3247.

[39] Y. Fan, P. Carré, and C. Fernandez-Maloigne, *Image splicing detection with local illumination estimation*, in 2015 IEEE international conference on Image processing (ICIP), IEEE, 2015, pp. 2940–2944.

[40] FanÔºåLiyue, *Practical image obfuscation with provable privacy*, 2019 IEEE International Conference on Multimedia and Expo (ICME), (2019).

[41] P. Ferrara, T. Bianchi, A. De Rosa, and A. Piva, *Image forgery localization via fine-grained analysis of cfa artifacts*, IEEE Transactions on Information Forensics and Security, 7 (2012), pp. 1566–1577.

[42] T. Filler, J. Judas, and J. Fridrich, *Minimizing embedding impact in steganography using trellis-coded quantization*, in Media forensics and security II, vol. 7541, SPIE, 2010, pp. 38–51.

[43] J. Frank, T. Eisenhofer, L. Schönherr, A. Fischer, D. Kolossa, and T. Holz, *Leveraging frequency analysis for deep fake image recognition*, in International conference on machine learning, PMLR, 2020, pp. 3247–3258.

[44] J. Fridrich and J. Kodovsky, *Rich models for steganalysis of digital images*, IEEE Transactions on information Forensics and Security, 7 (2012), pp. 868–882.

[45] O. Gafni, L. Wolf, and Y. Taigman, *Live face de-identification in video*, in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 9378–9387.

[46] A. Gandhi and S. Jain, *Adversarial perturbations fool deepfake detectors*, in 2020 International Joint Conference on Neural Networks (IJCNN), IEEE, 2020, pp. 1–8.

[47] G. Gao, H. Huang, C. Fu, Z. Li, and R. He, *Information bottleneck disentanglement for identity swapping*, in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2021, pp. 3404–3413.

[48] L. Guo, J. Ni, and Y. Shi, *Uniform embedding for efficient jpeg steganography*, IEEE transactions on Information Forensics and Security, 9 (2014), pp. 814–825.

[49] L. Guo, J. Ni, and Y. Q. Shi, *An efficient jpeg steganographic scheme using uniform embedding*, in 2012 IEEE International Workshop on Information Forensics and Security (WIFS), IEEE, 2012, pp. 169–174.

[50] A. Haliassos, K. Vougioukas, S. Petridis, and M. Pantic, *Lips don't lie: A generalisable and robust approach to face forgery detection*, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 5039–5049.

[51] J. Hayes and G. Danezis, *Generating steganographic images via adversarial training*, Advances in neural information processing systems, 30 (2017).

[52] K. He, X. Zhang, S. Ren, and J. Sun, *Deep residual learning for image recognition*, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[53] Y. He, N. Yu, M. Keuper, and M. Fritz, *Beyond the spectrum: Detecting deepfakes via re-synthesis*, arXiv preprint arXiv:2105.14376, (2021).

[54] Z. He, W. Zuo, M. Kan, S. Shan, and X. Chen, *Attgan: Facial attribute editing by only changing what you want*, IEEE transactions on image processing, 28 (2019), pp. 5464–5478.

[55] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, *Gans trained by a two time-scale update rule converge to a local nash equilibrium*, (2018).

[56] V. Holub and J. Fridrich, *Designing steganographic distortion using directional filters*, in 2012 IEEE International workshop on information forensics and security (WIFS), IEEE, 2012, pp. 234–239.

[57] V. Holub and J. Fridrich, *Digital image steganography using universal distortion*, in Proceedings of the first ACM workshop on Information hiding and multimedia security, 2013, pp. 59–68.

[58] V. Holub, J. Fridrich, and T. Denemark, *Universal distortion function for steganography in an arbitrary domain*, EURASIP Journal on Information Security, 2014 (2014), pp. 1–13.

[59] C.-T. Hsu and J.-L. Wu, *Hidden digital watermarks in images*, IEEE Transactions on image processing, 8 (1999), pp. 58–68.

[60] X. Hu, Z. Zhang, Z. Jiang, S. Chaudhuri, Z. Yang, and R. Nevatia, *Span: Spatial pyramid attention network for image manipulation localization*, in European conference on computer vision, Springer, 2020, pp. 312–328.

[61] Y. Huang, F. Juefei-Xu, R. Wang, Q. Guo, L. Ma, X. Xie, J. Li, W. Miao, Y. Liu, and G. Pu, *Fakepolisher: Making deepfakes more detection-evasive by shallow reconstruction*, in Proceedings of the 28th ACM International Conference on Multimedia, 2020, pp. 1217–1226.

[62] Y. Huang, Y. Wang, Y. Tai, X. Liu, P. Shen, S. Li, J. Li, and F. Huang, *Curricularface: adaptive curriculum learning loss for deep face recognition*, proceedings of the IEEE/CVF conference on computer vision and pattern recognition, (2020), pp. 5901–5910.

[63] H. Hukkelås, R. Mester, and F. Lindseth, *Deepprivacy: A generative adversarial network for face anonymization*, in Advances in Visual Computing: 14th International Symposium on Visual Computing, ISVC 2019, Lake Tahoe, NV, USA, October 7–9, 2019, Proceedings, Part I 14, Springer, 2019, pp. 565–578.

[64] S. Imaizumi and K. Ozawa, *Multibit embedding algorithm for steganography of palette-based images*, in Image and Video Technology: 6th Pacific-Rim Symposium, PSIVT 2013, Guanajuato, Mexico, October 28-November 1, 2013. Proceedings 6, Springer, 2014, pp. 99–110.

[65] A. Islam, C. Long, A. Basharat, and A. Hoogs, *Doa-gan: Dual-order attentive generative adversarial network for image copy-move forgery detection and localization*, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 4676–4685.

[66] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, *Image-to-image translation with conditional adversarial networks*, CVPR, (2017).

[67] J. Jing, X. Deng, M. Xu, J. Wang, and Z. Guan, *Hinet: deep image hiding by invertible network*, in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 4733–4742.

[68] K.-H. Jung and K.-Y. Yoo, *Steganographic method based on interpolation and lsb substitution of digital images*, Multimedia Tools and Applications, 74 (2015), pp. 2143–2155.

[69] S. Jung and M. Keuper, *Spectral distribution aware image generation*, in Proceedings of the AAAI conference on artificial intelligence, vol. 35, 2021, pp. 1734–1742.

[70] T. Karras, T. Aila, S. Laine, and J. Lehtinen, *Progressive growing of gans for improved quality, stability, and variation*, arXiv preprint arXiv:1710.10196, (2017).

[71] T. Karras, S. Laine, and T. Aila, *A style-based generator architecture for generative adversarial networks*, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 4401–4410.

[72] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, *Analyzing and improving the image quality of stylegan*, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 8110–8119.

[73] S. Katzenbeisser and F. Petitcolas, *Digital watermarking*, Artech House, London, 2 (2000).

[74] B. Li, M. Wang, J. Huang, and X. Li, *A new cost function for spatial image steganography*, in 2014 IEEE International conference on image processing (ICIP), IEEE, 2014, pp. 4206–4210.

[75] H. Li and J. Huang, *Localization of deep inpainting using high-pass fully convolutional network*, in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 8301–8310.

[76] J. Li, H. Xie, J. Li, Z. Wang, and Y. Zhang, *Frequency-aware discriminative feature learning supervised by single-center loss for face forgery detection*, in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2021, pp. 6458–6467.

[77] L. Li, J. Bao, H. Yang, D. Chen, and F. Wen, *Faceshifter: Towards high fidelity and occlusion aware face swapping*, arXiv preprint arXiv:1912.13457, (2019).

[78] L. Li, J. Bao, T. Zhang, H. Yang, D. Chen, F. Wen, and B. Guo, *Face x-ray for more general face forgery detection*, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 5001–5010.

[79] T. Li and L. Lin, *Anonymousnet: Natural face de-identification with measurable privacy*, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, (2019), pp. 0–0.

[80] Y. Li, M.-C. Chang, and S. Lyu, *In ictu oculi: Exposing ai created fake videos by detecting eye blinking*, in 2018 IEEE International Workshop on Information Forensics and Security (WIFS), IEEE, 2018, pp. 1–7.

[81] X. Liao, J. Yin, M. Chen, and Z. Qin, *Adaptive payload distribution in multiple images steganography based on image texture features*, IEEE Transactions on Dependable and Secure Computing, (2020).

[82] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, *Microsoft coco: Common objects in context*, in European conference on computer vision, Springer, 2014, pp. 740–755.

[83] Z. Lin, J. He, X. Tang, and C.-K. Tang, *Fast, automatic and fine-grained tampered jpeg image detection via dct coefficient analysis*, Pattern Recognition, 42 (2009), pp. 2492–2501.

[84] B. Liu, M. Ding, S. Shaham, W. Rahayu, F. Farokhi, and Z. Lin, *When machine learning meets privacy: A survey and outlook*, ACM Computing Surveys (CSUR), 54 (2021), pp. 1–36.

[85] C. Liu, H. Chen, T. Zhu, J. Zhang, and W. Zhou, *Making deepfakes more spurious: evading deep face forgery detection via trace removal attack*, arXiv preprint arXiv:2203.11433, (2022).

[86] H. Liu, X. Li, W. Zhou, Y. Chen, Y. He, H. Xue, W. Zhang, and N. Yu, *Spatial-phase shallow learning: rethinking face forgery detection in frequency domain*, in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2021, pp. 772–781.

[87] X. Liu, Z. Ma, J. Ma, J. Zhang, G. Schaefer, and H. Fang, *Image disentanglement autoencoder for steganography without embedding*, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 2303–2312.

[88] Z. Liu, P. Luo, X. Wang, and X. Tang, *Deep learning face attributes in the wild*, Proceedings of the IEEE international conference on computer vision, (2015), pp. 3730–3738.

[89] Z. Liu, X. Qi, and P. H. Torr, *Global texture enhancement for fake face detection in the wild*, in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 8060–8069.

[90] S.-P. Lu, R. Wang, T. Zhong, and P. L. Rosin, *Large-capacity image steganography based on invertible neural networks*, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 10816–10825.

[91] X. Luo, R. Zhan, H. Chang, F. Yang, and P. Milanfar, *Distortion agnostic deep watermarking*, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 13548–13557.

[92] Y. Luo, Y. Zhang, J. Yan, and W. Liu, *Generalizing face forgery detection with high-frequency features*, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 16317–16326.

[93] S. Lyu, *Deepfake detection: Current challenges and next steps*, in 2020 IEEE International Conference on Multimedia & Expo Workshops (ICMEW), IEEE, 2020, pp. 1–6.

[94] S. Lyu, X. Pan, and X. Zhang, *Exposing region splicing forgeries with blind local noise estimation*, International journal of computer vision, 110 (2014), pp. 202–221.

[95] F. Marra, D. Gragnaniello, L. Verdoliva, and G. Poggi, *Do gans leave artificial fingerprints?*, in 2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR), IEEE, 2019, pp. 506–511.

[96] I. Masi, A. Killekar, R. M. Mascarenhas, S. P. Gurudatt, and W. AbdAlmageed, *Two-branch recurrent network for isolating deepfakes in videos*, in Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VII 16, Springer, 2020, pp. 667–684.

[97] M. Maximov, I. Elezi, and L. Leal-Taixé, *Ciagan: Conditional identity anonymization generative adversarial networks*, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, (2020), pp. 5447–5456.

[98] S. McCloskey and M. Albright, *Detecting gan-generated imagery using color cues*, arXiv preprint arXiv:1812.08247, (2018).

[99] R. McPherson, R. Shokri, and V. Shmatikov, *Defeating image obfuscation with deep learning.*
arXiv eprint:1609.00408 cs.CR, 2016.

[100] S.-M. Mun, S.-H. Nam, H.-U. Jang, D. Kim, and H.-K. Lee, *A robust blind watermarking using convolutional neural network*, arXiv preprint arXiv:1704.03248, (2017).

[101] L. Nataraj, T. M. Mohammed, B. Manjunath, S. Chandrasekaran, A. Flenner, J. H. Bappy, and A. K. Roy-Chowdhury, *Detecting gan generated fake images using co-occurrence matrices*, Electronic Imaging, 2019 (2019), pp. 532–1.

[102] R. Natsume, T. Yatagawa, and S. Morishima, *Rsgan: face swapping and editing using face and hair representation in latent spaces*, arXiv preprint arXiv:1804.03447, (2018).

[103] T.-T. Ng, J. Hsu, and S.-F. Chang, *Columbia image splicing detection evaluation dataset*, DVMM lab. Columbia Univ CalPhotos Digit Libr, (2009).

[104] B. C. Nguyen, S. M. Yoon, and H.-K. Lee, *Multi bit plane image steganography*, in Digital Watermarking: 5th International Workshop, IWDW 2006, Jeju Island, Korea, November 8-10, 2006. Proceedings 5, Springer, 2006, pp. 61–70.

[105] M. Niimi, H. Noda, E. Kawaguchi, and R. O. Eason, *High capacity and secure digital steganography to palette-based images*, in Proceedings. International Conference on Image Processing, vol. 2, IEEE, 2002, pp. II–II.

[106] Y. Nitzan, A. Bermano, Y. Li, and D. Cohen-Or, *Face identity disentanglement via latent space mapping*, arXiv preprint arXiv:2005.07728, (2020).

[107] S. J. Oh, R. Benenson, M. Fritz, and B. Schiele, *Faceless person recognition: Privacy implications in social media*, European Conference on Computer Vision(ECCV), (2016), pp. 19–35.

[108] F. Pan, J. Li, and X. Yang, *Image steganography method based on pvd and modulus function*, in 2011 International Conference on Electronics, Communications and Control (ICECC), IEEE, 2011, pp. 282–284.

[109] X. Pan, X. Zhang, and S. Lyu, *Exposing image splicing with inconsistent local noise variances*, in 2012 IEEE International conference on computational photography (ICCP), IEEE, 2012, pp. 1–10.

[110] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, *Semantic image synthesis with spatially-adaptive normalization*, in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 2337–2346.

[111] T. Pevnỳ, P. Bas, and J. Fridrich, *Steganalysis by subtractive pixel adjacency matrix*, in Proceedings of the 11th ACM workshop on Multimedia and security, 2009, pp. 75–84.

[112] T. Pevnỳ, T. Filler, and P. Bas, *Using high-dimensional image models to perform highly undetectable steganography*, in Information Hiding: 12th International Conference, IH 2010, Calgary, AB, Canada, June 28-30, 2010, Revised Selected Papers 12, Springer, 2010, pp. 161–177.

[113] C. I. Podilchuk and E. J. Delp, *Digital watermarking: algorithms and applications*, IEEE signal processing Magazine, 18 (2001), pp. 33–46.

[114] N. Provos and P. Honeyman, *Hide and seek: An introduction to steganography*, IEEE security & privacy, 1 (2003), pp. 32–44.

[115] Y. Qian, G. Yin, L. Sheng, Z. Chen, and J. Shao, *Thinking in frequency: Face forgery detection by mining frequency-aware clues*, in European Conference on Computer Vision, Springer, 2020, pp. 86–103.

[116] C. Qin, C.-C. Chang, Y.-H. Huang, and L.-T. Liao, *An inpainting-assisted reversible steganographic scheme using a histogram shifting mechanism*, IEEE transactions on circuits and systems for video technology, 23 (2012), pp. 1109–1118.

[117] D. Rawat and V. Bhandari, *A steganography technique for hiding image in an image using lsb method for 24 bit color image*, International Journal of Computer Applications, 64 (2013).

[118] Z. Ren, Y. J. Lee, and M. S. Ryoo, *Learning to anonymize faces for privacy preserving action detection*, Proceedings of the european conference on computer vision (ECCV), (2018), pp. 620–636.

[119] E. Richardson, Y. Alaluf, O. Patashnik, Y. Nitzan, Y. Azar, S. Shapiro, and D. Cohen-Or, *Encoding in style: a stylegan encoder for image-to-image translation*, in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2021, pp. 2287–2296.

[120] J. Ruanaidh, W. Dowling, and F. M. Boland, *Phase watermarking of digital images*, in Proceedings of 3rd IEEE International Conference on Image Processing, vol. 3, IEEE, 1996, pp. 239–242.

[121] R. Salloum, Y. Ren, and C.-C. J. Kuo, *Image splicing localization using a multitask fully convolutional network (mfcn)*, Journal of Visual Communication and Image Representation, 51 (2018), pp. 201–209.

[122] H. Shi, J. Dong, W. Wang, Y. Qian, and X. Zhang, *Ssgan: Secure steganography based on generative adversarial networks*, in Advances in Multimedia Information Processing–PCM 2017: 18th Pacific-Rim Conference on Multimedia, Harbin, China, September 28-29, 2017, Revised Selected Papers, Part I 18, Springer, 2018, pp. 534–544.

[123] K. Shiohara and T. Yamasaki, *Detecting deepfakes with self-blended images*, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 18720–18729.

[124] A. K. Singh, M. Dave, and A. Mohan, *Hybrid technique for robust and imperceptible multiple watermarking using medical images*, Multimedia Tools and Applications, 75 (2016), pp. 8381–8401.

[125] Q. SUN, L. MA, S. J. OH, L. VAN GOOL, B. SCHIELE, AND M. FRITZ, *Natural and effective obfuscation by head inpainting*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 5050–5059.

[126] Q. SUN, A. TEWARI, W. XU, M. FRITZ, C. THEOBALT, AND B. SCHIELE, *A hybrid model for identity obfuscation by face replacement*, in Proceedings of the European conference on computer vision (ECCV), 2018, pp. 553–569.

[127] C. SZEGEDY, V. VANHOUCKE, S. IOFFE, J. SHLENS, AND Z. WOJNA, *Rethinking the inception architecture for computer vision*, Proceedings of the IEEE conference on computer vision and pattern recognition, (2016), pp. 2818–2826.

[128] A. A. TAMIMI, A. M. ABDALLA, AND O. AL-ALLAF, *Hiding an image inside another image using variable-rate steganography*, International Journal of Advanced Computer Science and Applications (IJACSA), 4 (2013).

[129] M. TANCIK, B. MILDENHALL, AND R. NG, *Stegastamp: Invisible hyperlinks in physical photographs*, in 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 2117–2126.

[130] W. TANG, B. LI, S. TAN, M. BARNI, AND J. HUANG, *Cnn-based adversarial embedding for image steganography*, IEEE Transactions on Information Forensics and Security, 14 (2019), pp. 2074–2087.

[131] W. TANG, S. TAN, B. LI, AND J. HUANG, *Automatic steganographic distortion learning using a generative adversarial network*, IEEE Signal Processing Letters, 24 (2017), pp. 1547–1551.

[132] P. TSAI, Y.-C. HU, AND H.-L. YEH, *Reversible image hiding scheme using predictive coding and histogram shifting*, Signal processing, 89 (2009), pp. 1129–1143.

[133] T. K. TSUI, X.-P. ZHANG, AND D. ANDROUTSOS, *Color image watermarking using multidimensional fourier transforms*, IEEE Transactions on Information Forensics and security, 3 (2008), pp. 16–28.

[134] N. VISHWAMITRA, B. KNIJNENBURG, H. HU, Y. P. KELLY CAINE, ET AL., *Blur vs. block: Investigating the effectiveness of privacy-enhancing obfuscation for images*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, (2017), pp. 39–47.

[135] D. Volkhonskiy, I. Nazarov, and E. Burnaev, *Steganographic generative adversarial networks*, in Twelfth international conference on machine vision (ICMV 2019), vol. 11433, SPIE, 2020, pp. 991–1005.

[136] C. Wang and W. Deng, *Representative forgery mining for fake face detection*, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 14923–14932.

[137] H.-P. Wang, T. Orekondy, and M. Fritz, *Infoscrub: Towards attribute privacy by targeted obfuscation*, arXiv preprint arXiv:2005.10329, (2020).

[138] R. Wang, F. Juefei-Xu, M. Luo, Y. Liu, and L. Wang, *Faketagger: Robust safeguards against deepfake dissemination via provenance tracking*, in Proceedings of the 29th ACM International Conference on Multimedia, 2021, pp. 3546–3555.

[139] S.-Y. Wang, O. Wang, R. Zhang, A. Owens, and A. A. Efros, *Cnn-generated images are surprisingly easy to spot... for now*, in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 8695–8704.

[140] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, *Image quality assessment: from error visibility to structural similarity*, IEEE transactions on image processing, 13 (2004), pp. 600–612.

[141] Y. Wen, B. Liu, R. Xie, Y. Zhu, J. Cao, and L. Song, *A hybrid model for natural face de-identiation with adjustable privacy*, 2020 IEEE International Conference on Visual Communications and Image Processing (VCIP), (2020), pp. 269–272.

[142] Y. Wen, L. Song, B. Liu, M. Ding, and R. Xie, *Identitydp: Differential private identification protection for face images*, arXiv preprint arXiv:2103.01745, (2021).

[143] X. Weng, Y. Li, L. Chi, and Y. Mu, *High-capacity convolutional video steganography with temporal residual modeling*, in Proceedings of the 2019 on International Conference on Multimedia Retrieval, 2019, pp. 87–95.

[144] D.-C. Wu and W.-H. Tsai, *A steganographic method for images by pixel-value differencing*, Pattern recognition letters, 24 (2003), pp. 1613–1626.

[145] Y. WU, W. ABDALMAGEED, AND P. NATARAJAN, *Mantra-net: Manipulation tracing network for detection and localization of image forgeries with anomalous features*, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 9543–9552.

[146] Y. WU, F. YANG, AND H. LING, *Privacy-protective-gan for face de-identification*, arXiv preprint arXiv:1806.08906, (2018).

[147] Y. XU, C. MOU, Y. HU, J. XIE, AND J. ZHANG, *Robust invertible image steganography*, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 7875–7884.

[148] C.-Z. YANG, J. MA, S. WANG, AND A. W.-C. LIEW, *Preventing deepfake attacks on speaker authentication by dynamic lip movement analysis*, IEEE Transactions on Information Forensics and Security, 16 (2020), pp. 1841–1854.

[149] J. YANG, D. RUAN, J. HUANG, X. KANG, AND Y.-Q. SHI, *An embedding cost learning framework using gan*, IEEE Transactions on Information Forensics and Security, 15 (2019), pp. 839–851.

[150] X. YANG, Y. LI, AND S. LYU, *Exposing deep fakes using inconsistent head poses*, in ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2019, pp. 8261–8265.

[151] Y. YANG, C. LIANG, H. HE, X. CAO, AND N. Z. GONG, *Faceguard: Proactive deepfake detection*, arXiv preprint arXiv:2109.05673, (2021).

[152] N. YU, L. S. DAVIS, AND M. FRITZ, *Attributing fake images to gans: Learning and analyzing gan fingerprints*, in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 7556–7566.

[153] N. YU, V. SKRIPNIUK, S. ABDELNABI, AND M. FRITZ, *Artificial fingerprinting for generative models: Rooting deepfake attribution in training data*, in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 14448–14457.

[154] P. YU, J. FEI, Z. XIA, Z. ZHOU, AND J. WENG, *Improving generalization by commonality learning in face forgery detection*, IEEE Transactions on Information Forensics and Security, (2022).

[155] C. ZHANG, P. BENZ, A. KARJAUV, G. SUN, AND I. S. KWEON, *Udh: Universal deep hiding for steganography, watermarking, and light field messaging.*, in 34th Conference on Neural Information Processing Systems, NeurIPS 2020, vol. 33, 2020, pp. 10223–10234.

[156] C. ZHANG, A. KARJAUV, P. BENZ, AND I. S. KWEON, *Towards robust deep hiding under non-differentiable distortions for practical blind watermarking*, in Proceedings of the 29th ACM international conference on multimedia, 2021, pp. 5158–5166.

[157] K. A. ZHANG, A. CUESTA-INFANTE, L. XU, AND K. VEERAMACHANENI, *Steganogan: High capacity image steganography with gans*, arXiv preprint arXiv:1901.03892, (2019).

[158] R. ZHANG, S. DONG, AND J. LIU, *Invisible steganography via generative adversarial networks*, Multimedia tools and applications, 78 (2019), pp. 8559–8575.

[159] R. ZHANG, P. ISOLA, A. A. EFROS, E. SHECHTMAN, AND O. WANG, *The unreasonable effectiveness of deep features as a perceptual metric*, Proceedings of the IEEE conference on computer vision and pattern recognition, (2018), pp. 586–595.

[160] X. ZHANG, S. KARAMAN, AND S.-F. CHANG, *Detecting and simulating artifacts in gan fake images*, in 2019 IEEE International Workshop on Information Forensics and Security (WIFS), IEEE, 2019, pp. 1–6.

[161] X. ZHANG AND S. WANG, *Vulnerability of pixel-value differencing steganography to histogram analysis and modification for enhanced security*, Pattern Recognition Letters, 25 (2004), pp. 331–339.

[162] Y. ZHANG, J. GOH, L. L. WIN, AND V. L. THING, *Image region forgery detection: A deep learning approach.*, SG-CRC, 2016 (2016), pp. 1–11.

[163] H. ZHAO, W. ZHOU, D. CHEN, T. WEI, W. ZHANG, AND N. YU, *Multi-attentional deepfake detection*, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 2185–2194.

[164] P. ZHOU, B.-C. CHEN, X. HAN, M. NAJIBI, A. SHRIVASTAVA, S.-N. LIM, AND L. DAVIS, *Generate, segment, and refine: Towards generic manipulation seg-*

mentation, in Proceedings of the AAAI conference on artificial intelligence, vol. 34, 2020, pp. 13058–13065.

[165] P. ZHOU, X. HAN, V. I. MORARIU, AND L. S. DAVIS, *Two-stream neural networks for tampered face detection*, in 2017 IEEE conference on computer vision and pattern recognition workshops (CVPRW), IEEE, 2017, pp. 1831–1839.

[166] P. ZHOU, X. HAN, V. I. MORARIU, AND L. S. DAVIS, *Learning rich features for image manipulation detection*, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 1053–1061.

[167] J. ZHU, R. KAPLAN, J. JOHNSON, AND L. FEI-FEI, *Hidden: Hiding data with deep networks*, in Proceedings of the European conference on computer vision (ECCV), 2018, pp. 657–672.