UTS UNIVERSITY OF TECHNOLOGY SYDNEY

Deep Image Forgery: An Investigation on Forensic and Anti-forensic Techniques

by Chi Liu

Thesis submitted in fulfilment of the requirements for
the degree of

Doctor of Philosophy

under the supervision of A/Prof. Tianqing Zhu & Prof. Wanlei Zhou

University of Technology Sydney
Faculty of Engineering and Information Technology

Jan 07, 2023

Certificate of Original Authorship

Required wording for the certificate of original authorship
CERTIFICATE OF ORIGINAL AUTHORSHIP I, Chi Liu, declare that this thesis is submitted in fulfillment 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. **Signature:** Production Note: Signature removed prior to publication. **Date:** 11/13/2023

C02029: Doctor of Philosophy
CRICOS Code: 0000
PhD Thesis: Computer Science
Dec 2022

Deep Image Forgery:
An Investigation on Forensic and Anti-forensic Techniques

Chi Liu

School of Computer Science
Faculty of Engg. & IT
University of Technology Sydney
NSW - 2007, Australia

Deep Image Forgery:
An Investigation on Forensic and Anti-forensic Techniques

A thesis submitted in partial fulfilment of the requirements for the degree of

Doctor of Philosophy
in
Computer Science

by

Chi Liu

to

School of Computer Science
Faculty of Engineering and Information Technology
University of Technology Sydney
NSW - 2007, Australia

Jan 2023

Abstract

Deep image forgeries powered by deep learning models, e.g., deepfakes, are increasingly challenging the belief that seeing is believing. The image privacy and security threats raised by deep image forgery, such as misleading information on social media, have become a major concern in the security community. Effective countermeasures are impelling. A common countermeasure is developing detection systems to distinguish fake images from real ones. Despite a series of forensic detectors having been proposed, there are still several open challenges, such as the cross-domain generalization ability and the robustness against attacks. Also, the countermeasures should be constantly updated given the continuous technical advances behind deep image forgery. These challenges can be further understood and facilitated from two rival technical perspectives: forensics and anti-forensics. The forensic direction aims to develop more robust and generalized detection systems that can deal with forgeries in complex or unknown environments. The anti-forensic direction aims to reveal the vulnerability and weakness of a detection system by designing possible attacks to enable forged images to bypass the detection.

In this thesis, we study the deep image forgery detection problem with a focus on resolving the open challenges newly emerging in this field. We investigate the problem from both forensic and anti-forensic perspectives to provide comprehensive solutions. Regarding the forensic direction, we have proposed two forgery detection methods: one exploits multi-level GAN model fingerprinting to enable task-specific forensics, and the other uses a multi-view reconstruction-classification learning framework for generalized and robust detection. Regarding the anti-forensic direction, we have designed a novel black-box attack specific to deep image forgery detection systems, called the trace removal attack. In addition, we have provided a closer look at the generalization and robustness issues of deep image forgery detection from a frequency perspective, which link the forensic and anti-forensic research with a novel frequency alignment method benefiting both directions. For each proposed method, we have conducted extensive experimental evaluations where multiple datasets and security scenarios are involved. We also compare the methods with state-of-the-art baselines to demonstrate their superiority.

AUTHOR'S DECLARATION

I, *Chi Liu* declare that this thesis, submitted in partial 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, Australia, 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.

SIGNATURE: Chi Liu

DATE: 07th Jan, 2023

PLACE: Sydney, Australia

DEDICATION

To Xiaotong,
my love, my life, my light,
and to my little angel Shuyi,
and our former selves in the past four years . . .

Acknowledgments

I have received much support and encouragement throughout my Ph.D. project, which carries me forward on this fantastic research journey. I would like to express my deepest gratitude to my supervisors, Prof. Wanlei Zhou and A/Prof. Tianqing Zhu. You open up a whole new world of research for me. Without your kind support, insightful advice, and warm encouragement, this endeavor would not have been possible. I am also thankful to Prof. Jun Zhang, who instructed me unreservedly on how to do research surveys and scientific writing. Thanks should also go to Dr. Dayong Ye for sharing your research experience with me. I also acknowledge all team members at the Center for Cyber Security and Privacy, UTS, for the enjoyable time we learned, researched, worked, and discussed together.

Last but not least, I am deeply indebted to my wife, Xiaotong Han, and my family. Without your wholehearted support, love, and understanding, I could not have undertaken this journey.

My Ph.D. project started at the end of 2019 and was accomplished in early 2023. What a special and challenging period during which we lived with the COVID-19 epidemic, and all things, including our life and research styles, significantly changed. I am deeply indebted to all of you for standing with me and supporting me in this difficult time. This experience will definitely be a special, precious, and lifelong memory for me.

List of Publications

Related to the Thesis :

1. Chi Liu, Tianqing Zhu, Jun Zhang, and Wanlei Zhou. 2022. Privacy Intelligence: A Survey on Image Privacy in Online Social Networks. ACM Computing Surveys. 55, 8, Article 161 (August 2023), 35 pages. https://doi.org/10.1145/3547299

2. Chi Liu, Huajie Chen, Tianqing Zhu, Jun Zhang, and Wanlei Zhou. Making DeepFakes More Spurious: Evading Deep Face Forgery Detection via Trace Removal Attack. IEEE Transactions on Dependable and Secure Computing, vol. 20, no. 6, pp. 5182-5196, Nov.-Dec. 2023, doi: 10.1109/TDSC.2023.3241604.

3. Chi Liu, Tianqing Zhu, Yuan Zhao, Jun Zhang, and Wanlei Zhou. Multi-level model fingerprinting for task-specific forensics on GAN-generated images. Accepted to Computer Standards & Interfaces.

4. Chi Liu, Tianqing Zhu, Sheng Shen, and Wanlei Zhou. Towards Robust Gan-Generated Image Detection: A Multi-View Completion Representation. Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence (IJCAI-23), 464–472, 2023. 10.24963/ijcai.2023/52.

5. Chi Liu, Tianqing Zhu, Wanlei Zhou. Frequency bias matters: diving into robust and generalized deep image forgery detection. Under review in IEEE Transactions on Dependable and Secure Computing.

Others :

6. Shuai Zhou, Chi Liu, Dayong Ye, Tianqing Zhu, Wanlei Zhou, and Philip S. Yu. 2022. Adversarial Attacks and Defenses in Deep Learning: From a Perspective of Cybersecurity. ACM Computing Surveys. 55, 8, Article 163 (August 2023), 39 pages. https://doi.org/10.1145/3547330

7. Huajie Chen, Chi Liu, Tianqing Zhu, Wanlei Zhou. When Deep Learning Meets Watermarking: a Survey of Application Attacks and Defences. Under review in Computer Standards & Interfaces.

8. Huajie Chen, Tianqing Zhu, Chi Liu, Shui Yu, and Wanlei Zhou. An Overwriting Attack and Defence for Image-processing Model Watermarking. Under review in IEEE Transactions on Services Computing.

Table of Contents

List of Figures

Figure **Page**

List of Tables

Part I

Part I

CHAPTER 1

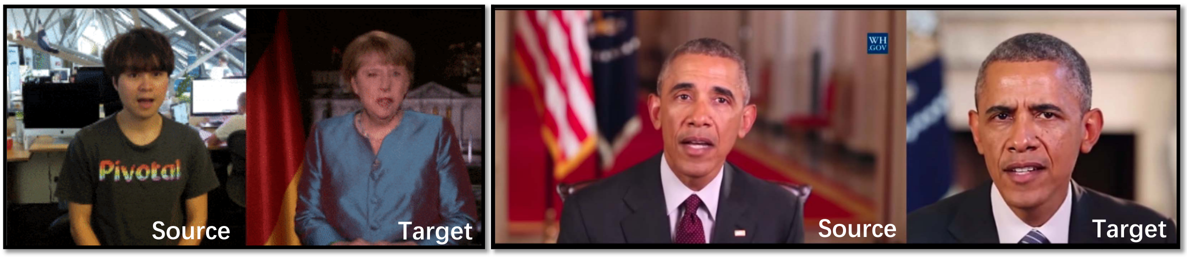
INTRODUCTION

1.1 Background

Today we are living in a digital world where digital information has become an indispensable strategic asset for almost every entity. Recently, the rapid development of Internet technologies, as well as the unprecedented prevalence of mobile camera devices have dramatically advanced the acquisition and exchange of digital data, such as images, videos, and audios, making it possible anytime and anywhere. Among these types of data, digital images, as an important and convenient medium for information presentation, storage, and communication, have become ubiquitous in our daily lives, playing crucial roles in many areas, such as journalism, judiciary, social networks, arts, and business [90].

The increasing popularity and value of digital images make image security and privacy a critical and enduring concern for society and the research community. Image forgery is a long-lasting problem challenging digital security [11, 70, 143]. Taking advantage of advanced automated image processing techniques, attackers are able to manipulate the real content of the original image, leading to severe challenges to the authenticity and originality of the images. Especially, some image forgeries with political conspiracy or commercial interests will seriously threaten social, economic, and political security. Incidents that caused public uproars due to image manipulation have occurred frequently world-wide [12, 80, 104].

Since around the year 2016, image forgery technology has entered a new era thanks to the revolutionary progress brought about by deep learning in computer vision. A new type of image forgery powered by deep learning-based generative and rendering models emerges, which we refer to as "*deep image forgery*" in this thesis. The main deep learning methods used for deep image forgery involve training generative neural network architectures, such as autoencoder [50], or generative adversarial network (GAN) [45]. Deep image forgery came into the public's consciousness along with the release of some famous programs, such as Face2Face [135] and "Synthesizing Obama" [130] (as shown in Figure 1.1), and the coining of the term "DeepFakes" which describes synthetic face-exchanged media [13]. Due to the continuous improvements in the artificial intelligence techniques behind deep image forgery, it now becomes more intelligent, automated, and easy to implement. In addition, the forged images are high-fidelity and high-quality, which are easily deceptive to human eyes. For example, Figure 1.2 shows some non-existent fake faces created by one of the latest deep generative models, StyleGAN2[69]. The images look surprisingly photorealistic, even at high resolution. As a result, the pertinent technologies have quickly attracted widespread attention regarding their potential use in the production of, e.g., child sexual abuse material, celebrity pornographic media, fake news, bullying, financial fraud, and biometric spoofing [2, 62, 121, 125]. These significant security threats have elicited responses from industry, academia, and the government to fight against deep image forgery.



(a) Face2Face result

(b) “Synthesizing Obama” result

Figure 1.1: (a) The Face2Face result. The source person's facial expression is modified in accordance with the target person's facial expression, which is called facial expression reenactment. Image credit to [139]. (b) The "Synthesizing Obama" result. Given an input Obama audio and a reference video, the system can synthesize photorealistic, lip-synced video of Obama speaking those words. Image credit to [131].

Recently, academic research has been focusing on countering deep image forgery, particularly images manipulated or automatically generated by GANs. GAN-generated images have sparked unprecedented public concern regarding deep image forgery. This concern arises from GANs outperforming traditional image processing-based forgery

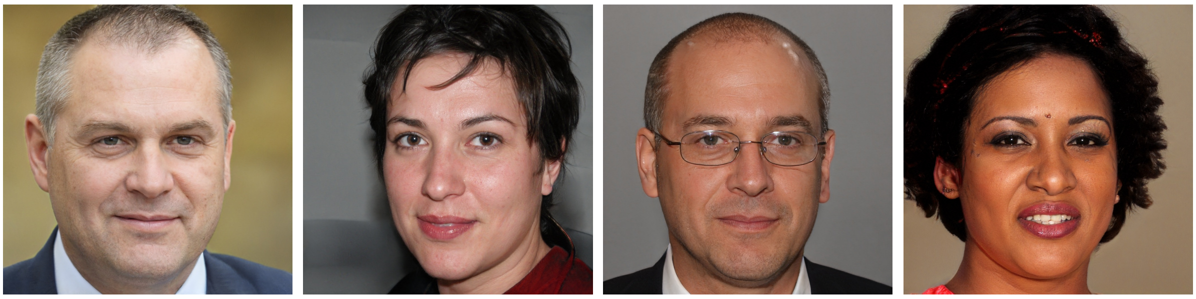


Figure 1.2: Four fake face images imagined by a StyleGAN2 model. Images are at a high resolution of 1024 × 1024.

models. GANs achieve this by pushing the fidelity of generated images to higher limits, significantly reducing forgery costs through a fully automatic pipeline. Moreover, expert knowledge in image processing is no longer a prerequisite. Existing research in this field can be typically divided into two opposite directions: **forensics** and **anti-forensics**.

- **Forensic research** aims to develop automated tools to verify the authenticity of suspicious images [115]. The possible directions include 1) detecting whether an image is forged or not, 2) attributing the source of digital images, and 3) identifying and localizing the tempered region of the images. Among these directions, GAN-generated image detection and attribution have received a surge of interest [63, 112]. An accurate detector can provide a precaution to prevent users from being jeopardized by deep image forgery. For example, it enables a filtering mechanism integrated with online services that can alert users about the risk of fakeness and raise users' awareness of security breaches caused by deep image forgery in the case that the forged images are imperceptible to humans.

- **Anti-forensic research** aims to investigate the countermeasures from an adversary's perspective [63, 117]. The investigators want to reveal the vulnerabilities and weaknesses of the target forensic systems, in most cases machine learning-based forgery detectors, in a complex context, for example, under unanticipated or malicious attacks. In anti-forensic studies, it may be necessary to conduct a security analysis with some strong, specific attacks to see how far the target forgery detectors can be pushed in the worst possible conditions. In this way, the knowledge gained from anti-forensic research can inversely help further perfect the forgery detectors. In a nutshell, forensic and anti-forensic research on deep image forgery are forming a long-term active battleground to promote the countermeasures against deep image forgery in this field, as shown in Figure 1.3.

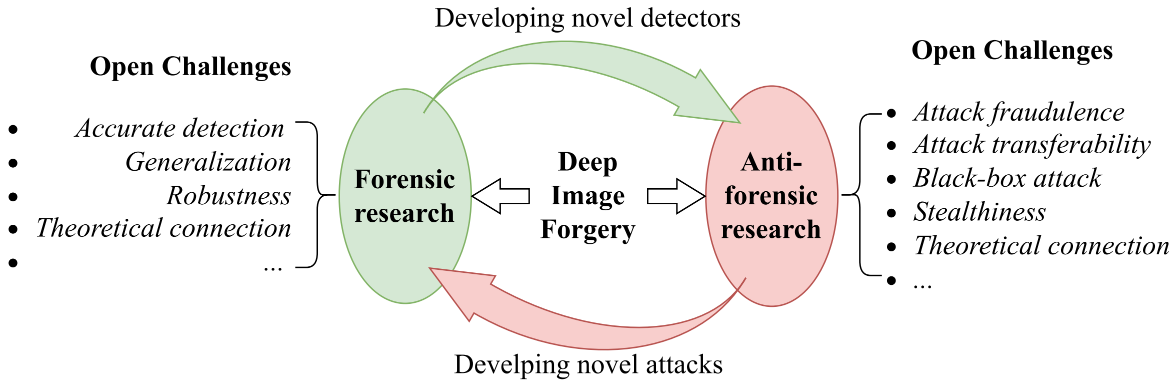


Figure 1.3: Forensic and anti-forensic research on deep image forgery are forming a long-lasting battleground where many open challenges exist for both sides.

1.2 Motivation

Even though there have been a number of groundbreaking studies that have obtained impressive achievements in both the forensic or anti-forensic directions, there are still some problems that need to be further solved on both sides. We identify some significant concerns from literature [63, 105], shown as follows.

Open challenges for forensic research:

- Accurate detection. The detection technology should be continuously upgraded to maintain high detection accuracy to deal with the fast iteration of deep generative techniques behind deep image forgery.
- Generalization. The detector is desired to generalize across various conditions, such as different semantic domains, image formats, or out-of-distribution and unseen source GAN classes.
- Robustness. The detector is desired to be robust against complex online environments or malicious attacks, where the test samples may be contaminated by different noise perturbations.

Open challenges for anti-forensic research:

- Attack fraudulence. The attack model should maintain a high attack success rate to fool the target detectors, including the latest and most sophisticated detectors.
- Attack transferability. The attack model is desired to be transferable to different detectors, even if it is optimized based on one specific detector.

- Black-box attack. The attack model is desired to be effective against black-box detectors, where the attacker does not require knowledge of or access to the target detector.

- Stealthiness. The attack model should cause negligible changes or invisible noises to the original fake samples so as to bypass human inspection.

Furthermore, there is a lack of a high-level theoretical understanding in the current literature that is capable of fully explaining the existing challenges in this field and linking the forensic and anti-forensic research on deep image forgery [63]. Figure 1.3 summarizes the major concerns to be resolved. There are also several design principle-related challenges that should be considered in developing forensic or anti-forensic techniques, such as lightweight design, low computational overhead, and good compatibility with other image-based services or online applications.

All of the above challenges motivate us to seek out novel forensic or anti-forensic techniques to contribute to winning the enduring battle against deep image forgery.

1.3 Research Objectives

In this thesis, our ultimate research goal is to address the focusing challenges demonstrated in Figure 1.3. Concretely, the research objectives include:

- We will investigate the deep image forgery problem from the forensic perspective, including the development of different detection models, frameworks, or algorithms to satisfy the requirements of forensic detectors discussed above. We will also demonstrate the effectiveness of the proposed methods in extensive experiments, and try to provide theoretical explanations if available.

- We will investigate the deep image forgery problem from anti-forensic perspectives. We will try to develop different novel attack models or algorithms to satisfy the requirements of anti-forensic attacks discussed above. We will also demonstrate the effectiveness of the proposed methods in extensive experiments, and try to provide a unified theoretical framework to connect the forensic and anti-forensic research.

1.4 Major contributions of this thesis

This thesis provides the following four major contributions lying in two directions:

1.4.1 Forensic techniques

In the forensic direction, we proposed two techniques to overcome the key forensic challenges highlighted in Figure 1.3. The first one explores the intrinsic fingerprint mechanism of GAN, which allows a fingerprint-based detection model for accurate identification of GAN-generated images and can also be used for model copyright protection and GAN model attack. The second one involves a multi-view reconstruction-classification framework, a strong feature representation method that aims at the generalization ability and robustness of GAN-generated image forensics.

1.4.1.1 A GAN fingerprint-based model

We explore the forensic solution based on detecting the tell-tale marks GANs leave behind in an image, known as GAN fingerprints. We are the first to focus on the problem that different image forensics tasks may require different distinguishability levels, and propose a task-specific GAN fingerprinting framework that supports flexible operation at different forensic levels. Concretely, we perform an in-depth analysis of GAN fingerprint dependency, providing theoretical and empirical evidence on the existence of architecture-level and instance-level GAN fingerprints in the spatial and frequency domains, respectively. From this finding, we proposed a decoupling representation framework to separate and extract the two levels of GAN fingerprints from different domains. Then we devise different implementations of the two levels of GAN fingerprints for task-specific fingerprinting in three typical forensics tasks, including fake image detection, model intellectual property protection, and fingerprinting attack and defense.

1.4.1.2 A multi-view reconstruction-classification framework

We focus on the generalized and robust detection of GAN-generated images. We draw attention to the fact that while existing detectors tend to overfit unstable features in the training set, which in turn causes failures when dealing with out-of-distribution GANs or unknown perturbation attacks. To overcome the issue, we propose a novel representation framework for GAN-generated image detection based on multi-view reconstruction classification learning. The framework first learns multiple view-to-image reconstructors to model a variety of genuine image distributions. Features represented from the distributional discrepancies characterized by the reconstructors are stable and robust for detecting unknown fake patterns. Then, a multi-view classification is devised with several novel modules and learning strategies to enhance intra-view and cross-view feature representations. The generalization and robustness of the proposed framework are confirmed through extensive experimental evaluations.

1.4.2 Anti-forensic techniques

In this direction, we proposed two novel anti-forensic attacks that satisfy the attack requirements highlighted in Figure 1.3. The first attack takes advantage of adversarial learning, by which the model can remove detectable traces from GAN-generated images to make them bypass detectors. The second one dives into the frequency domain, making fake images undetectable by calibrating their frequency distribution to that of real images. Both proposed attacks can achieve high fraudulence and transferability, rely little on the knowledge of detectors so that they can be launched in a black-box manner, and induce negligible degradation of image quality.

1.4.2.1 A trace removal attack model

We provide a universal, black-box and detector-agnostic attack called trace removal attack to evade image forgery detectors. We find that previous attacks, such as adversarial attacks, have typical detector-specific designs, which require prior knowledge about the detector, leading to poor transferability. Furthermore, these attacks only take into account basic security scenarios. It is less clear how effective they are in complex situations where the detector's defense or the attacker's knowledge may vary. The trace removal attack, in contrast, looks into the original DeepFake creation pipeline and makes an effort to remove all traces of DeepFakes in order to make the fake images appear more "authentic." The attack is more effective against arbitrary or even unknown detectors thanks to such a detector-agnostic design. The attack is implemented in the following steps. We first carry out a thorough DeepFake trace discovery, which identifies different types of distinguishable traces. Then, an adversarial learning framework with a single generator and multiple discriminators is proposed, where each discriminator is responsible for one individual trace removal task. These multiple discriminators are arranged in parallel, which prompts the generator to remove various traces simultaneously. We evaluate the efficacy of the attack in heterogeneous security scenarios where the detectors were embedded with different levels of defense and the attackers' background knowledge of data varies.

1.4.2.2 A win-win frequency alignment algorithm

We provide a closer look at the forensic and anti-forensic challenges in this field from a frequency perspective, which links the two sides, and propose a novel image processing algorithm to benefit both sides simultaneously. Specifically, we explore the root causes and connections between the generalizability and robustness issues of forgery detectors. We established a comprehensive, unified frequency analysis framework for GAN-generated image detection. Through the analysis, we confirm the frequency bias of DNN-based detectors, which can be used to fundamentally explain a number of unresolved issues associated with the robustness and generalizability of DNN-based detectors. Based on the discovery, we propose a two-step frequency alignment algorithm for removing the frequency discrepancy between real and fake images, which has the following double-sided advantages: In the anti-forensic aspect, it can be used as a strong black-box attack against forgery detectors, or, inversely, in the forensic aspect, as a universal defense to improve the reliability of forgery detectors. We also devise the corresponding attack and defense implementations, and verified the effects interactively in a wide range of experimental settings.

1.5 Thesis organization

The reminder of the thesis is organized as follows.

- **Chapter 2** gives a brief literature review of the relevant research, where we first summarize the technical progress of deep image forgery, and then discuss the recent achievements on forensic and anti-forensic research.

- **Chapter 3** introduces preliminaries of the thesis and the formulations of forensic and anti-forensic problems.

- **Chapter 4** and **Chapter 5** are the investigations on forensic techniques. **Chapter 4** introduces the multi-level GAN fingerprint-based forensic model in detail, and **Chapter 5** describes the how to use the multi-view reconstruction-classification learning for robust and generalized forgery detection.

- **Chapter 6** and **Chapter 7** include the investigations on anti-forensic techniques. **Chapter 6** introduces design and implementation of the trace removal attack, and **Chapter 7** provides a closer look at forensics and anti-forensics from a frequency perspective, along with the design and applications of the frequency alignment algorithm.

- **Chapter 8** summarizes this thesis with discussions, future insights and a conclusion.

CHAPTER 2

RELATED WORK

2.1 Deep image forgery: an overview of technical progress

Driven by the enormous potential profit, digital image forgery has a long technical history and has recently been rapidly renewed. This area can be dated back to the 19th century, during which time technology steadily improved [37]. When entering the 21st century, interests burgeoned with the development of computer-aided image processing software, such as Adobe Photoshop [132] and GNU image manipulation programs (GIMP) [78], which allow skilled attackers to create sophisticated forgeries manually [56]. However, manipulating images with these tools is commonly skill-intensive and time-consuming, while more automation is in demand. Then, recently, we have seen the arrival of fully automated image forgery techniques powered by artificial intelligence algorithms such as deep generative models. Deep image forgery techniques have been developed by academic researchers beginning in the 2010s, and later widely improved, adapted and expanded by academia and industry in the recent decade [11, 63, 70, 162].

There has not been a standard, explicit classification of deep image forgery technology. Previous taxonomies tended to be task-oriented. For example, In 2004, Farid [35] outlined six different image forgery types, i.e., compositing, morphing, re-touching, enhancing, computer generating and computer painting. More recent surveys on DeepFakes prefer to divide forgeries into four general types: face synthesis, attribute manipulation, identity

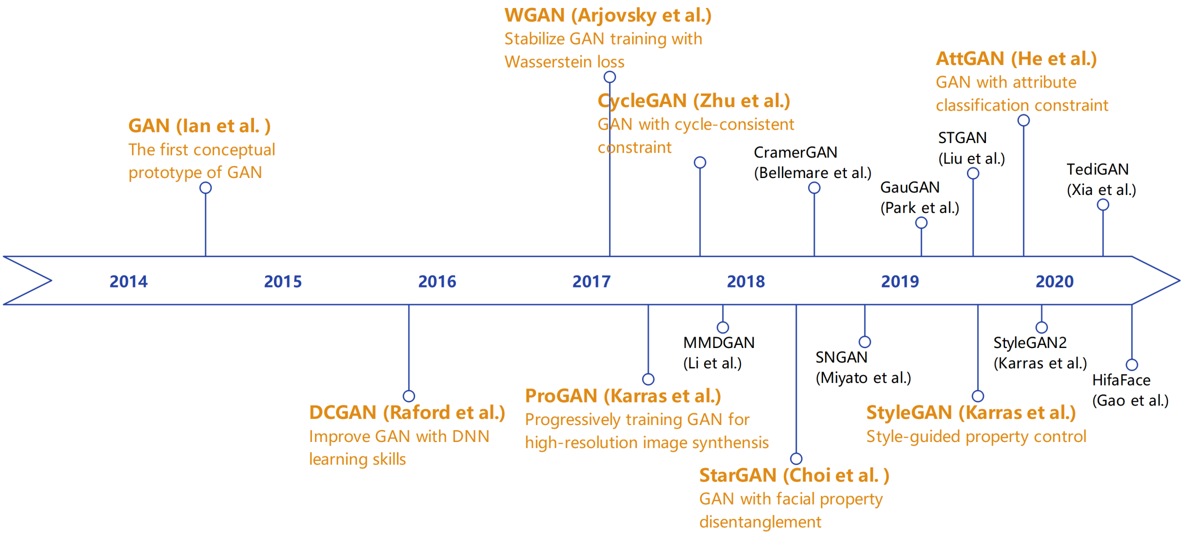


Figure 2.1: The technical evolution of the backend GAN techniques for deep image forgery, with highlighted milestones.

swap, and expression swap (also known as face reenactment) [63, 137]. In this thesis, we provide a simpler, technique-oriented taxonomy from the aspect of forensic focuses, which can facilitate the forensic and anti-forensic investigations, as follows:

- **End-to-end generated forgery**. This type of forgery is synthesized by end-to-end deep generative models, particularly GANs, without any other post-processing operations. This can be unconditional generations, where non-existent images are created from random noises, or conditional generations, where additional reference images and semantic labels may be needed, such as attribute manipulation and identity swap.

- **Phase-in forgery**. This type of forgery requires not only a deep generation process but also several post-processing operations, such as direction alignment, blending, rendering, denoising, etc., to enable a more smooth, photorealistic composition. This is a common procedure for identity swap and expression swap. Note that these post-processing operations can also be completed by deep generative models, which means the whole process will involve multiple independent deep generative models.

The reason we define such two types is that the forensic focuses on the two types of forgery are different since deep generative models and post-processing operations are likely to lead to significantly different forensic traces, artifacts, or noises. Despite the difference, one common thing is that both types employ deep generative models, particularly GANs, for a high-fidelity generation. Therefore, the detection of GAN-generated images is a fundamental countermeasure against deep image forgery and, thus, is the primary focus of this thesis. Figure 2.1 illustrates the technical evolution of the backend GAN techniques for deep image forgery, with highlighted milestones.

2.2 The forensic research

2.2.1 Forensics against conventional image forgery

In conventional image forensic tasks such as copy-move detection and camera source identification, A large number of forensic algorithms have been proposed, which can be divided into two categories: feature engineering-based algorithms and deep learning-based algorithms. Feature engineering-based algorithms focus on mining forgery features from image contents including color space, texture and shape. Deep learning-based forensics algorithms are mainly data-driven, i.e., learning from a large number of data to automatic recognize of image deep features. In this way, more abundant features can be extracted, and the forensic effect will also be improved significantly. Currently, deep learning-based forensics algorithms have been widely recognized and applied.

2.2.1.1 Feature engineering-based algorithms

Kot et al. [75] pointed out that feature engineering-based algorithms can be modeled as a classification problem in the field of machine learning. The complete forensic framework is shown in Figure 2.2, including the model training process and the forensic detection process. The model training process is to supervise the learning of images from N different sources, including the modules of feature selection, feature extraction and classifier training. First, selecting features according to the classification purpose. The candidate features include e.g. image color, texture and shape. Second, extracting features from the training dataset. Finally, using the extracted features to train the classifier. The commonly used machine learning classifiers include but not limit to Linear Discriminant Analysis (LDA), Support Vector Machine (SVM), and Naive Bayes (NB) and ensemble classifiers. The forensic detection process is to use the well trained classifier to identify the source of the unknown test image. According to the different features used by each algorithms, the feature engineering-based algorithms can be divided into

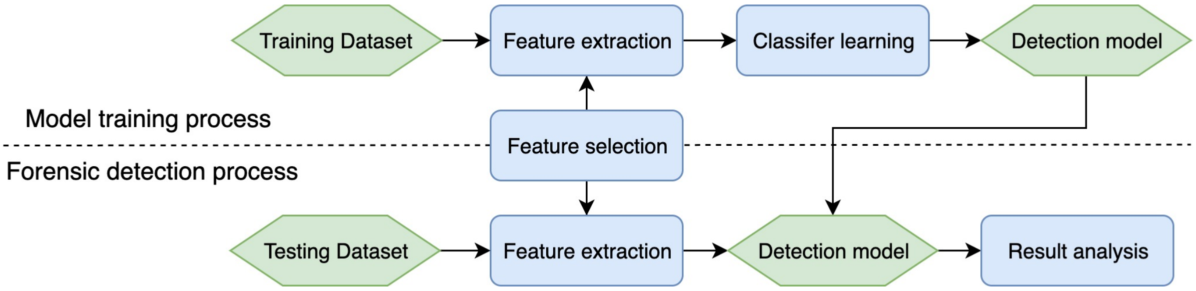


Figure 2.2: The common framework of feature engineering-based algorithms in conventional image forensic tasks.

three categories: statistical features-based forensics, texture features-based forensics, and imaging-based forensics.

Statistical features-based forensics Image statistical features refer to statistics that can be directly calculated from the image pixel values themselves, or from the image transform domain, such as pixel mean, variance, covariance, correlation coefficient, etc. [24, 38, 71, 97, 148]

Texture features-based forensics The image texture feature is represented by the gray distribution of pixels and surrounding neighborhoods, depicting the structural properties of the object surfaces that change slowly or periodically, such as image residuals, local binary patterns, contour wave decomposition, edge contours. [15, 16, 30, 84, 86, 87, 136]

Imaging-based forensics The imaging process of a digital image inside the camera is shown in Figure 2.3, the light reflected by the natural scene is focused by an optical lens, and first passes through a color filter array (CFA), so that each pixel has only one color component of red, green and blue. Then it is projected on the camera sensor to convert the optical signal into an electrical signal. Then the single-channel image is interpolated into RGB three-channel image by the CFA interpolation algorithm. Finally, a series of image processing algorithms inside the camera are performed sequentially, such as white balance, gamma correction, image sharpening and so on. In order to save camera memory, the processed digital images are usually stored after JPEG compression. Each operation described above will leave different traces in the image, and digital images from different sources will definitely produce different structural characteristics during the imaging process. By extracting and analyzing these characteristics, forensics can be effectively performed. [1, 8, 23, 28, 36, 95]

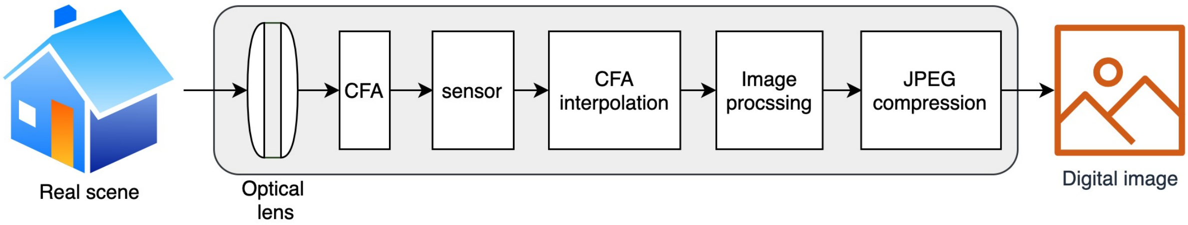


Figure 2.3: Imaging process of a digital image inside the camera

2.2.1.2 Deep learning-based algorithms

Unlike feature engineering-based algorithms, deep learning algorithms integrate feature extraction and feature classification into the same network structure. Compared with feature engineering-based algorithms that require sophisticated domain knowledge and more time consumption, deep learning methods benefit from data-driven representation learning and pattern recognition, which can learn more accurate and richer representations of implicit features at different levels autonomously, avoiding the limitations caused by hand-crafted features. Currently, Convolutional Neural Network (CNN)-based deep learning algorithms have gradually been widely adopted in the field of image source forensics. Barofio et al. [140] initially tried to use AlexNet [76] to identify the source of the device, which achieved the second price in the camera detection task as a part of the IEEE‚Äôs Signal Processing Camera identification Challenge. Bayar et al. [6, 7] proposed a self-learning restricted convolution structure to replace the high-pass filtering residual kernel and further improve the effectiveness of the model. Their method can detect multiple different image editing operations with up to 99.97% accuracy. Yang et al. [152] proposed an algorithm using a Laplace filter to enhance the noise signal introduced by the re-capturing. The proposed method achieved detection accuracy above 95% on four kinds of small-size image databases. Ye et al. [155] proposed a camera source forensics algorithm based on CNN, which adds high-pass filtering residual processing to the front of CNN to enhance the signal-to-noise ratio of the relevant signals. They proved that adding high-pass filter can achieve better results than no high-pass filter. Edmar et al. [119] proposed a transfer learning scheme to obtain an effective forensic model. The proposed method is able to distinguish computer-generated images or natural photos with an accuracy higher than 94%.

2.2.2 Forensics against deep image forgery

2.2.2.1 Normal detection methods

Normal detectors aim for high detection accuracy when identifying GAN-generated images in a known dataset. The recent technology can be divided into two mainstreams: Spatial-based detection and Frequency-based detection [105].

Spatial-based detection The image-domain detectors typically extract detectable traces from the image pixel information. Earlier research tended to train a DNN-based classifier to learn deep representative features from the images directly [99, 134]. In contrast, more current works chose to combine heuristic feature mining or a specific learning pattern with DNN classifiers to improve detection accuracy. For example, Nataraj et al. [108] and Barni et al. [5] proposed to use the co-occurrence matrices on different color channels for GAN-generated image detection. McCloskey et al. [103] revealed the difference between a GAN and a camera in forming color, resulting in a detection model based on the saturated and underexposed pixels. Hu et al. [51] show that GAN-synthesized faces are exposed with the inconsistent corneal specular highlights between two eyes, which can be exploited for detection. Some researchers pointed out that, similar to camera fingerprints, GANs will leave unique model fingerprints in the generated images, which can be leveraged to identify the source of the fake images. GAN fingerprints can be extracted as noise residuals from image pixels [100], or encoded from global image representations by a DNN [158]. There are also several works improving on the network architecture or learning pattern of the detector. For example, Marra et al. [101] designed an incremental learning framework to continuously evolve the detection models as new types of generated data appear. Jeon et al. [57] introduced a lightweight image-based self-attention module that can be integrated with pre-trained models for efficient fine-tuning with only small amount of data.

Frequency-based detection The frequency-domain detectors mine features from the frequency representations of images. These works were motivated by the observation that statistical frequency discrepancy exists between real and GAN-generated images and between images generated by two different GANs. Previous studies have explored the feasibility of exploiting the frequency discrepancy for forgery detection [32, 33, 39]. Images are normally transformed in a particular spectral representation, such as the Fourier spectrum and the Discrete Cosine Transform (DCT) coefficients, to enable the detector to learn the frequency discrepancy. [39] have pointed out that, a simple classifier, for example a shallow CNN, is able to achieve a high detection accuracy with the DCT spectral inputs. [33] and [32] proposed to simplify the input spectral representation by transforming the 2D Fast Fourier Transform (FFT) magnitude into 1D spectral profile for a lightweight detection. [91] found that the spectral discrepancy is more significant in the phase spectrum than in the amplitude spectrum, and accordingly proposed to combine image pixel and phase spectrum for detection. However, despite the frequency features being highly distinguishable, some recent studies further pointed out that these frequency features are unstable and easy to be perturbed [31, 32, 53, 64]. As a result, the detectors heavily relying on the frequency features are vulnerable and weakly generalized.

In addition, there is a branch of detectors that target sophisticated deep forgeries involving not just GANs but also post-processing procedures such as face alignment, rendering, and compression. The fundamental detection techniques are similar to the methods outlined above, also relying on features extracted from the image domain, frequency domain, or a combination of both. This paper is particularly concentrated on the problem of detecting end-to-end GAN-generated face forgeries.

2.2.2.2 Generalized and robust detection methods

Besides the demand for high detection accuracy, there forms a surge of interest in the generalization ability and robustness of GAN-generated image forgery detection. Existing studies mostly rely on developing more complex detectors or feature engineering techniques to learn more generalized and robust feature representations. Chai et al. [19] investigated what semantic properties of fake images make them detectable and identified what generalizes across different GANs. They also proposed a patch-based classifier with limited receptive fields to focus on local patches that are more generalized than global structure. Zhang et al. [159] proposed AutoGAN, a generator that can simulate the common spectral artifacts of GAN-generated images. A generalized detector can be trained using the simulated fake samples. Wang et al. [146] investigated the best combination of different augmentation strategies, such as JPEG compression and Gaussian blurring, for improving the generalizability and robustness of DNN detectors. Jeong et al. [58] proposed a bilateral high-pass filter-based preprocessing method for fake images, which strengthens the representations of common frequency-domain artifacts shared by different GANs. Bui et al. [14] proposed a representation mix-up training strategy and a novel loss to make the detector invariant to semantic changes and improve its robustness to common image transformations changing in quality, resolution, shape, etc. Jeong et al. [59] designed a frequency-level adversarial perturbation (FLAP) learning framework that can suppress the unstable GAN-specific frequency artifacts in the training samples during the training of the detector. He et al. [49] proposed a re-synthesis residual (RSR)-based detection method. A re-synthesis model pre-trained with real images is applied to real and fake images to obtain distinguishable re-synthesis residuals, which contain robust features.

2.2.2.3 Proactive detection methods

The spatial-based and frequency-based detection methods discussed above are all passive detection methods. There are also a few studies investigated proactive detection methods which involve a watermarking mechanism. An invisible watermark that is fragile to forgery manipulation is initially injected into the original image before sharing. A decoder is equipped to verify the integrity of the watermark to determine whether or not the image has been forged. For example, Yang et al. [154] and Neekhara et al. [110] proposed deep-learning-based watermarking methods which are robust to normal image post-processing but fragile to deepfake manipulation. Wang et al. [145] devised FaceTagger, a simple yet effective encoder and decoder design along with channel coding to embed recoverable message to the facial image to track DeepFake provenance. Zhao et al. [161] proposed to embed watermarks as anti-Deepfake labels into the facial identity features disentangled with attribute features to further improve the robustness of watermarks against conventional image modifications.

2.3 The anti-forensic research

Anti-forensic research in this realm aims to understand the robustness issues of a GAN-generated forgery detector by actively exposing the target detector to possible attacks. In most cases, a security analysis is required to assess the vulnerability of the target detectors, where different scenarios such as black-box and white-box tests and some novel attack methods are involved.

2.3.1 Adversarial attacks

The majority of anti-forensic research prefers adversarial attacks for security analysis since most forgery detectors are machine learning models. A successful adversarial example is created by embedding imperceptible noise perturbations into the original fake image to evade the detector. Commonly, the noises are crafted based on the learning gradients of machine learning detectors. Thus, adversarial attacks are a powerful white-box attack and have potential black-box transferability. Several classic adversarial attack methods, such as Fast Gradient Sign Method (FGSM) [46], iterative FGSM [77], Carlini and Wagner l_2-norm Attack [18], DeepFool [107] and Projected Gradient Descent (PGD) [98], are explored to attack GAN-generated forgery detectors in both white- and black-box scenarios [4, 17, 34, 42, 55, 109, 160]. Liao et al. [88] improved the efficacy of adversarial attack by adding perturbations to the key regions of the forged samples instead of the entire image. Wang et al. [147] found that the addition of adversarial noise to a transformed color space mitigates the perceptual degradation of the raw forged image.

2.3.2 Reconstruction attacks

In addition to adversarial attacks, some recent studies have designed novel attacks specific to GAN-generated forgery detectors. Most of them require a generative reconstruction process to re-synthesize the forgery samples and alter their forgery traces in order to reduce the degree of fakeness. For example, Huang et al. [53] proposed FakePolisher, a dictionary learning-based reconstruction model, to project DeepFake images onto the manifold learned from real images to reduce their spectral artifacts. Neves et al. [111] proposed GANprintR, a convolutional autoencoder that learns the reconstruction-related representation from real images. Then, the learned autoencoder can be used to remove the GAN fingerprints contained in the forged images. Ding et al. [29] proposed an adversarial learning framework that re-synthesizes face-swapping images with narrowing down the distributional gap between real and fake faces. Peng et al. [114] proposed a bidirectional conversion between GAN-generated and natural facial images based on a GAN composed of noise encoding and content encoding for anti-forensics. Liu et al. [89] proposed the trace removal attack (TR-Net), an adversarial learning network that can simultaneously remove multiple forgery traces from the fake images to evade detection.

2.4 Frequency analysis of DNNs' behavior

Some pioneering studies have shed light on the behaviors of generic DNNs in learning natural images through frequency analysis. For example, Xu et al. [151], Wang et al. [144] and Rahaman et al. [118] have pointed out that DNNs have a bias in learning information at different frequency bands, and investigated the influence of the frequency bias in generalizing to out-of-distribution images. Yin et al. [156] explored DNNs' robustness to adversarial attacks from a frequency perspective. Despite these existing advances in natural image classification, the frequency-level understanding of detecting machine-generated images has yet to be fully explored. The frequency patterns of GAN-generated images are significantly different from those of natural images, making it impractical to apply previous findings directly. Moreover, the connection between the generalization and robustness of GAN-generated forgery detectors is unclear.

CHAPTER 3

PRELIMINARY

In this chapter, we will briefly introduce some preliminary knowledge of this thesis, including the basic structure and workflow of deep learning classifiers, deep generative models such as autoencoder and generative adversarial network (GAN), adversarial attack, image frequency-domain transformation. We will also introduce the formulation of DeepFake forensic and anti-forensic problems.

3.1 Deep learning classifiers

Nowadays, deep image forgery detectors are mostly deep learning classifiers. A deep learning classifier refers to a deep neural network (DNN) comprising multiple layers with a large number of computational neurons and nonlinear activation functions [79]. The workflow of a typical deep learning classifier includes two phases: the training phase and the inference phase. In the training phase, the parameters of the DNN are updated continuously through iterative forward and backward propagations. Specifically, given a input space $\mathscr{X}$ and a label space $\mathscr{Y}$, the DNN f_θ with parameters θ are expected to minimize the loss function $\mathscr{L}$ on the training dataset $(\mathscr{X},\mathscr{Y})$, which can be defined as:

$$\underset{\theta}{\arg\min} \sum_{x_i \in \mathscr{X}, y_i \in \mathscr{Y}} \mathscr{L}(f_\theta(x_i), y_i),$$

where f_θ is the DNN model to be trained; x_i is a training sample, and y_i and $f_\theta(x_i)$ are the corresponding ground-truth label and the predicted label, respectively. In the inference phase, the optimal models $f_\theta^\star$ with fixed optimal parameters $\theta^\star$ are applied to

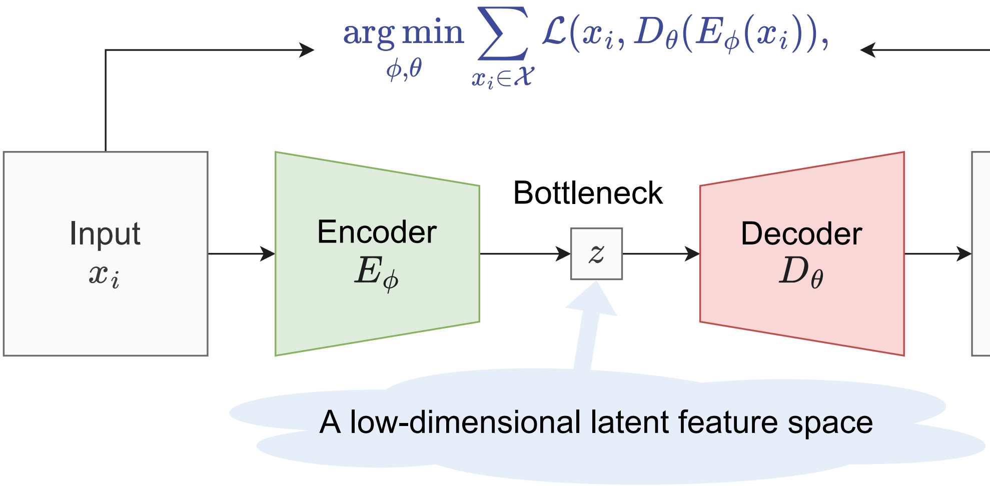

$\arg\min_{\phi,\theta} \sum_{x_i \in \mathcal{X}} \mathcal{L}(x_i, D_\theta(E_\phi(x_i))),$
Input
x_i
Encoder
E_ϕ
Bottleneck
z
Decoder
D_θ
Output
x_i'
A low-dimensional latent feature space

Figure 3.1: The sketch of a basic autoencoder.

provide decisions on test samples that are not included in the training dataset. Given an unseen input x_j, its predicted label can be computed through a single-step forwarding $y_j = f_\theta^\star(x_j)$.

3.2 Deep generative models and adversarial learning

There are two main types of deep generative models adopted for creating deep image forgeries, namely autoencoder and generative adversarial network.

Autoencoder is an unsupervised artificial neural network that learns to translate the original high-dimension input into the latent low-dimensional code, then learns to recover the data back from the encoded representation [50]. A basic autoencoder consists of two parametrized networks, the encoder $E_\phi : \mathscr{X} \to \mathscr{Z}$ parametrized by ϕ and the decoder $D_\theta : \mathscr{Z} \to \mathscr{X}$ parametrized by θ, where $\mathscr{X}$ and $\mathscr{Z}$ indicate the data space and the latent feature space, respectively. The encoder E_ϕ and decoder D_θ are in together trained in a reconstruction task, which can be denoted as:

$$\arg\min_{\phi,\theta} \sum_{x_i \in \mathscr{X}} \mathscr{L}(x_i, D_\theta(E_\phi(x_i))),$$

Figure 3.1 shows the sketch of an autoencoder.

GAN is another unsupervised artificial neural network consists of two networks: a generator G and a discriminator D [44]. In an image generation task, G learns to map a target distribution p_{data} of the given images $\mathscr{X}$ from a noise space $\mathscr{Z}$, while D distinguishes the candidates produced by the generator from the true distribution. This

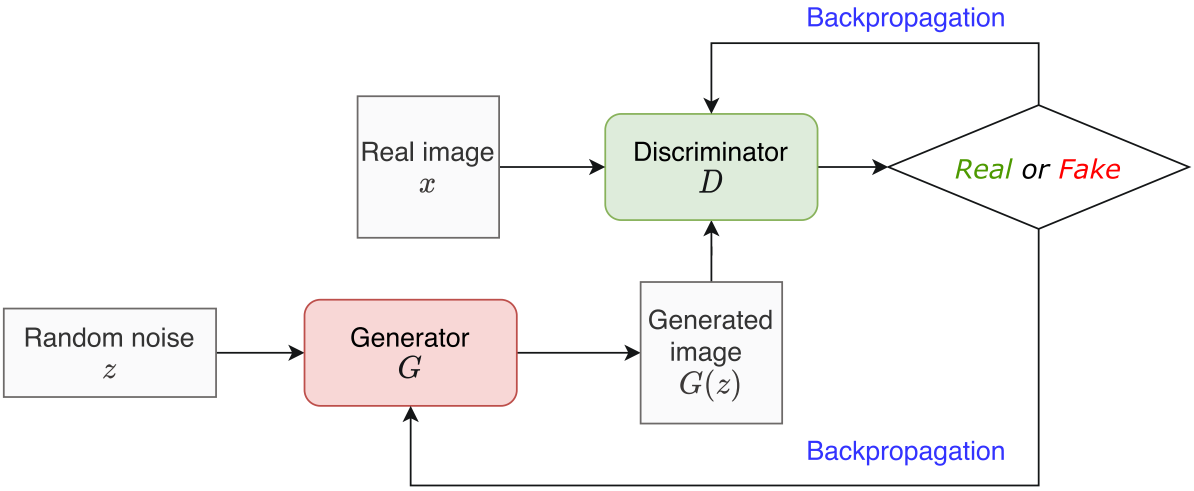


Figure 3.2: The sketch of a basic generative adversarial network.

learning process is constructed as a dynamic contest, where a min-max game between between G and D is played. The learning pattern is also known as adversarial learning. Learning alternates between G and D by optimising the following adversarial loss function:

$$\min_{G}\max_{D}\mathscr{L}_{adv}(D,G)=\mathbb{E}_{x_i\sim p_{\text{data}}}[\log D(x_i)]+\mathbb{E}_{z\sim p_z}[\log(1-D(G(z)))], \tag{3.1}$$

where $G(\cdot)$ and $D(\cdot)$ are the outputs of G and D, respectively. I_{real} is the real image, z is the random input seed of G, p_{data} and p_z are the distributions of $\mathscr{X}$ and $\mathscr{Z}$, respectively. When the training converges to the point where D is successfully 'fooled' by the images G has generated, i.e., $D(G(z))$ reaches approximately 0.5, G is able to generate images within the target distribution, i.e., $G(z)\sim p_{\text{data}}$. Then the well-trained G can be applied as the desired generative model for future image generation. Figure 3.2 shows the sketch of a GAN.

In addition, if an extra control is imposed over the modes of the data to be generated, such as adding an attribute label y as prior guidance for supervision, a conditional GAN will be created then.

3.3 Adversarial attack

Adversarial attacks are a common type of attack against machine learning classifiers and have been exploited to attack deep learning-based forgery detectors in previous anti-forensic research. In an adversarial attack, the attacker often crafts adversarial examples originating from the original forged images. Szegedy et al. [133] first introduced the concept of adversarial examples, which can mislead the target machine learning classifiers with a high success rate in the inference phase. The primitive method is to search for the minimally distorted adversarial examples with the targeted label through Equation 3.2.

$$\min \left\| x' - x \right\|_2^2 \quad \text{s.t.} \quad f(x') = t \quad \text{and} \quad x' \in [0,1]^m \tag{3.2}$$

Through this equation, attackers can find the closest x' which has a minimal distance with benign sample x by minimizing $\left\| x' - x \right\|_2^2$ and would be misclassified as targeted label t by the condition of $f(x') = t$. This problem can be further formulated to the optimization problem in Equation 3.3:

$$\min c \left\| x' - x \right\|_2^2 + \mathscr{L}(f(x'), t) \quad \text{s.t.} \quad x' \in [0,1]^m \tag{3.3}$$

3.4 Image frequency-domain transformation

Some of our proposed methods require transforming images from the spatial domain to the frequency domain. The spatial domain contains visual information and is normally represented as 8-bit pixels in RGB color mode (i.e., three values within the range [0,255]). The frequency domain can be depicted as the frequency spectrum transformed from the spatial information by discrete Fourier transformation (DFT) or discrete cosine transformation (DCT) [43]. Given an image $I \in \mathbb{R}^{M \times N}$, the DFT is computed via a two dimensional transformation that maps the value of each pixel to a frequency value $\mathscr{F}(u,v)$:

$$\begin{gathered}\mathscr{F}(I)(u,v) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} I(m,n) \cdot e^{-2\pi i \cdot \frac{um}{M}} e^{-2\pi i \cdot \frac{vn}{N}} \\ \text{for } m = 0,1,\ldots,M-1, \quad n = 0,1,\ldots,N-1\end{gathered} \tag{3.4}$$

In practice, DCT, a variant of DFT, is more widely used for frequency transformation [43] given it compacts the real part of DFT information and avoids the imaginary part. The DCT for an image is normally computed as:

$$\mathscr{D}(I)(u,v) = C(u)C(v) \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} I(m,n) \cdot \cos\left[\frac{\pi u}{M}\left(m + \frac{1}{2}\right)\right] \cos\left[\frac{\pi v}{N}\left(n + \frac{1}{2}\right)\right] \tag{3.5}$$

where for $\forall u = 0, 1, \ldots, N-1$, $C(0) = \sqrt{1/N}$ and $C(u > 0) = \sqrt{2/N}$ to ensure the orthonormality. The DCT spectrum is typically depicted as a coefficient heatmap, where the magnitude of each coefficient measures the contribution of the corresponding frequency component to the overall image. The top left area of the heatmap indicates lower frequencies, which represent the major visual information of an image, while the right bottom area corresponds to higher frequencies, which reflect the spatial information associated with edges, structural details, and noises.

3.5 Problem formulation

The DeepFake forensic is commonly formulated as a real/fake detection problem. Formally, let $\mathbb{D} = \{\mathbb{I}^+, \mathbb{I}^-\}$ be a dataset consisting of real images $\mathbb{I}^+$ and fake images $\mathbb{I}^-$. A binary machine learning classifier $\mathscr{C}$ trained with $\mathbb{D}$ is able to predict the correct real/fake labels for a given unseen test sample, i.e.,:

$$p\left(\mathscr{C}(I^+_{test}) = \text{‘real’}\right) \approx 1, \quad \text{and} \quad p\left(\mathscr{C}(I^-_{test}) = \text{‘fake’}\right) \approx 1 \tag{3.6}$$

Oppositely, in DeepFake anti-forensic tasks, the attacker aims to develop an attack model $\mathscr{A}$ to perturb the original DeepFake samples, resulting in attack samples that can evade the target detector $\mathscr{C}_t$. In addition, the perturbation is desired to be imperceptible, which means an attack sample is visually indistinguishable from its source DeepFake sample. Specifically, given an arbitrary DeepFake sample I^-_{test}, the anti-forensic goal can be described as:

$$p\left(\mathscr{C}_t(\mathscr{A}(I^-_{test})) \neq \text{‘fake’}\right) \approx 1, \quad \text{and} \quad d\left(\mathscr{A}(I^-_{test}), I^-_{test}\right) \leq \epsilon \tag{3.7}$$

where $d(\cdot,\cdot)$ is a visual distance measurement.

Part II

Investigation on forensic techniques

Chapter 4

Task-specific forensics by multi-level GAN fingerprinting

Recent advancements in image generation by Generative Adversarial Networks (GANs) have introduced new security challenges in image forensics. One promising solution involves detecting tell-tale marks left by GANs in images, commonly referred to as GAN fingerprints. Existing methods for GAN fingerprinting often focus on a single forensics task, relying on noisy post-processing or exhibiting pixel bias in their extraction methods. In this chapter, we delve into the next evolution of image forensics in this battleground, specifically identifying and applying GAN fingerprints for practical forensics against GAN-generated images. We are the first to focus on the problem of different distinguishability levels required by different image forensics tasks and propose a task-specific GAN fingerprinting framework to deal with the problem. Our research began by exploring GAN fingerprint dependency across two image signal domains, which revealed two distinct levels of fingerprints, including the instance-level fingerprint in the spatial domain and the architecture-level fingerprint in the frequency domain. From this finding, we proposed an explicit decoupling representation framework to separate and extract the two types of GAN fingerprints from different domains. An adversarial data augmentation strategy plus a transformation-invariant loss is added to the extraction pipeline to enhance the robustness of fingerprints to image perturbations. Then we elaborated on how to leverage the two types of GAN fingerprints to perform task-specific fingerprinting in three typical forensics tasks, including fake image detection, model intellectual property protection, and fingerprinting attack and defence. Extensive experiments have verified our dependency analysis, and the effectiveness and robustness of the proposed fingerprint extraction framework were well demonstrated in a benchmark test. The task-specific fingerprinting methods were tested in real-world or simulation-based scenarios.

4.1 Background

Like many other areas of computer science, deep learning has been responsible for significant advancements in the field of image and signal processing [79]. Among these, the generative adversarial network (GAN) [44], a typical deep generative model, has seen substantial development in automated image generation, synthesis, and editing. However, alongside this success, new privacy and security risks are rising continuously. For example, DeepFake, an emerging face forgery technique driven by GANs, can seamlessly synthesize fake image or video records according to victims' faces [138]. On the other hand, nowadays, both the well-trained GAN models and the images created by these GANs are increasingly deemed as valuable digital assets given the significant costs in training a GAN model [158]. These types of intellectual property (IP) needs to be carefully protected from theft or plagiarism.

Image forensics, a cluster of techniques that can determine whether the image content is authentic or modified, or identify the true source of an image, helps to alleviate these concerns [40]. Despite most current GAN-generated image forensics methods considering different tasks such as DeepFake image detection and model IP protection individually, an intriguing question is raised: Is there a universal mechanism to deal with different forensics tasks simultaneously? Model fingerprinting, which detects the intrinsic clues already in the image that might hint at its source model [96], offers a promising direction for this problem. Unlike conventional camera fingerprints that have been extensively explored, the research on GAN fingerprints is just at its beginning.

Currently, there are only a few GAN fingerprinting studies. For example, Marra et al. [100] estimated GAN fingerprints as average image noise residuals and Yu et al. [158] used Deep convolutional neural network (CNN) features to represent GAN fingerprints. These pioneering studies tend to focus on how to extract GAN fingerprints only, lacking further exploration of how to apply the extracted fingerprints in different forensic tasks. More critically, none considered the problem of different distinguishability levels required by different image forensics tasks, which is a crucial concern in practice. A fingerprinting method that can flexibly operate at different distinguishability levels specific to different

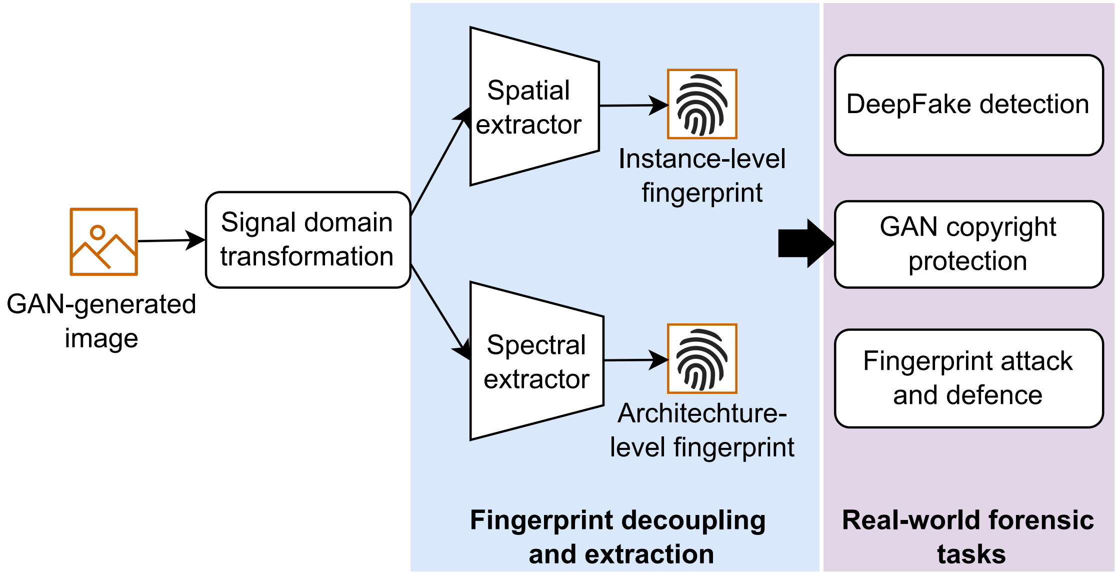

GAN-generated image
Signal domain transformation
Spatial extractor
Instance-level fingerprint
Spectral extractor
Architechture-level fingerprint
Fingerprint decoupling and extraction
DeepFake detection
GAN copyright protection
Fingerprint attack and defence
Real-world forensic tasks

Figure 4.1: Overview of multi-level GAN fingerprinting for task-specific forensics.

tasks can be highly preferable (see detailed analysis in Section 4.2 and 4.4). Meanwhile, existing fingerprint extraction methods both involve noisy post-processing operations for explicit fingerprint representation, which can compromise the visualization performance. Another issue is that previous GAN fingerprints were extracted directly from the pixel information, which are easily biased by visual content and may be vulnerable to common image perturbations such as cropping or compressing.

In this chapter, we present a novel multi-level model fingerprinting method for task-specific forensics on GAN-generated images and make a step further in this fledgling area by addressing the above issues. Our work revolves around solving three key problems:

- **RQ1.** How to extract GAN fingerprints with different levels of distinguishability?
- **RQ2.** How to make results of the GAN fingerprint extraction process more effective and robust?
- **RQ3.** How to perform task-specific GAN fingerprinting in practical image forensics tasks?

For **RQ1** and **RQ2**, we first demonstrate, both theoretically and empirically, GAN fingerprints have different dependencies in the spatial and frequency domains, resulting in fingerprints with *architecture-level* distinguishability and *instance-level* distinguishability (In this chapter, architecture-level distinguishability means the fingerprints can differentiate two GANs with different network architectures (e.g., ProGAN [66] versus

StarGAN [26]); Instance-level distinguishability means the fingerprints can differentiate any two GAN instances resulted from two individual training processes, irrespective of their architectural similarity). Then, based on the analysis, a decoupling representation framework is proposed to extract the two levels of fingerprints separately from the two signal domains to provide different levels of distinguishability required in real-world tasks. The framework involves an end-to-end mechanism for explicit fingerprint representation to avoid any noisy post-processing operations. An adversarial data augmentation strategy combining a transformation-invariant loss is additionally proposed to enhance the robustness of fingerprints to common image perturbations. For **RQ3**, we elaborate three typical downstream image forensics tasks including fake image detection, model IP protection, and fingerprinting attack and defence, and show how to leverage the two levels of GAN fingerprints to perform task-specific fingerprinting. The overview of our work is shown in Figure 4.1.

To the best of our knowledge, we are the first to consider the distinguishability level required in practical GAN image forensics tasks, and propose a task-specific model fingerprinting method which supports flexible operation at different distinguishability levels. Our contributions are as follows:

- We performed an in-depth analysis of GAN fingerprint dependency, providing theoretical and empirical evidence on the existence of architecture-level and instance-level GAN fingerprints in the spatial and frequency domains respectively (**RQ1**).
- We developed an explicit decoupling representation framework that can separate and extract the two levels of GAN fingerprints from the two domains (**RQ2**).
- We explored the applicability and usefulness of the two levels of GAN fingerprints for task-specific fingerprinting in multiple image forensics tasks (**RQ3**).

4.2 GAN fingerprint dependency

We undertook an analysis of the root dependencies of GAN fingerprints, which have not yet been fully revealed in previous GAN fingerprinting studies. Importantly, we theorised the differences between real images and GAN-generated images, as well as between images generated by two GANs in the spatial domain and the frequency domain, respectively. The experiments in Section 4.5.2 provide empirical supports to the analysis. Through the analysis, we identify a cause of GAN fingerprints for each domain, sparking the idea that decoupling the fingerprint representation via domain transformation might offer different distinguishability levels required in practice.

4.2.1 Dependence on GAN architecture

GANs are known to leave distinguishable quasi-periodic spectral artifacts in the high frequency domain[39, 146, 159], which are supposed to be one possible origin of GAN fingerprints. For example, Figure 4.2.a-b shows the average discrete Fourier transformation (DFT) DFT spectra of $1,000$ real images and of $1,000$ images generated by StarGAN [26]. The quasi-periodic spectral artifact in the StarGAN spectrum is obvious.

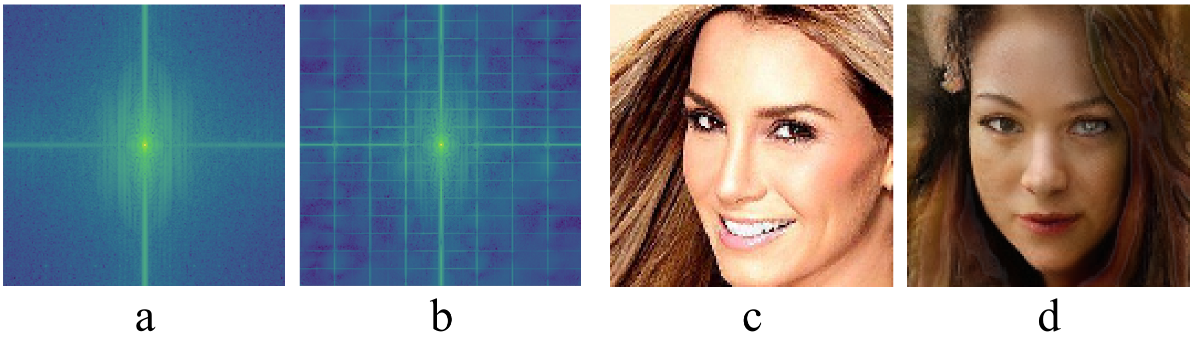


Figure 4.2: The differences between real images and GAN-generated images in the frequency and spatial domains.

We theorised that the GAN fingerprints derived from spectral artifacts are GAN-architecture-dependent, more specifically, being associated with the upsampling units sitting in the generator G of GAN. The upsampling units are necessary elements in GAN's generator responsible for constructing higher-dimensional data/features (e.g., an image I) from low-dimensional ones (e.g., an input seed z). There are two common types of upsampling units in GAN, namely interpolation upsampling and transposed convolution. As shown in Figure 4.3, both can be formulated as a similar pipeline [159]: the input tensor is first upscaled by zero-padding interpolation that pads the raw values with zeros (indicated as white grids) and then convolved with a filter kernel (indicated as grey grids). The only difference is that in interpolation upsampling the convolution kernel is fixed, while in transposed convolution the convolution kernel is learnable.

To demonstrate the dependence of GAN fingerprints on upsampling unit, we need to reveal the relations between the spectral artifacts and the upsampling units. Zhang et al. [159] proved it in a special two-time upsampling case, and we extend their proof to a general upsampling case with an upscale factor of s. DFT is used to compute the

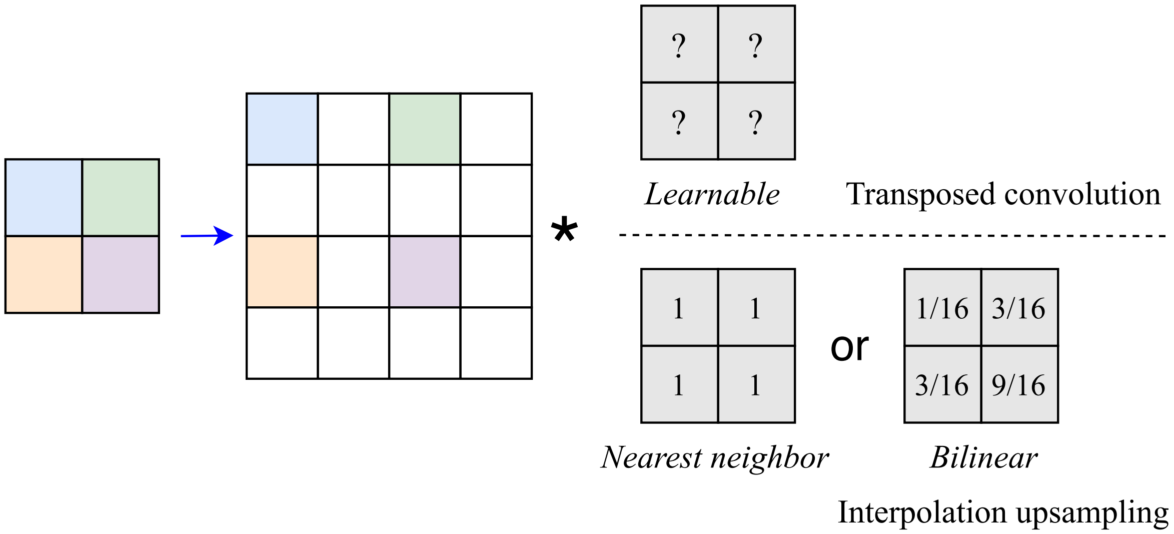


Figure 4.3: Schematic of two upsampling units: transposed convolution (up) and interpolation upsampling (down). In this example a 2 × 2 feature is upsampled to 4 × 4.

natural frequency spectrum. We first consider the common zero padding interpolation process in the above-mentioned upsampling pipeline in the one-dimensional case. Let $f(n), n = 0, 1, ..., N-1$ be a one-dimensional signal and its DFT is

$$\mathscr{F}(u) = \sum_{n=0}^{N-1} f(n) \exp\left(-2\pi i \frac{un}{N}\right) \quad u = 0, 1, ..., N-1 \tag{4.1}$$

By zero-padding with a scaling factor s, the spatial signal is expanded to an sN-point signal $f'(m), m = 0, 1, ..., sN-1$:

$$f'(m) = \begin{cases} f(m/s) & m = sn \\ 0 & m \neq sn \end{cases} \tag{4.2}$$

From the standpoint of sampling theory, Eq. 4.2 can be recasted as:

$$f'(m) = \sum_{t=-\infty}^{\infty} f(\frac{m}{s}) \cdot \delta(m - st) \quad m = 0, 1, ..., sN-1 \tag{4.3}$$

where $\delta(\cdot)$ indicates the Dirac impulse comb.

The DFT of $f(m/s)$ is computed first, denoted as $\mathscr{F}'(v)$:

$$\mathscr{F}'(v) = \sum_{m=0}^{sN-1} f(\frac{m}{s}) \exp\left(-2\pi i \frac{vm}{N}\right) \tag{4.4}$$

Let $m' = m/s$, and we have:

$$\mathscr{F}'(v) = \sum_{m'=0}^{N-1} f(m') \exp\left(-2\pi i \frac{svm'}{N}\right) = \mathscr{F}(sv) \tag{4.5}$$

The final equality indicates that the frequency spectrum of $f(m/s)$ is a scaled replica of the frequency spectrum of $f(n)$ by a factor of $1/s$ within the region of $[0, \frac{N-1}{s}]$. Now let us consider the DFT of $f'(m)$, denoted as $\widehat{\mathscr{F}(v)}$. Assuming a periodic signal and applying the convolution theorem in [43] to Eq. 4.3, we have:

$$
\begin{aligned}
\widehat{\mathscr{F}(v)} &= \sum_{\tau=-\infty}^{\infty} \mathscr{F}'(\tau) \cdot \frac{1}{s} \sum_{t=-\infty}^{\infty} \delta(v - \tau - \frac{t}{s}) \\
&= \frac{1}{s} \sum_{t=-\infty}^{\infty} \sum_{\tau=-\infty}^{\infty} \mathscr{F}'(\tau) \delta(v - \tau - \frac{t}{s}) \\
&= \frac{1}{s} \sum_{t=-\infty}^{\infty} \mathscr{F}'(v - \frac{t}{s}) = \frac{1}{s} \sum_{t=-\infty}^{\infty} \mathscr{F}(sv - t)
\end{aligned}
\tag{4.6}
$$

which is equivalent to a periodic sampling of the scaled replicas of the frequency spectrum of $f(n)$ in the new frequency spectrum. Therefore, in the one-dimensional case, all frequencies beyond $(N-1)/s$ will be potential artifacts. When generalised to the two-dimensional case, the replicas in the horizontal and vertical directions of the high-frequency space will be superimposed onto each other, resulting in periodic artifacts. Then, with both transposed convolution upsampling and interpolation upsampling, the zero-padding interpolation is connected to a convolution kernel. However, neither a learnable kernel nor a fixed one can act as an ideal filter to completely eliminate these high-frequency artifacts. Thus, the final result is the periodic artifacts in the spectrum of the generated image, yielding an architecture-dependent GAN fingerprint in the frequency domain.

4.2.2 Dependence on training randomness

We now demonstrate the existence of another GAN fingerprint which is independent of the GAN's architecture information. We suppose this kind of GAN fingerprint is associated with the randomness in its own training process.

Essentially, a GAN model is a machine learning model that learns to approximate a target distribution p_{data} from real-world images. According to the "No Free Lunch (NFL)" theorem [150], no prior distinction exists between any two models. Therefore, for a given task, if the generators G in all the GANs were able to converge on the perfect optimum $G^\star$, it would be possible to arrive at $\forall G, p_{\text{data}} \equiv p_{G^\star}$. This equivalence suggests that, ideally, the images generated by different GANs should each be identical to the real image without any difference. However, in reality, training a GAN model is often subject to the limitations of dataset and model. For example, it is highly improbable that the sampled data will cover the full target distribution p_{data}, and the model's capacity will inevitably be bounded by factors such as data noise and neuron amount. As a result, the optimisation process often falls into a random local optimum, leading to random bias in different GAN instances.

A typical example is the visual defect phenomenon. Some sub-optimal GAN instances will leave visual defects in the spatial content of their generated images, which may present in possible forms of distortion, deformity, dissonance, incompleteness or inconsistency [54, 163]. These defects can occur with any of the visual information from low-level regional details or noise patterns to high-level semantic content. Figure 4.2.c-d show a real image and a ProGAN-generated image with distortions in the hair and background and inconsistency in the two eyes. Previous studies have shown that distinguishable features can be extracted from these visual defects (even the defects are nearly invisible to human eyes) to identify GAN sources [54, 163]. The findings imply that unique GAN fingerprints could be associated with the visual defects which are a result of the training randomness.

Generally, the training randomness of a GAN instance is dependent with the following specific factors: 1) the semantic information of the training data of GAN; 2) the differences in the training settings, such as the sample size, the convergence point, the initialisation parameters, etc. Accordingly, the representation of the GAN fingerprints derived from the random bias should be directly influenced by these factors.

4.2.3 Architecture-level and instance-level GAN fingerprints

The multi-level distinguishability problem. In practice, model fingerprints for different image forensics tasks often necessitate different levels of distinguishability. For example, a DeepFake creator is likely to update their models regularly with new data while keeping the model architecture invariant. As a countermeasure, in DeepFake image detection, the fingerprints identified in images from the same DeepFake source should generalize well among GANs with the same architecture, and insensitive to model parameter changes. By comparison, in a GAN IP protection task, the model owner may desire to build a unique signature for the GAN instance in hand. Therefore, the model fingerprint for this use should ensure an exclusive directing to the current source GAN instance, i.e., any two GAN instances resulted from different training processes should have distinct fingerprints, even if their architectures are identical.

Through the analysis of GAN fingerprint dependency, we identified upsampling bias in the frequency domain and random bias in the spatial domain as two origins of GAN fingerprints. The GAN fingerprints depend on the two biases can exhibit different levels of distinguishability that we desire: Upsampling bias is closely associated with GAN's architecture information. Hence, the resulting fingerprints should be distinguishable for GANs with different architectures, while share a similar pattern for GANs with the same architecture. We refer to these fingerprints as **architecture-level**. In comparison, random bias is highly related to a GAN's training process and independent with its architecture. Hence, the resulting fingerprint corresponding to any one GAN instance should be unique. We refer to these fingerprints as **instance-level**. The architecture-level and instance-level GAN fingerprints are perfectly applicable to resolving the multi-level distinguishability problem. Another benefit is that the architecture-level fingerprints are more applicable to black-box fingerprinting scenarios since it can be obtained from publicly-available GAN substitutes whose architectures are the same as the target one, while the instance-level ones requiring the knowledge of a GAN instance's training details are more applicable to white-box scenarios.

4.3 GAN fingerprint extraction

With the insights into GAN fingerprint dependency, the next challenge is how to decouple and extract the architecture-level and instance-level GAN fingerprints. Taking practical applicability into consideration, an effective extraction method should:

- be able to decouple instance-level GAN fingerprints in the spatial domain and architecture-level GAN fingerprints in the frequency domain;
- represent GAN fingerprints in a 2D view that is human-interpretable in line with the prior GAN fingerprinting studies [100, 158], but in an end-to-end way without any extra post-processing; and
- be robust against common image processing approaches.

The above requirements are the objectives of designing our fingerprint extraction method.

4.3.1 Fingerprint decoupling

GAN fingerprint is a unique and stable feature that can be extracted from the spatial domain or the frequency domain of a GAN-generated I to indicate its origin. Thus, the extraction can be formulated as feature representation learned in an image attribution task [158]. The successful attribution of a GAN-generated image I_g is defined as an exclusive match between the image and the GAN instance $\mathscr{G}_i$ that originally generated it, i.e., $I_g \to \mathscr{G}_i$. A practical assumption is that the candidate sources for a test sample I belong to a finite set $\mathbb{G} = \{\mathscr{G}_0, \mathscr{G}_1, \mathscr{G}_2, ..., \mathscr{G}_n\}$. For ease, we use $\mathscr{G}_0$ to indicate the sample is a real-world image, and $\mathscr{G}_i$ $(i \neq 0)$ is a GAN instance. The problem can be formulated as a multi-classification task where the distinguishable patterns are recognised to classify the image to the correct source.

The GAN fingerprints can be represented implicitly as the distinguishable and unique features encoded from the given image in the latent feature space [158]. And the decoupling of multi-level distinguishability can be performed in the latent feature space accordingly. To this end, we designed a CNN encoder, which learns the features in the above classification task. The CNN involves four convolution layers with a 3 × 3 kernel size, two max pooling layers with a 2 × 2 kernel size, a flatten layer and a dense connection layer. Table 4.1 shows the specifications of the encoder.

Table 4.1: The details of the CNN-based fingerprint encoder.

Layer name	**Kernel size&depth**
Input	-
Convolution	(3, 3, 3)
Convolution	(3, 3, 8)
Max Pooling	(2, 2)
Convolution	(3, 3, 16)
Max Pooling	(2, 2)
Convolution	(3, 3, 32)
Flatten	-
Dense	-

4.3.1.1 Instance-level fingerprint learning

According to the root cause analysis in Section 4.2, the instance-level fingerprints are derived from the random bias, which can occur in any visual content of the image. Therefore, learning such fingerprints requires that all the information in the spatial domain be taken into account. Hence, the CNN encoder is trained on image-source pairs $\{(I, \mathscr{G})\}$ in a supervised manner, where $I \in \mathbb{R}^{N\times N}$ is the original RGB image sampled from the training dataset and $\mathscr{G} \in \mathbb{G}$ denotes the ground-truth source. This approach ensures the CNN learns the overall spatial information directly from the raw RGB pixels. Figure 4.4 shows the learning workflow of the instance-level fingerprint encoder (indicated by the red arrows) and a schematic overview of decoupling in the latent feature space.

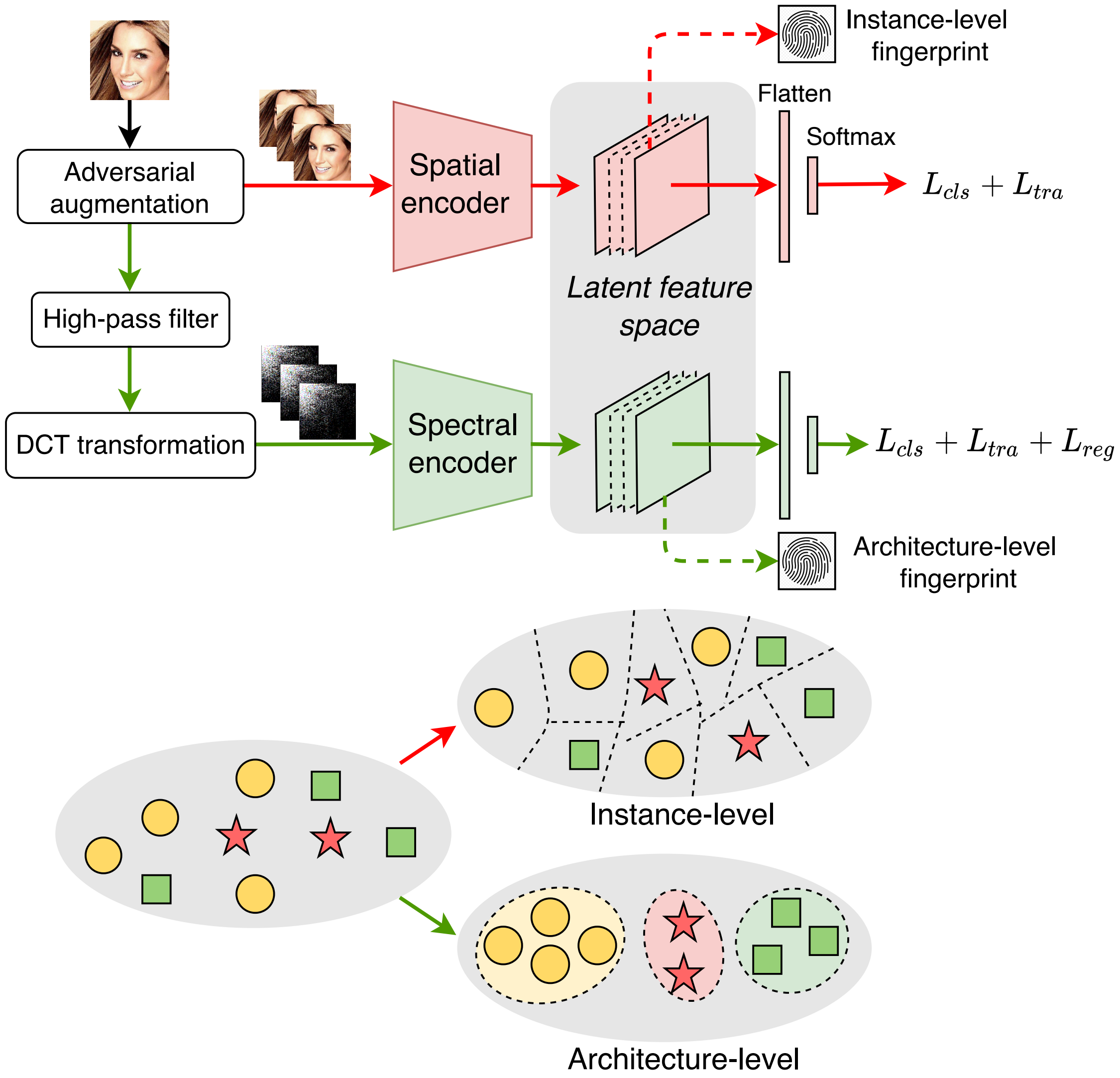


Figure 4.4: The workflow of GAN fingerprint decouple learning, with a schematic showing feature decoupling in the latent feature space. The red and green arrows are the learning flows of instance-level fingerprints and architecture-level fingerprints, respectively. Shapes of the same colour indicate GAN instances with the same model architecture.

The latent feature space is represented as the features output by the last convolutional layer of the CNN. In the latent feature space, it is preferable for the encoder to separate each class with equivalent probability. This can be done by minimising a cross-entropy classification loss in optimization:

$$L_{cls}(I) = -\sum_{\mathcal{G}_i \in \mathbb{G}} \mathcal{G}_i \log(p_i(I)), \tag{4.7}$$

where $\mathcal{G}_i \in \mathbb{G}$ denotes the source class and $p_i(I)$ is the probability output by the softmax function in the final layer of the CNN encoder given an image I.

4.3.1.2 Architecture-level fingerprint learning

Recall that architecture-level fingerprints are typically located in the high-frequency bands of the image. Decoupling these fingerprints can be done by a two-step frequency transformation of the image before feeding the image to the encoder. First, low-frequency signals are removed from the original image with a high-pass filter $H(\cdot)$ to result in "more pure" architecture-level fingerprints in the frequency domain. This is because the low frequency signals represent the major visual information which is typically associated with the image content rather than the GAN architecture information. We apply a Gaussian high-pass filter $H(u,v) = 1 - exp(\frac{-b^2(u,v)}{2b_{thre}^2})$ on the centre-shifted DFT spectrum, where b denotes the spectrum radius from the point (u,v) to the spectrum centre, and b_{thre} is the band threshold for filtering. The filtered DFT spectrum is then transformed back into the spatial image. Second, the filtered images are transformed from raw RGB pixels into two-dimensional discrete cosine transformation (DCT) coefficients. The reason we perform DCT instead of DFT is that, the DFT coefficients contain imaginary components incompatible with the CNN encoder, while DCT coefficients are all real numbers. The use the spectral coefficients as input ensures the frequency information to directly influence the representation of the architecture-level fingerprints. In this way, the learning of the CNN encoder is supervised by the pairs $\{\hat{I}, \mathcal{G})\}$, where $\hat{I} = \mathcal{D}(\mathcal{H}(I)$, $\mathcal{D}(\cdot)$ indicates the DCT transformation and $\mathcal{H}(\cdot)$ indicates the high pass filter. The green arrows in Figure 4.4 demonstrate the workflow of the learning process.

Alongside the image domain transformation at the encoder's input side, disentangling the features in the latent space is additionally performed to further strengthen the decoupling representation of multi-level fingerprints. Since the major difference between the architecture-level fingerprints and instance-level fingerprints is the GAN architecture-dependency, the idea is to enhance the ability of the architecture-level fingerprint encoder in recognizing the patterns associated with the architecture information. As shown in the schematic overview of the latent feature space in Figure 4.4, compared with the instance-level fingerprint learning which seeks to equally separate every GAN instance, the architecture-level fingerprint learning is desired to accurately cluster the GAN instances having the same architecture. This can be done by imposing an architecture-invariant regularisation L_{reg} on the original classification loss, which can enforce to minimize the inner-class distance while maximize the inter-class distance at the architectural level.

Suppose there are C types of GAN architecture in $\mathbb{G}$. For each type of architecture, there are K GAN instances with the same architecture, e.g., $\{\mathscr{G}_0^c, \mathscr{G}_1^c, ..., \mathscr{G}_K^c\}$. Then, given an input pair $(\hat{I}, \mathscr{G}_k^c)$, L_{reg} can be denoted as:

$$L_{reg}(\hat{I}) = \|\boldsymbol{A}(\hat{I}) - (\overline{\boldsymbol{A}^c})\|_2 \tag{4.8}$$

where $\|\cdot\|_2$ is the Euclidean distance, and $\boldsymbol{A}$ is the encoded feature in the latent feature space (i.e., the output of the last convolutional layer of the CNN). $\overline{\boldsymbol{A}^c}$ is the mean feature vector of K GAN instances having the same architecture c, computed as an empirical estimation over the whole training set:

$$\overline{\boldsymbol{A}^c} = \mathop{\mathbb{E}}_{\{\hat{I}^c\}} \boldsymbol{A}(\hat{I}^c) = \frac{1}{K}\sum_{k=0}^{K} \boldsymbol{A}(\hat{I}^{\mathscr{G}_k^c}) \tag{4.9}$$

where $\hat{I}^{\mathscr{G}_k^c}$ indicates the input $\hat{I}$ belongs to the source $\mathscr{G}_k^c$.

4.3.1.3 Robust representation

In practice, GAN-generated images are often distributed online and may undergo several unknown perturbations, such as compression and noise-adding for communication convenience. Thus reliable fingerprints for forensics are desired to be robust to these perturbations. We propose an adversarial augmentation strategy along with a transformation-invariant loss regularisation to meet the need of robustness.

The adversarial augmentation strategy performed on the training dataset includes six empirical online image transformation models:

- **Flipping** the image horizontally or vertically with a probability of 50%.

- **Blurring** the image using a Gaussian filter with kernel size randomly selected from $\{1, 3, 5, 7, 9\}$.

- **Compressing** the image using JPEG compression with the quality factor randomly sampled from [10,75].
- **Cropping** the image along a random axis with percentage sampled from [5%,20%], then resizing the cropped image back to the original resolution.
- **Rotating** the image with an angle randomly sampled from {45°,90°,135°,180°,225°, 270°,315°}.
- **Noise** adds i.i.d. Gaussian noise with a Gaussian variance randomly sampled from [5.0,20.0].

These transformation models are combined in the above order and executed with a probability $p = 0.5$ for each during the augmentation. For each training sample I, we repeat the combined transformation $\mathscr{T}$ for T times, i.e., $\mathscr{T} = \{\mathscr{T}_1, \mathscr{T}_2, ..., \mathscr{T}_T\}$. Additionally, since whatever the transformation is, it cannot change the true source information of the image, we add a transformation-invariant loss regularisation. This loss is to ensure the images from the same source to get closer to each other at the fingerprint level in the presence of transformations, denoted as:

$$L_{tra}(x) = \|\boldsymbol{A}(x) - \frac{1}{T}\sum_{t=0}^{T}\boldsymbol{A}(x_t)\|_2, \tag{4.10}$$

where for instance-level fingerprint encoder, $x = I$ and $x_t = \mathscr{T}(I)$; for architecture-level fingerprint encoder, $x = \hat{I}$ and $x_t = \mathscr{D}(\mathscr{H}(\mathscr{T}(I)))$.

The final optimisation objectives for the spatial and spectral fingerprint encoders are

$$\begin{gathered} \min_I \mathbb{E}(L_{cls} + \lambda_1 L_{tra}), \\ \min_I \mathbb{E}(L_{cls} + \lambda_1 L_{tra} + \lambda_2 L_{reg}) \end{gathered} \tag{4.11}$$

respectively, where λ_1 and λ_2 are the balancing weights. After training, the encoded feature vector $\boldsymbol{A}$ can be used as the implicit fingerprint.

4.3.2 Explicit fingerprint representation

The implicit fingerprints learned in the latent feature space are difficult to interpret. According to Yu et al. [158], explicit exposure of the fingerprint in either the spatial or frequency domains with a scale-consistent localisation mechanism is more rational and feasible. That is to say, with both pixel input $I \in \mathbb{R}^{N\times N}$ and DCT spectrum input $\hat{I} \in \mathbb{R}^{N\times N}$, the fingerprint should have a visible 2D view with the scale $N \times N$. Inspired by the Gradient-weighted Class Activation Mapping (Grad-CAM) mechanism for deep neural network interpretation [127], we proposed a Grad-CAM-based fingerprint representation method. Compared with the existing GAN fingerprint models [100, 153, 158] which require noisy post-processing or reconstruction operations, our implementation utilizes the inherent attention map of a CNN, where no extra noise is introduced and the computational overhead is lower.

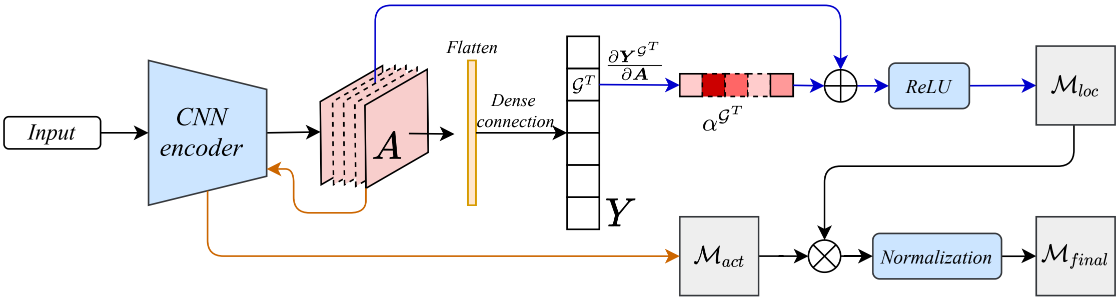


Figure 4.5: The end-to-end mechanism that creates the localisation mask $\mathcal{M}_{final}$ for the explicit fingerprint representations from the latent feature space. The orange arrow indicates the data flow associated with the class-specific localisation map $\mathcal{M}_{loc}$. The blue arrow indicates the data flow for the back-propagation activation map $\mathcal{M}_{act}$. ⊗ denotes element-wise multiplication, and ⊕ denotes a weighted linear combination.

As shown in Figure 4.5, a class-specific localisation map $\mathcal{M}_{loc}$ is computed first, which can indicate the feature importance regarding the given target class in the latent feature space. Formally, given a target class $\mathcal{G}^T$ of a GAN instance, let $\boldsymbol{Y}^{\mathcal{G}^T}$ be the predicted class score vector of $\mathcal{G}^T$ before the softmax function in the forward propagation of the CNN encoder, and $\boldsymbol{A}$ has a length of S. The localization map $\mathcal{M}^{\mathcal{G}^T}_{loc}$ can be computed as a rectified partial linear combination of $\boldsymbol{A}$, weighted by the back-propagation gradients:

$$\mathcal{M}^{\mathcal{G}^T}_{loc} = \mathrm{ReLU}\left(\sum_s \alpha_s^{\mathcal{G}^T} \boldsymbol{A}^s\right), s \in [1,S] \tag{4.12}$$

$$\alpha_s^{\mathcal{G}^T} = \frac{1}{Z}\sum_i\sum_j \frac{\partial \boldsymbol{Y}^{\mathcal{G}^T}}{\partial \boldsymbol{A}^k_{ij}} \tag{4.13}$$

where the weight $\alpha_k^{\mathcal{G}^T}$ is a global average of the gradients of $\boldsymbol{Y}^{\mathcal{G}^T}$ with respect to each point (i,j) in the k-th feature map of $\boldsymbol{A}^k$, and Z is the normalisation constant. ReLU(·) is the rectification function that forces the map to focus on the features with a positive influence over the class of interest.

The class-specific localisation map $\mathscr{M}_{loc}$ can only highlight a rough region of interest for a particular class. Hence, the map is fused with a back-propagation activation map $\mathscr{M}_{act}$ to obtain fine-grained importance of the fingerprint response at each entry of the input. The activation map $\mathscr{M}_{act}$ is computed via Guided Back-propagation [129], which inverts the data flow of the current CNN, i.e., backward passing the gradients through the network to reconstruct a point-wise max activation map from the input data. Given $\mathscr{G}^T$ and the l-th layer of the CNN encoder, the guided back-propagation from the l-th layer to the prior $l-1$-th layer is formulated as:

$$\boldsymbol{R}_{l-1}^{\mathscr{G}^T} = \left(\boldsymbol{f}_{l-1}^{\mathscr{G}^T} > 0\right) \cdot \left(\boldsymbol{R}_{l}^{\mathscr{G}^T} > 0\right) \cdot \boldsymbol{R}_{l}^{\mathscr{G}^T} \tag{4.14}$$

$$\boldsymbol{R}_{l}^{\mathscr{G}^T} = \frac{\partial \boldsymbol{A}}{\partial \boldsymbol{f}_{l}^{\mathscr{G}^T}}, \quad \boldsymbol{f}_{l}^{\mathscr{G}^T} = \text{ReLU}\left(\boldsymbol{f}_{l-1}^{\mathscr{G}^T}\right) \tag{4.15}$$

where $\boldsymbol{f}_l$ is the activated feature vector output by the l-th layer in the forward propagation. When the computation is propagated to the first layer, i.e., $l = 1$, the max activation map $\mathscr{M}_{act}^{\mathscr{G}^T} = \boldsymbol{R}_0^{\mathscr{G}^T}$ results. $\mathscr{M}_{loc}$ and $\mathscr{M}_{act}$ are then fused through point-wise multiplication and normalised to obtain the final localization mask:

$$\mathscr{M}_{final} = \mathscr{N}(\mathscr{M}_{loc} \otimes \mathscr{M}_{act}) \tag{4.16}$$

where $\mathscr{N}(\cdot)$ is a zero-mean normalisation function to scale each entry of the mask into the range $[0,1]$, and $\otimes$ denotes the point-wise multiplication. Note that the raw scale of $\mathscr{M}_{loc}$ is the same as that of the feature map in $\boldsymbol{A}$ and should be rescaled to $N \times N$ before fusion. The rescaling will not inject extra noise as $\mathscr{M}_{loc}$ only represents the localization information.

Finally, the final localisation mask reveals a visible 2D fingerprint. For a target GAN class $\mathscr{G}^T$ and an image $\boldsymbol{I}$, the explicit two levels of fingerprints are:

$$\begin{aligned} \boldsymbol{F}_{ins} &= \mathscr{M}_{final}(\boldsymbol{I}) \otimes \boldsymbol{I}, \quad \boldsymbol{F}_{ins} \in \mathbb{R}^{N\times N} \\ \boldsymbol{F}_{arc} &= \mathscr{M}_{final}(\hat{\boldsymbol{I}}) \otimes \hat{\boldsymbol{I}}, \quad \boldsymbol{F}_{arc} \in \mathbb{R}^{N\times N} \end{aligned} \tag{4.17}$$

4.4 Task-specific forensics

Once the architecture-level and instance-level GAN fingerprints have been extracted, a challenge is that how to perform task-specific fingerprinting with the two levels of fingerprints in practice. We elaborate three typical GAN-generated image forensics scenarios and the corresponding fingerprinting methods. Table 4.2 shows the threat models and related assumptions in practice of the three scenarios.

Table 4.2: Threat models of three typical GAN-generated image forensics scenarios.

Task	**Subject role**	**Subject's target**	**Subject's knowledge**
Black-box fake image detection	DeepFake forensic investigator	Detect malicious fake images	zero knowledge of the source GAN; only have real images
Model IP protection	GAN model owner	Prevent GANs from pirate by unique fingerprint	full access to the source GAN;
Fingerprinting attack and defense	Fingerprinting attacker	Model inversion attack using fingerprint approximation	zero knowledge of the source GAN
	Defender/model owner	Anonymize GAN fingerprint	full access to the source GAN and the training process;

4.4.1 Black-box fake image detection

The architecture-level fingerprint is more applicable than the instance-level one in fake image detection tasks. The reason is that, in reality, a DeepFake attacker can incrementally refine the back-end GAN with new data, without changing the model architecture. Compared with the instance-level fingerprints, the architecture-level fingerprints can generalize well for GANs with the same architecture, which would not lose the forensics efficacy after the DeepFake GAN has got updated.

The major challenge in this task is that the detection is often performed in black-box where the source DeepFake GAN is inaccessible. In this way, training a fingerprint encoder to obtain the DeepFake GAN's fingerprint is no longer available. We propose to formulate this problem as a one-class anomaly detection problem where only real images are regarded as "normal", and use the architecture-level fingerprints with the assistance of auxiliary background knowledge to detect anomalies. First, we sample sufficient real and GAN-generated images from publicly available databases and train a CNN encoder with a great ability in identifying real images. Once trained, the encoder is embedded into the proposed extraction workflow of architecture-level fingerprints to act as an extractor of the "fingerprints" associated with the real image class. In the detection phase, given an unseen sample, we first search its top-q most visually similar images from a real image dataset. Then the authenticity of the test sample can be identified by measuring the fingerprint correlation:

$$\mathscr{C} = \mathscr{N}(\overline{\boldsymbol{F}}_{arc}^{real}) \odot \mathscr{N}(\boldsymbol{F}_{arc}^{test}) \tag{4.18}$$

where $\overline{\boldsymbol{F}}_{arc}^{real}$ denotes the mean architecture-level fingerprint averaging over the q real-world images, which acts as a ground truth of the "real" class, $\boldsymbol{F}_{arc}^{test}$ denotes the architecture-level fingerprint of the test sample, and $\odot$ indicates inner product.

4.4.2 Model IP protection

GAN fingerprints can be used as an intrinsic and intentional signature embedded in its products to prevent theft and plagiarism. Since the model owner obviously has full access to and knowledge of their own technology, this application is a white-box attribution scenario. As such, instance-level fingerprinting may be more appropriate than architecture-level protection because the instance-level fingerprints can precisely differentiate between any two GAN instances trained in settings with even subtle differences, irrespective of whether they have the same architecture. The same cannot be said of architecture-level fingerprints (see Section 4.2). The fingerprinting process is as follows: First, the model owner pretrains an instance-level fingerprint encoder $f_{ins}(\theta)$ with parameters θ using the images generated by the owned GAN model. Then the owner queries a batch of images $\mathbf{I}_{test}$ from the suspicious target GAN and perform an classification $f_{ins}(\mathbf{I}_{test}|\theta)$ to verify whether the target GAN is a plagiarism.

Note that the proposed fingerprint representation method is independent of the encoder's network topology, and thus is compatible with arbitrary CNN configurations. In this way, the model owner can hold the parameter of the encoder θ as a unique private key Key(θ) to secure the fingerprinting process, i.e., $f_{ins}(\mathbf{I}_{test}|\theta, \text{Key}(\theta))$. Others without the correct encoder cannot perform the fingerprinting, which ensures high-level protection.

4.4.3 Fingerprinting attack and defence

We also identify that the architecture-level GAN fingerprinting can be exploited as a black-box attack to steal the model information of a GAN service. Given enough queries to a GAN API, an attacker could perform architecture-level fingerprinting to obtain the fingerprint distribution, and thus can infer the exact architecture by distribution matching with pre-trained GAN models.

The architecture information should be protected because once the attacker knows the architecture, they can 1) perform direct reverse engineering of the back-end model at a low cost [3]; or 2) be possible to build a shadow model to launch a member inference attack [22]. To defend against the fingerprinting attack, we propose to anonymize architecture information by manipulating the architecture-level GAN fingerprint during the training process of GAN via adding an additional fingerprint anonymization loss to the general generator loss:

$$\mathscr{L}(G(z)) = \underbrace{\log(1 - D(G(z))}_{\text{original generator loss}} + \underbrace{\beta \cdot \|\hat{\boldsymbol{F}}_{arc}^{real} - \boldsymbol{F}_{arc}^{G(z)}\|_1}_{\text{fingerprint anonymisation loss}} \tag{4.19}$$

where $\boldsymbol{F}_{arc}^{G(z)}$ is the architecture-level fingerprint extracted from the output image $G(z)$ of the generator. $\hat{\boldsymbol{F}}_{arc}^{real}$ is the architecture-level fingerprint averaged over a number of real images randomly sampled from a real-world dataset at each training iteration. $\|\cdot\|$ is the element-wise ℓ_1-norm and β is the parameter to balance weight.

The regularisation loss enforces a minimum distance between the fingerprints in generated images versus real ones. Alternatively, the loss could be extended to anonymise the current fingerprint by mimicking another GAN architecture. This could be done by simply changing the samples for averaging the target fingerprint from real images to ones generated by another GAN.

4.5 Experimental evaluation

The experiments we conducted are designed with three goals: 1) to verify our theoretical analysis of GAN fingerprint dependency; 2) to evaluate the performance and robustness of the proposed GAN fingerprint extraction method; and 3) to estimate the practical usefulness of fingerprint decoupling in real-world image forensic tasks.

4.5.1 General settings

4.5.1.1 Datasets

Two datasets of real images are used:

- The *CelebA* dataset [93], which consists of 202,599 images of celebrity face sized $178 \times 218 \times 3$.

- The *LSUN* dataset [157], which consists of more than 3 million $256 \times 256 \times 3$ pictures of bedroom scenes.

All images are center-cropped or resized down to a size of $128 \times 128 \times 3$ to facilitate the training of different GAN models. The experiments involve six different GAN architectures, among which four are popular GANs for image generation: ProGAN [66], SNGAN [106], CramerGAN [9], MMDGAN [10], and two are widely-used GANs for image-to-image translation: CycleGAN [164] and StarGAN [26].

4.5.1.2 Encoder setup

The input data for the instance-level fingerprint encoder is RGB pixel values, rescaled down to $[0,1]$ with a rescaling factor of 1/255. The band threshold b_{thre} of the high-pass filter $\mathscr{H}(\cdot)$ for the architecture-level fingerprint encoder is set to 40 for the *CelebA* images and 70 for the *LSUN* images. The filtered images are then transformed into DCT spectra and used as the input data. The coefficients in the spectrum are log-scaled and normalised via zero-mean normalisation. The weight λ_1 and λ_2 in Eq.4.11 are set to 1 and 0.1 respectively. An Adam optimiser [72] with a learning rate of 0.001 and a training batch size of 128 are used to optimise both encoders. The maximum number of iteration epochs is 50.

4.5.1.3 Evaluation metrics

- F_1 score. Since the source attribution task is formulated as a classification problem, we report the F_1 score for each individual class, computed as $F_1 = \frac{2 \times P \times R}{P+R}$, where P and R are the precision and recall values. We also use Macro F1 scores (mF_1) to measure the overall performance, which is the average F_1 over all classes.

- AP score. When the attribution task is narrowed down to a binary classification problem, We use AP instead of F_1 scores as the measurement, which is computed as $AP = \sum_t (R_t - R_{t-1}) P_t$, where P_t and R_t are the precision and recall at the t-th threshold.

4.5.2 Verification of GAN fingerprint dependency

To verify the theories that outline GAN fingerprint dependency in Section 4.2, we conduct several evaluations with a particular focus on the correspondences between upsampling bias and architecture-level fingerprinting ($\boldsymbol{F}_{arc}$), as well as between random bias and instance-level fingerprinting ($\boldsymbol{F}_{ins}$).

4.5.2.1 Upsampling bias

This experiment involves performing a series of source attribution tests on images generated by GANs with different upsampling units. The GANs tested are:

- *StarGAN-V0*: A standard StarGAN consisting of two transposed convolution-based upsampling layers with a kernel size of 4 in the generator [26]. We used the official pre-trained StarGAN release [1].
- *CycleGAN*: A standard CycleGAN whose generator architecture is similar to StarGAN. We used the pre-trained CycleGAN in [159].
- *StarGAN-V1*: A modified version of the standard StarGAN with the transposed convolution units in the first upsampling layer replaced by interpolated up-convolution units. This model was trained from scratch.
- *StarGAN-V2*: A modified version of the standard StarGAN with interpolated up-convolution units to replace the transposed convolution units in both upsampling layers. This model was also trained from scratch.
- *StarGAN-V3*: Another modified version of standard StarGAN with the kernel size in the first upsampling layer reduced to 2 and duplicated to act as the first two layers in the sequence ‚Äì again, trained from scratch.

All GANs are pre-trained or trained from scratch with the *CelebA* dataset and act as candidate sources. The fingerprint encoders are trained with $10,000$ images generated by the standard StarGAN (*StarGAN-V0*) and $10,000$ real images from *CelebA*, with the data samples split into training and validation sets at a ratio of 4/1. The sample size ($10,000$ per class) and ratio of training/validation sets (4/1) are also applied in the subsequent experiments unless otherwise specified. We independently query $1,500$ images per source to build a testing dataset.

The results for this series of experiments, reported as *AP* scores, are shown in Table 4.3. Both $\boldsymbol{F}_{arc}$ and $\boldsymbol{F}_{ins}$ perform source attribution well on the *StarGAN-V0* samples. Additionally, $\boldsymbol{F}_{arc}$ generalizes better than $\boldsymbol{F}_{ins}$ with the samples from *CycleGAN*, which has a similar architecture to *StarGAN-V0*. In contrast, the *AP* scores for $\boldsymbol{F}_{arc}$ drop

[1] https://github.com/yunjey/stargan

significantly with the modified versions of *StarGAN* than for $\boldsymbol{F}_{ins}$. This result confirms that architecture-level fingerprinting is more sensitive to changes in the upsampling unit than instance-level fingerprinting.

Table 4.3: The $AP(\%)$ scores of $\boldsymbol{F}_{arc}$ and $\boldsymbol{F}_{ins}$ from a source attribution test for GANs with different upsampling units.

	StarGAN-V0	*CycleGAN*	*StarGAN-V1*	*StarGAN-V2*	*StarGAN-V3*
$\boldsymbol{F}_{arc}$	100.00	87.55	38.11	21.76	31.61
$\boldsymbol{F}_{ins}$	99.83	58.78	67.08	55.71	60.40

4.5.2.2 Random bias

As mentioned in Section 4.2, there are two main factors that influence random bias:

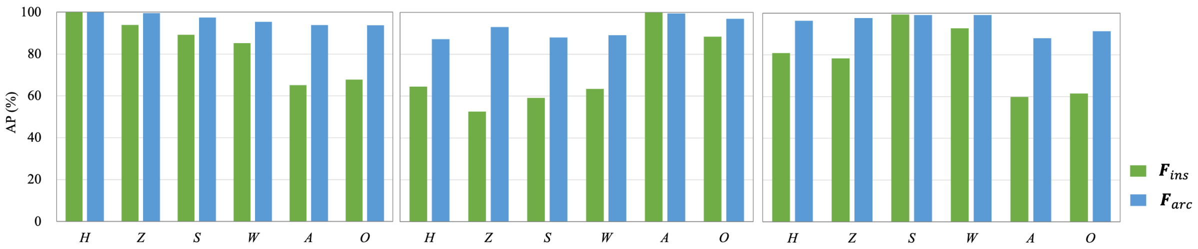
- the semantic information in the training set;
- the differences in the training settings.

Hence, in this next set of experiments, we test each of the two factors in turn.

Bias from semantic information Here, we take six officially released CycleGAN models [2] which were previously pre-trained for three paired image-to-image translation tasks, each belonging to a different semantic domain: horse (*H*) ⇔ zebra (*Z*), summer (*S*) ⇔ winter (*W*) and apple (*A*) ⇔ orange (*O*). $\boldsymbol{F}_{ins}$ and $\boldsymbol{F}_{arc}$ encoders are trained for binary attribution ('real' versus 'CycleGAN') tasks for the H, A and S categories, and tested on all categories. Figure 4.6 shows the results.

Unsurprisingly, in cases that the testing samples belong to the same or very similar semantic domain with the training samples, e.g., horse versus zebra, both fingerprints return high AP scores. However, in other domains, the AP scores for $\boldsymbol{F}_{ins}$ are significantly lower in comparison to those for $\boldsymbol{F}_{arc}$. In addition, $\boldsymbol{F}_{ins}$ generalizes relatively better across H, Z, S and W domains than the A and O ones. One possible reason is that images in the first four domains all have landscapes as backgrounds, which is quite different from the ones in the latter two domains. These results suggest that instance-level fingerprinting is much more sensitive to semantic information than architecture-level fingerprinting. And the latter one has a pronounced ability in generalising to source GANs that have the same architecture (e.g., CycleGAN), despite that they are trained in different domains.

[2] https://junyanz.github.io/CycleGAN/



(a) Encoders trained on the Horse (H) images (b) Encoders trained on the Apple (A) images (c) Encoders trained on the Summer (S) images

Figure 4.6: Bias in semantic information for two different fingerprints, evaluated in terms of AP. Three encoders were trained for each fingerprint with the real and GAN-generated images in the horse (H), apple (A) and summer (S) domains, then tested in the H, A, S, zebra (Z), winter (W) and orange (O) domains.

Bias from the training settings Here, we train several ProGANs from scratch varying only one of three variables at a time as follows:

- *Var #1 Sample size*: varies between 40%, 60%, 80% and 99% of 150,000 randomly sampled *CelebA* images.
- *Var #2 Convergence point*: five instances trained with the base dataset that each converges at a different (sub-optimum) point.
- *Var #3 Initialisation seeds*: five instances, each with a different number of random seeds specified for the weight initialisation.

The mF_1 scores of these tests appear in Table 4.4. Remarkably, $\boldsymbol{F}_{ins}$ outperforms $\boldsymbol{F}_{arc}$ in all tasks, indicating that $\boldsymbol{F}_{ins}$ is highly sensitive to different training settings. Even a minor difference, such as a 1% difference in the number of training samples, leads to distinguishable instance-level fingerprinting. By contrast, $\boldsymbol{F}_{arc}$ is much less sensitive to the training settings, especially the convergence points and initialisation weights. One reason may be that changes in the frequency domain resulting from weight settings are filtered by the convolution kernels during training.

Table 4.4: $mF_1(\%)$ scores for two fingerprinting models in three tasks with different settings.

	Var #1	*Var #2*	*Var #3*
$\boldsymbol{F}_{ins}$	99.24	96.11	97.53
$\boldsymbol{F}_{arc}$	84.19	69.92	64.09

4.5.3 GAN attribution performance and robustness

We compare the source attribution effectiveness of the proposed fingerprints with five state-of-the-art source attribution methods, including the deep CNN-based method (**CNN**) [146], the spectral distribution-based method (**SD**) [32], the DCT spectrum-based method (**DCTA**) [39], and other two GAN fingerprint models: the residual-based fingerprints (**RF**) [100] and the learning-based fingerprints (**LF**) [158]. All baselines are configured with the hyper-parameters recommended in the original papers.

The assigned benchmark task is an five-source image attribution task described in both [158] and [39]. The goal is to attribute images from the *CelebA* and *LSUN* datasets to one of five candidate sources: real, ProGAN, SNGAN, CramerGAN, and MMDGAN. Hence, we randomly sample $10,000$ images as the real class, then use the pre-trained GAN instances in [158] to generate $10,000$ image samples for each GAN source class. The generated samples are then divided into $7,500$ training, $1,500$ validation, and $1,500$ test images, resulting in a combined set of $37,500$ training, $7,500$ validation and $7,500$ test images, for each dataset. Source attribution is performed using CNN encoders to create fingerprint representations at the instance and model levels.

4.5.3.1 GAN source attribution performance

Table 4.5 shows F_1 score results. Generally, all methods perform better with the *CelebA* dataset than the *LSUN* dataset. This is not surprising as the *LSUN* images (of bedroom scenes) are more complex and diverse than the face images in *CelebA*.

With high overall F_1 scores, the proposed architecture-level fingerprinting and instance-level fingerprinting are both highly effective (98.00% versus 99.64% in the *CelebA* dataset and 96.24% versus 97.73% in the *LSUN* dataset) at identifying the correct source. The architecture-level fingerprinting $\boldsymbol{F}_{arc}$ which is performed directly in the DCT spectrum shows the comparable or superior performance than the state-of-the-art baselines, on almost all sources in both datasets. This result illustrates the feasibility of extracting distinguishable GAN fingerprints from the frequency domain. The instance-level fingerprinting method $\boldsymbol{F}_{ins}$ deliveres comparable performance to **LF**. This is not unexpected as both techniques learn fingerprints through a similar pipeline of RGB pixels fed into a CNN encoder. Of all the seven methods, **RF**'s performance is the inferior. The reason is that **RF** is estimated by a statistical averaging process that is prone to stochastic errors and, thus, is less precise. The other three techniques all incorporate learning processes that support the automated discovery of fine-grained

Table 4.5: F1 scores ($mF_1(\%)$) evaluated in the five-source benchmark attribution task in the *CelebA* and *LSUN* datasets. Best performances per source are highlighted in bold.

		Real	*ProGAN*	*SNGAN*	*CramerGAN*	*MMDGAN*	**Overall** (mF_1)
CelebA	**CNND** [146]	96.09	97.45	95.59	92.39	92.94	94.89
	SD [32]	96.90	98.18	97.61	96.37	96.03	97.02
	RF [100]	53.03	79.63	93.94	71.80	67.30	73.14
	DCTA [39]	**99.85**	**99.49**	99.78	98.18	99.09	99.60
	LF [158]	98.70	98.85	99.18	97.86	97.53	98.42
	$\boldsymbol{F}_{ins}$ (ours)	98.19	98.58	98.81	97.35	97.07	98.00
	$\boldsymbol{F}_{mod}$ (ours)	99.60	99.33	**99.79**	**98.64**	**99.67**	**99.64**
LSUN	**CNND** [146]	97.62	98.53	90.96	76.80	74.22	87.62
	SD [32]	87.72	86.63	95.64	92.07	91.52	90.71
	DCTA [39]	**99.79**	98.51	98.70	93.82	93.35	96.83
	RF [100]	62.38	66.08	63.72	77.93	76.59	69.34
	LF [158]	98.70	98.60	**99.82**	93.00	92.20	96.46
	$\boldsymbol{F}_{ins}$ (ours)	98.58	98.61	99.79	91.84	92.40	96.24
	$\boldsymbol{F}_{mod}$ (ours)	99.09	**98.91**	94.74	**99.85**	**94.56**	**97.43**

fingerprint features.

4.5.3.2 Explicit representation performance

Figure 4.7 shows some of the explicit representations produced by $\boldsymbol{F}_{ins}$ and $\boldsymbol{F}_{arc}$ from the *CelebA* and *LSUN* datasets. The results are presented as an original-fingerprint pair for each GAN source. All examples are averaged over 256 samples for better interpretability. The brighter areas in the fingerprints correspond to the components of the original input that make a more significant contribution to the final source attribution. $\boldsymbol{F}_{ins}$ and $\boldsymbol{F}_{arc}$ each reveales significantly different patterns in terms of location and intensity, providing greater insight into why some models are better at correctly identifying GAN-generated images than others.

In discussing the efficacy of $\boldsymbol{F}_{ins}$, we largely focus on the examples from *CelebA* since the faces and fingerprints are more discernible in these images than in the *LSUN* ones. As can be seen, the fingerprints of ProGAN, SNGAN, and CramerGAN are centralised around the eyes, nose, and mouth, while the fingerprints of MMDGAN are more global. These relationships add evidence to the analysis that $\boldsymbol{F}_{ins}$ is closely associated with the semantic information in visual content.

In terms of $\boldsymbol{F}_{arc}$, the fingerprints directly point out which frequency components contribute remarkably in source attribution. From the figure, we can see that $\boldsymbol{F}_{arc}$ was able to capture quasi-periodic artifacts in the high-frequency space, especially for

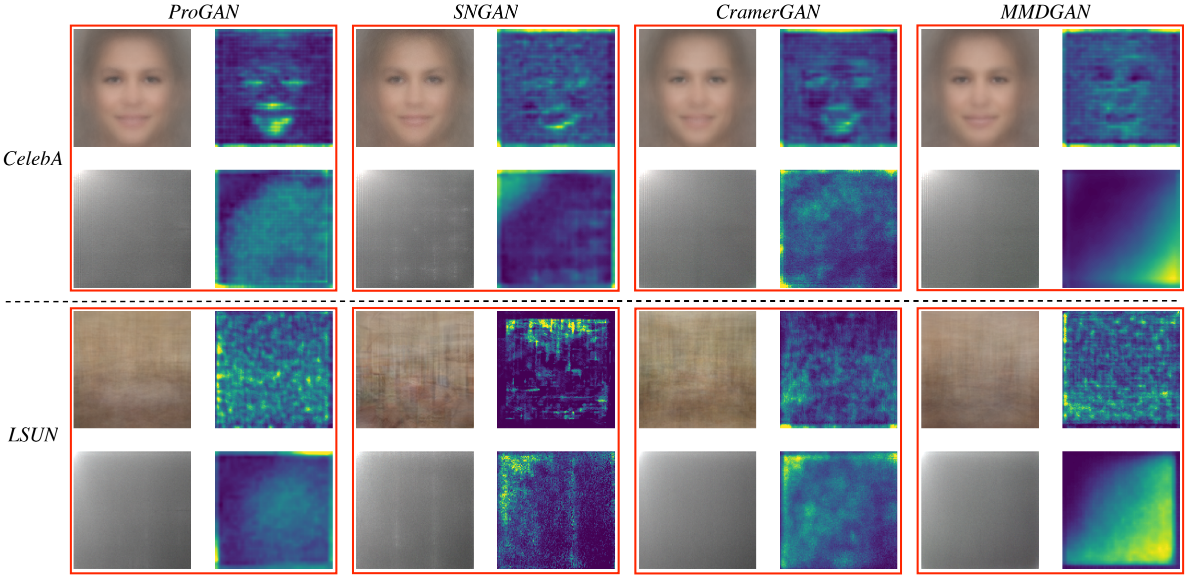


Figure 4.7: Explicit representations of the instance-level fingerprints in the spatial domain and the architecture-level fingerprints in the frequency domain for four GAN sources from two datasets. Each red box shows the input-fingerprint pair. Both were averaged over 256 samples for better interpretability.

SNGAN. Furthermore, the fingerprints left by models with the same architecture but trained on different datasets share a certain similarity. For example, the architecture-level fingerprints left by MMDGAN in either dataset are both notably centralised in the high-frequency domain, and the ones left by CramerGAN are dispersed over the global spectrum.

4.5.3.3 Robustness

We evaluate the robustness of the proposed fingerprints against image perturbations with/without the strategy described in Section 4.3.1.3. Besides the six aforementioned image perturbations, we also consider a mixture perturbation which randomly combines two or more of the six perturbations. This is because images are not uncommon to undergo multiple perturbations in practice.

This evaluation is also made through the above benchmark attribution task on both datasets and in comparison to **RF** and **LF**. Table 4.6 provides the final results as mF_1 scores. The labels $w/.$ and $w/o.$ indicate whether the fingerprint encoders are trained with the robustness-enhancing strategy or not. It should come as no surprise that the mF_1 scores for all were significantly lower in this exercise than the last in Table 4.5. $\boldsymbol{F}_{arc}$ demonstrates generally better robustness than the other methods against most treatments except for blurring and noise. This exception is because blurring and noise usually lead to considerable changes directly in the high-frequency components of an image. After adopting the data augmentation strategy and transformation-invariant loss regularisation, both $\boldsymbol{F}_{ins}$ and $\boldsymbol{F}_{arc}$ show a significant improvement in their ability to withstand perturbation.

Table 4.6: Macro F1 scores ($mF1(\%)$) reflecting the robustness of four GAN fingerprinting methods to common image perturbations. The highest value is highlighted in bold. $w/$ and w/o indicate whether the fingerprint encoders are trained with the robustness-enhancing strategy or not.

		Flipping		**Blurring**		**Compression**		**Cropping**		**Rotation**		**Noise**		**Mixture**	
		w/o	$w/$	w/o	$w/$	w/o	$w/$	w/o	$w/$	w/o	$w/$	w/o	$w/$	w/o	$w/$
CelebA	**RF[100]**	70.34	–	38.70	–	41.64	–	31.97	–	41.19	–	56.12	–	43.70	–
	LF[158]	81.68	93.18	**69.39**	88.40	52.02	74.73	74.03	95.76	57.17	95.96	**76.45**	**95.22**	50.66	67.82
	$\boldsymbol{F}_{ins}$	80.26	94.27	61.42	**88.41**	51.16	70.23	72.46	90.47	57.30	92.42	75.44	92.17	49.79	86.95
	$\boldsymbol{F}_{arc}$	**87.32**	**95.74**	52.15	87.91	**60.49**	**89.59**	**77.13**	**96.75**	**76.32**	**96.90**	55.83	85.37	**57.54**	**89.52**
LSUN	**RF[100]**	63.42	–	31.78	–	33.86	–	35.14	–	34.87	–	55.63	–	42.49	–
	LF[158]	75.62	92.67	**69.29**	79.34	53.96	73.76	66.26	**91.20**	52.27	93.55	**71.93**	88.49	45.72	74.18
	$\boldsymbol{F}_{ins}$	70.15	90.53	67.27	75.88	52.25	67.04	63.03	88.12	48.61	91.74	68.48	84.67	**53.14**	85.85
	$\boldsymbol{F}_{arc}$	**76.11**	**93.33**	64.01	**80.20**	**61.73**	**80.71**	**76.16**	90.37	**82.29**	**93.70**	53.40	**91.67**	49.60	**90.59**

4.5.4 Task-specific fingerprinting

4.5.4.1 Black-box fake image detection

To create the simulation of a black-box detection scenario, we use two GAN-based online image generation tools (`https://generated.photos/`, denoted as *Tool #1*, and `https://thispersondoesnotexist.com`, denoted as *Tool #2*). Both can generate 1024 × 1024 × 3 RGB fake face images. Neither site publishes details of its back-end GAN model, save that the images are "imagined by" StyleGAN (Dec 2018, which we traced back to [69].

We assume the images from *Tool #1* are available and we already know they are generated by GAN. The goal of the task we design is to leverage these images plus real-world images to detect a fake image generated by *Tool #2*. To this end, we sample 9,000 real images from the *CelebA-HQ* dataset [66] (a 1024 × 1024 × 3-resolution version of the *CelebA* dataset and generate 9,000 fake images from *Tool #1*. We use these to train a binary $\boldsymbol{F}_{arc}$ encoder. We then sample/generate another 1,000 images each from *CelebA-HQ* and *Tool #2* for testing, applying the detection method outlined in Section 4.4.1 to test each sample. The parameter q is set to 100. The results are shown in Figure 4.8. The two distributions are well separated, allowing for reliable discrimination. The fake test samples centre around zero, meaning that there is little correlation with the real samples even though, visually, they may be very realistic. In this simulation, we are able to reach a mF_1 score of 92.10% with a simple cut-off-based rule.

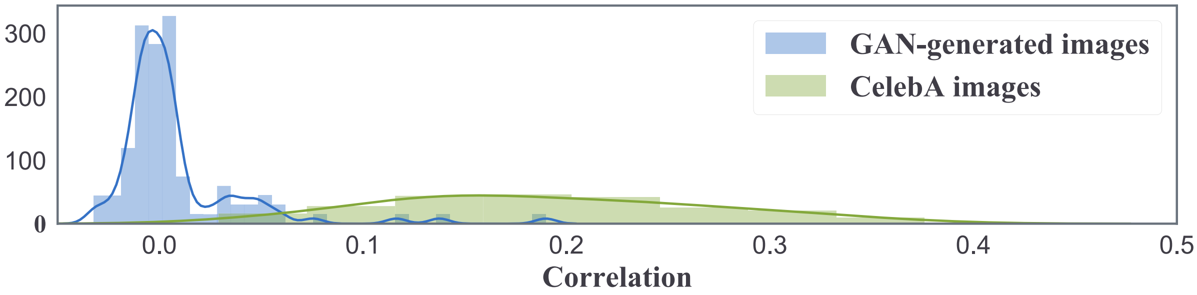


Figure 4.8: The correlation distribution of the test sample fingerprints with the ground truth fingerprints. Blue indicates the fake test samples from *Tool #2*; Green indicates the real samples from *CelebA-HQ*.

4.5.4.2 Model IP protection

We do not conduct a specific simulation for this scenario, since the key features can be substantially verified in the previous experiments: The experimental results in Section 4.5.2.2 show that, two GAN instances resulting from any two slightly different training settings, such as different sample sizes or converge points, are well differentiable using the instance-level fingerprints. And the instance-level fingerprinting efficacy is significantly superior than using the architecture-level fingerprints.

4.5.4.3 Fingerprinting attack and defence

For the simulation to test the scenario in Section 4.4.3, we train the encoders for a binary classification task using *CelebA* images and the images queried from the pre-trained ProGAN in [158]. All the same setting for training a ProGAN as in [158] are followed except that we apply the fingerprint anonymisation loss (Eq. 4.19) to train a new ProGAN model from scratch with anonymised fingerprints. The parameter β is set to 1.0. We test the attribution model with 1,000 *CelebA* images, 1,000 original ProGAN images and 1,000 anonymised ProGAN images.

Figure 4.9 illustrates the confusion matrix of the classification. The fingerprint distinguishes the original ProGAN images accurately from the real images with the mF_1 score of 99.80%. However, after the ProGAN fingerprints have been anonymised, the fingerprinting model fails to make correct decisions with 946 more ProGAN images being misclassified as real *CelebA* images and the mF_1 score decreasing to 5.40%. The result confirms the anonymisation effectiveness.

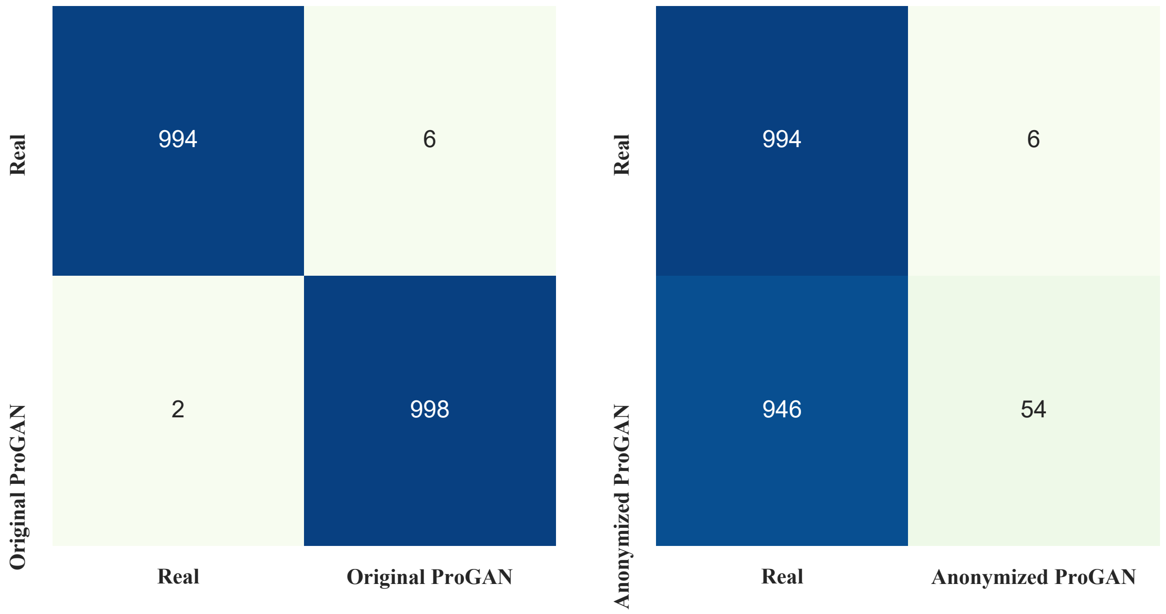


Figure 4.9: The confusion matrices of binary detection between *CelebA* images and original ProGAN images, as well as *CelebA* images and anonymised ProGAN images.

4.6 Summary

In this chapter, we explored the use of GAN fingerprints as an image forensics tool. In practice, different image forensics tasks may require the GAN fingerprinting to perform at different levels of distinguishability. From a dependency analysis of GAN fingerprints, we found out that GANs leave instance-level fingerprints in the spatial domain and model-level fingerprints in the frequency domain. Based on this finding, we designed a fingerprint decouple learning method that offers distinguishability at these granularities. The method also improves fingerprinting over the status quo with an end-to-end explicit representation mechanism. A data augmentation strategy plus transformation-invariant loss regularisation helps improve the robustness of GAN fingerprints to common image perturbations. In a benchmark source attribution test, the proposed method achieves better performance than its two predecessors, e.g., 36.23% and 1.23% improvement on **RF** and **LF**, respectively, in terms of the mF_1 score in the *CelebA* dataset. Furthermore, three different but common image forensics case studies illustrate the usefulness of multi-level GAN fingerprinting.

Chapter 5

Robust Detection via Multi-view Reconstruction-classification Learning

Images manipulated using deep generative models, also known as deepfakes, have posed great threats to the trustworthiness of visual media. Detecting GAN-generated images now becomes a critical task to prevent malicious deepfakes. Although many detectors have shown high detection accuracy on specific GANs, the success is largely attributed to overfitting unstable frequency features, which in turn leads to failures when facing unknown GANs or perturbation attacks. To overcome the issue, in this chapter, we propose a novel detection framework based on multi-view reconstruction classification learning. The framework first learns multiple view-to-image reconstructors to model diverse distributions of genuine images. Frequency-irrelevant features can be learned from the view-specific distributional discrepancies characterized by the reconstructors, which are stable and robust for detecting unknown fake patterns. Then, a multi-view classification is devised with specific intra-view and intra-view learning strategies to enhance view-specific feature representation and cross-view feature aggregation, respectively. We evaluated the generalization of our framework across six popular GANs at different resolutions and the robustness against a broad range of perturbation attacks. The results show the improved effectiveness, generalization, and robustness of our method compared with various baseline detectors.

5.1 Background

Deepfakes are an emerging type of machine-synthesized media. The image generation and manipulation techniques behind deepfakes are constantly evolving thanks to the continuous advances in generative adversarial networks (GANs) [44]. The quality and fidelity of the generated images have reached a photo-realistic level that is indistinguishable from real images by human eyes. Alongside the technical advance, society is raising significant concerns regarding the abuse of these techniques to create and spread misleading information, which will cause the trust crisis that "seeing is no longer believing". To tackle the issues, the research community has been dedicated to developing powerful forensics tools against malicious deepfakes. One crucial and promising direction is detecting GAN-generated fake images considering the ubiquitous applications of GANs in image manipulation tasks.

Recent detection methods typically train CNN classifiers to learn specific features to distinguish GAN-generated fake images from real ones [32, 33, 39, 51, 94, 100], which can work perfectly in detecting clean test samples from the same GAN models used in training. However, their performances will dramatically decrease when facing samples generated by unknown GANs or noisy samples, leading to limited applicability in practice [39, 146, 158]. One primary reason is that a deep CNN classifier may easily overfit unstable GAN-specific features of the training samples, particularly the low-level frequency artifacts in the generated images. Previous studies have proved that conspicuous artifacts exist in the spectra of GAN-generated images. Despite being easily identified by classifiers, these artifact patterns are inconsistent, varying significantly among different GAN models or perturbations [39, 146]. As a result, the classifier overfitting a specific frequency pattern will lead to weak generalization ability and robustness in detecting other frequency patterns.

Based on the understanding of the overfitting issue, we are motivated to design a more generalized and robust detection model with two requirements: 1) reduce the dependency on unstable low-level frequency features; and 2) learn a robust feature representation from other types of information, such as regional consistency, color, or textural details of images. Instead of directly learning traceable features from fake images, which potentially leads to the frequency overfitting problem, we propose a novel detection framework that incorporates a multi-view reconstruction learning process and a cross-view classification learning process, as sketched in Fig. 5.1. The framework can learn a strong and stable feature representation from diverse frequency-independent,

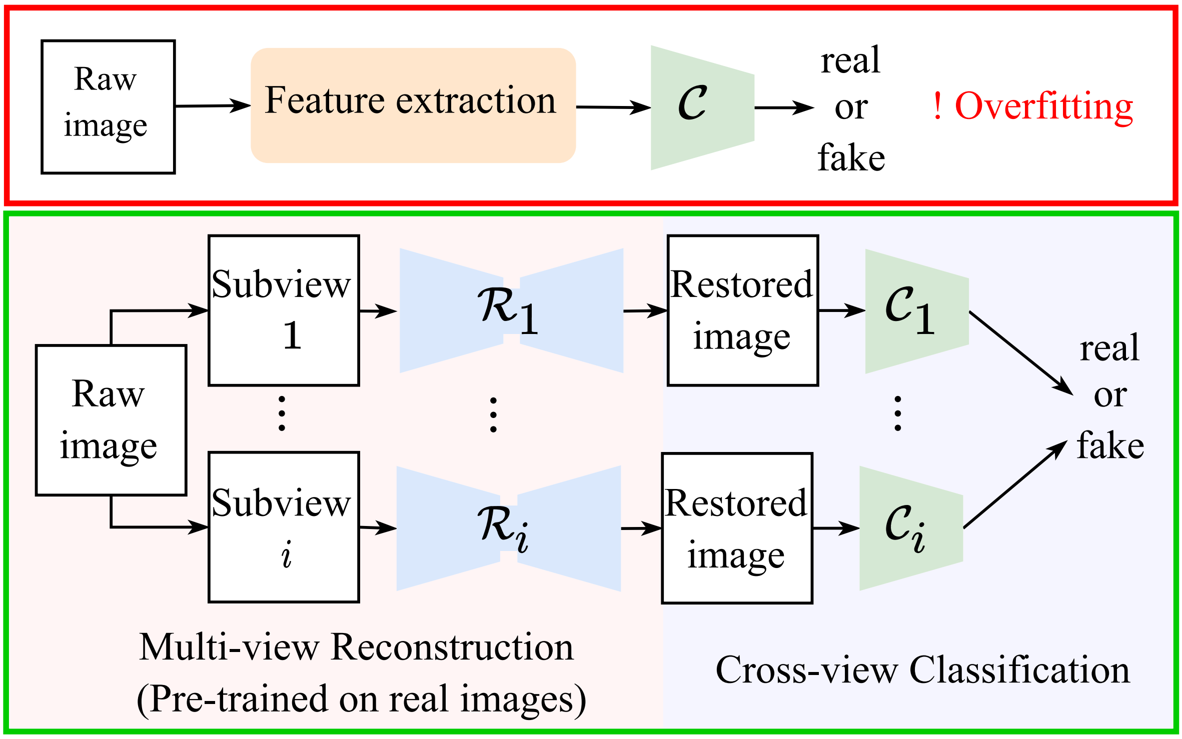


Figure 5.1: Instead of learning specific forgery features directly from fake images which may lead to overfitting, our framework incorporates multi-view reconstruction and classification to learn diverse distributional discrepancies between real and fake images, which can generalize to unknown deepfake patterns.

In the multi-view reconstruction process, multiple view-to-image reconstructors are learned *with real images only* and then used to characterize diverse distributional discrepancies between real and fake images. In contrast to overfitting specific DeepFake patterns, the compact distributions of the view-missing characteristics modeled from real images are more likely to distinguish unknown DeepFakes from real images [124]. In addition, the reconstruction can align the frequency patterns of different fake samples to that of real images, which helps reduce frequency dependency. Then, in the cross-view classification, the real and fake samples reconstructed from each view are fed into an independent classifier to learn fake detection. A multi-scale feature pyramid and a residual-guided attention module are devised to strengthen the classifier's ability to mine rich intra-view features. The independent classifiers are finally combined using an adaptive loss fusion strategy to enhance the learning from cross-view information. Our contributions are highlighted as follows:

- We propose a novel DeepFake detection framework using multi-view reconstruction classification learning to build a robust and generalized feature representation for detecting unknown GANs and perturbations.
- We devise several novel modules and learning strategies that effectively benefit the model's ability to capture and incorporate diverse view-specific features.
- We perform extensive evaluations which validate the significantly improved generalization and robustness of our framework in a wide range of settings varying in image resolutions, GAN types, and perturbation methods.

5.2 The proposed framework

We design the **M**ulti-view **R**econstruction-**C**lassification **L**earning (MRCL) to learn a novel multi-view, frequency-independent feature representation for generalized and robust detection of GAN-generated images. As shown in Figure 5.2, the framework jointly trains a set of reconstructors and classifiers. The reconstructors are trained with real images only, and each learns to recover the full image from one particular sub-view. Then, both real and fake images are processed by each reconstructor through the same view-to-image pipeline. Since the missing information is restored according to the real images' characteristics, the distributional difference between the reconstructed real and fake images can be reflected in the restored information. Then, a classifier is trained based on the reconstructed samples to capture the distributional discrepancy specific to each view. We combine the multi-scale features encoded by different layers of each reconstructor's decoder with the restored image as the classifier's input. A low-frequency residual guided attention module is employed at the entry of the classifier to highlight the stable visual difference between real and fake images. A self-adaptive loss fusion module is additionally designed to combine the decisions of multiple classifiers to facilitate inter-view learning.

5.2.1 Multi-view reconstruction learning

Several independent encoder-decoder-based reconstructors $\mathscr{R} = \{\mathscr{R}^v\}_{v=1}^N$ are trained to recover the full image from different partial views. Particularly, the reconstructors are trained only on real images, such that the recovery is governed by the characteristics of real images. We consider three reconstruction tasks: Masked Image Modeling, Gray-

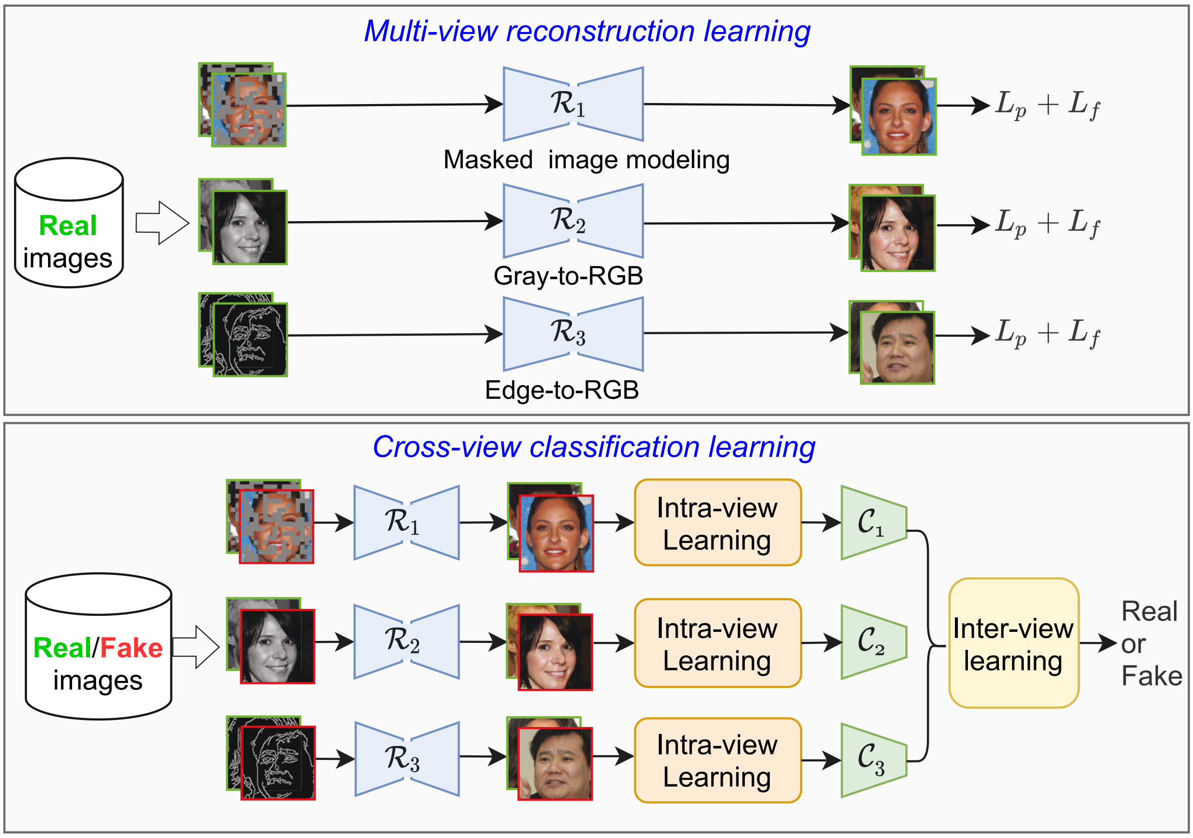


Figure 5.2: The overview of our framework. Several reconstructors first learn different distributions of real images via view-to-image reconstruction learning. Then for each view, a classifier captures the view-specific distributional discrepancy between real and fake images via intra-view learning. The classifiers are finally fused to perform inter-view learning for robust detection.

to-RGB, and Edge-to-RGB, where the missing regional details, color information, and textural information are learned by the reconstructors, respectively.

- **Masked Image Modeling** (MIM) is an emerging approach for visual representation learning [47], which masks a portion of an image and predicts the masked area. We employ MIM to model the regional consistency of natural images. The masking strategy is that, given an image $X \in \mathbb{R}^{w \times h \times 3}$, we randomly mask 50% non-overlapping patches with a patch size of $(\frac{w}{16}, \frac{h}{16})$.

- **Gray-to-RGB** aims to learn the color information from real images. We first transform the RGB image into the gray-scale version, and then predict the raw RGB pixel values from the gray input.

- **Edge-to-RGB** aims to learn the textural information from real images. We first extract the binary edge sketch from the RGB image using the Canny edge detector, and then predict the raw RGB pixel values from the edge input. Figure 5.3 shows an example of different views.

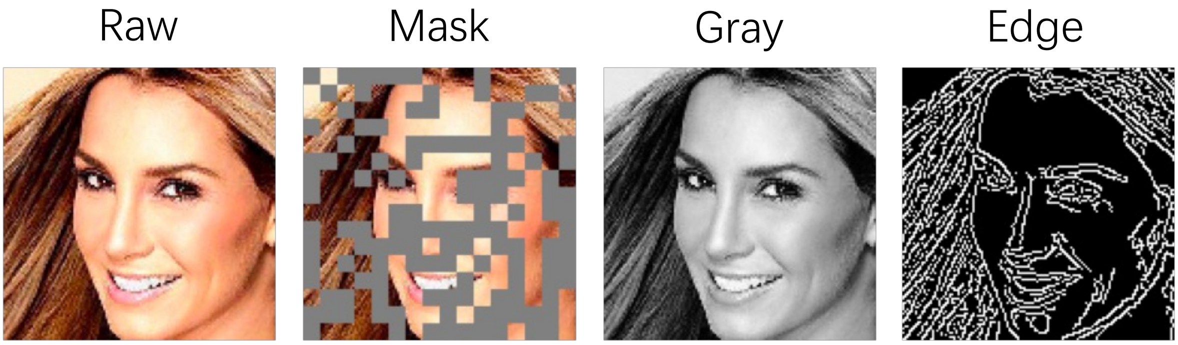


Figure 5.3: Three partial views used for reconstruction.

Mathematically, given an image X and an individual view X^v, the reconstruction is formulated as $\tilde{X}^v = \mathscr{R}^v(X^v)$. The training of $\mathscr{R}^v$ is supervised by a pixel-level regression loss

$$L_{pix} = ||X - \tilde{X}^v||_1 = ||X - \mathscr{R}^v(X^v)||_1. \tag{5.1}$$

In addition to the pixel loss, a frequency loss is employed to further enhance the ability of $\mathscr{R}^v$ in learning the frequency property of real images [60]:

$$L_{fre} = ||\mathscr{F}(X) - \mathscr{F}(\tilde{X}^v)||_2^2, \tag{5.2}$$

which computes the element-wise Euclidean distance between the 2D FFT spectra of original and reconstructed images. $\mathscr{F}(\cdot)$ denotes the 2D FFT function.

5.2.2 Intra-view classification learning

After training $\mathscr{R}^v$ with real images, both real and fake images are processed by $\mathscr{R}^v$ via the same view-to-image reconstruction workflow to enable the subsequent classification learning. In order to mine more generalized and frequency-irrelevant features from each view's pathway, we propose a multi-scale feature pyramid and a residual-guided attention module to improve intra-view feature representation, as shown in Figure 5.4.

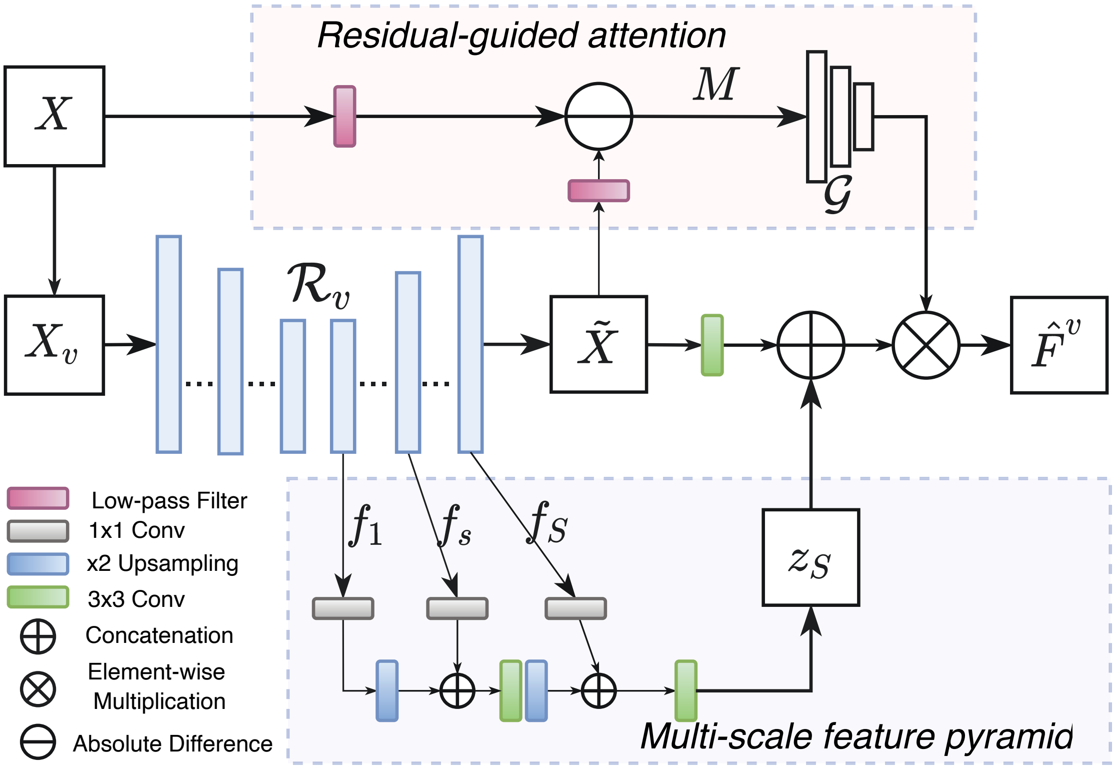


Figure 5.4: The details of the residual-guided attention and multi-scale feature pyramid modules.

5.2.2.1 Multi-scale feature pyramid

Since $\mathscr{R}^v$ is an encoder-decoder consisting of multiple layers, during the reconstruction, the missing information of the original image is progressively recovered by the layers stacked in $\mathscr{R}^v$'s decoder. Thus, the useful features for distinguishing real and fake images are embedded not only in the final output image, but also in every intermediate feature map of the decoder. To this end, we build a feature pyramid to incorporate the intermediate features at different scales. For a decoder of $\mathscr{R}^v$ with a total of S layers, let f_s be the feature map of the s-th layer, then the s-th feature of the pyramid is computed as:

$$z_s = \begin{cases} \text{Conv}_3(\text{Concat}(\text{Conv}_1(f_s), \text{Up}(z_{s-1}))), & s \geq 2 \\ \text{Conv}_1(f_s), & s = 1 \end{cases} \tag{5.3}$$

where Up$(\cdot)$ is a upsampling layer with a scaling factor of 2 to align the scales between two feature maps; Conv$_1(\cdot)$ is a 1×1 convolutional layer to reduce channel dimensions; Conv$_3(\cdot)$ is a 3×3 convolutional layer to suppress the aliasing effect of upsampling;

Concat(·) indicates the concatenation of two tensors. Finally, the last layer of the feature pyramid z_S is combined with the reconstructed image $\tilde{X}$ to get the enhanced feature F in the following way:

$$F = \text{Concat}\big(\text{Conv}_3(\tilde{X}), \text{Conv}_3(z_S)\big) \tag{5.4}$$

5.2.2.2 Residual-guided attention

The distinguishable features are contained in the restored regional, color, and textural information of the image. Thus, it is possible to leverage the reconstruction residual to provide spatial attention to improve intra-view learning. However, one challenge is that, since the original image X is involved in computing the residual, both stable and unstable features in the original image potentially remain in the residual. As discussed earlier, unstable features that are detrimental to generalization and robustness should be avoided. Prior studies have found that these unstable features are low-level artifacts that mainly cluster in high-frequency components [32, 39]. Thus, we propose only using the low-frequency residual to guide the classifier to focus on more stable features. Given an image X and its reconstructed version $\tilde{X}$, the low-frequency residual is:

$$M = |\mathscr{H}(X) - \mathscr{H}(\tilde{X})|, \tag{5.5}$$

where $\mathscr{H}(\cdot)$ is the first-order low-pass Butterworth filter and $|\cdot|$ is the absolute function. An attention mechanism is then devised to exploit the low-frequency residual. A functional network is used to process M to get the attention map, i.e., $\hat{M} = \mathscr{G}(M)$, where $\mathscr{G}(\cdot)$ consists of a 7×7 convolutional layer, an average pooling layer and a sigmoid function. The attention map is applied to the enhanced feature F in Eq. 5.4 to obtain the residual-guided feature:

$$\hat{F} = \hat{M} \otimes \text{Conv}_3(F), \tag{5.6}$$

where $\otimes$ indicates the element-wise multiplication.

5.2.3 Inter-view classification learning

When the intra-view feature enhancement is ready, we can get a set of features $\{\hat{F}^v\}_{v=1}^N$ corresponding to different views. For each view, an independent neural network classifier $\mathscr{C}^v$ is trained on the feature $\hat{F}^v$. Since the features provide view-specific information, the classifiers will learn diverse representations and contribute differently facing the same data instance. To ensure the complementarity and interactivity across different views during training, we propose a self-adaptive cross-view loss fusion strategy.

5.2.3.1 Self-adaptive loss fusion

The self-adaptive loss fusion strategy aims to combine the losses of different classifiers using adaptive weights, such that the importance of each view-specific representation can be estimated and respected in the final decision. The weights are learned and autonomously adjusted during training.

Formally, given a view-specific feature instance $\hat{F}^v$ and the corresponding label y ($y = 0$ if the the sample is a real image, otherwise 1), let p^v be the probability that the sample is fake predicted by $\mathscr{C}^v$. The training of $\mathscr{C}^v$ is supervised by the cross-entropy loss:

$$L_{ce}^v = -[y\log(p^v) + (1-y)\log(1-p^v)]. \tag{5.7}$$

The self-adaptive loss fusion strategy can be denoted as a minimization problem with respect to the weights $\boldsymbol{\beta}$:

$$\min_{\boldsymbol{\beta}} \sum_{v=1}^{N} \beta_v^\tau L_{ce}^v \quad s.t. \quad \boldsymbol{\beta}^\top \mathbf{1} = \mathbf{1}, \beta_v \geq 0, \tag{5.8}$$

where $\tau > 1$ is the power exponent parameter to avoid the trivial solution of $\boldsymbol{\beta}$ during the classification.

5.2.3.2 Optimization

The components of MRCL that require optimization include the parameters of $\{\mathscr{R}^v\}_{v=1}^N$, $\{\mathscr{C}^v\}_{v=1}^N$ and several building blocks for intra-view learning (for simplicity, the latter two are denoted in together as $\{\mathscr{C}^v\}_{v=1}^N$), as well as the self-adaptive loss weights $\boldsymbol{\beta}$. The optimization is performed in the following alternative way:

- **Update network parameters.** The reconstruction and classification networks with respect to different views are updated independently in parallel. For the view v, $\mathscr{R}^v$ and $\mathscr{C}^v$ can be updated in an end-to-end mode by optimizing the following objective function:

$$\min L_{ce}^v + \lambda_1 L_{pix}^v + \lambda_2 L_{fre}^v, \tag{5.9}$$

where λ_1 and λ_2 are weights to balance different losses. During the optimization, the loss weights $\boldsymbol{\beta}$ are fixed.

- **Update loss weights $\boldsymbol{\beta}$.** Next, we fix the parameters of $\{\mathscr{R}^v\}_{v=1}^N$ and $\{\mathscr{C}^v\}_{v=1}^N$, and update $\boldsymbol{\beta}$ by solving Eq. 5.8. To satisfy the constraints in Eq. 5.8, the Lagrangian function of Eq. 5.8 is

$$\mathscr{L}(\boldsymbol{\beta},\zeta) = \sum_{v=1}^{N} \beta_v^{\tau} L_{ce}^{v} - \zeta \left(\sum_{v=1}^{N} \beta_v - 1 \right) \tag{5.10}$$

where ζ is the Lagrange multiplier. By derivation of Eq. 5.10 with respect to β_v and ζ, the optimal solution of Eq. 5.8 is:

$$\beta_v = (L_{ce}^{v})^{\frac{1}{1-\tau}} / \sum_{n=1}^{N} (L_{ce}^{n})^{\frac{1}{1-\tau}} \tag{5.11}$$

5.3 Experiments

5.3.1 Datasets

Real images Our experiments are conducted on facial images, given that the human face is the primary target of deepfake. We choose the large-scale facial image dataset CelebA [93] and its high-quality version CelebA-HQ [67] to perform evaluations at different resolutions. The image resolution is 128×128 in CelebA while 1024×1024 in CelebA-HQ.

GAN-generated images A total of six popular GAN types are considered, including ProGAN [67], CramerGAN [9], SNGAN [106], MMDGAN [81], StyleGAN [68] and StyleGAN2 [69]. In the low-resolution setting, we follow the setting in [158], using the pre-trained ProGAN, CramerGAN, SNGAN, and MMDGAN models [1] to generate fake faces. All the four GANs are pre-trained with CelebA. In the high-resolution setting, we adopt the dataset released by [49] [2], which includes images generated by ProGAN, StyleGAN, and StyleGAN2. Note that the ProGAN and StyleGAN are pre-trained with CelebA-HQ while the StyleGAN2 with another facial image dataset FFHQ [68]. Since FFHQ has a larger variety in facial attributes compared with CelebA-HQ, StyleGAN2 is included for cross-domain evaluation. Table 5.1 shows the details of dataset setting.

5.3.2 Implementation Details

The reconstructors and classifiers are implemented based on U-Net [120] and Xception [27], respectively. The U-Net we use has five skip connection blocks (i.e., $S = 5$), and their

[1] `https://github.com/ningyu1991/GANFingerprints`

[2] `https://github.com/SSAW14/BeyondtheSpectrum`

Table 5.1: The details of the experimental dataset setting.

128x128	CelebA	ProGAN	CramerGAN	SNGAN	MMDGAN
Training	60,000	60,000	–	–	–
Test	6,000	6,000	6,000	6,000	6,000
1024x1024	CelebA-HQ	ProGAN	StyleGAN	StyleGAN2*	
Training	25,000	25,000	25,000	–	
Test	2,500	2,500	2,500	2,500	

* Pre-trained in a different real image dataset FFHQ

output feature maps are employed to build the feature pyramid. We train the whole framework with a batch size of 80 using the Adam optimizer [73]. The initial learning rate is 1e-3, and we reduce it to half after every ten epochs. τ in Eq. 5.8, and λ_1 and λ_1 in Eq. 5.9 are empirically set to 4, 0.1, 1, respectively. We also use random Gaussian noise, color jitter, and blurring for data augmentation at the reconstructor side.

5.3.3 Baseline Detection Models

We compare our method with two representative general detectors, including an image-domain detector using GAN fingerprints (GAN-FP) [158] and a frequency-domain detector based on 2D DCT coefficients (2d-DCT) [39], as well as three state-of-the-art detection methods specific to improve generalization and robustness, including the data augmentation-based method (DA) [146], frequency-level perturbation (FLP) [59], and super-resolution re-synthesis (SRR) [49]. To evaluate the detection performance, we report the classification accuracy (Acc.) and the average precision score (A.P.) commonly used in related studies [59, 146].

5.3.4 Results of Generalization

5.3.4.1 Low-resolution setting

In the low-resolution setting, we train the detection model with the CelebA images and the corresponding 128 × 128 ProGAN images and test it with the ProGAN, CramerGAN, SNGAN, and MMDGAN images to evaluate the cross-GAN generalization ability. The results with comparison to five baselines are listed in Table 5.2. We conclude that: 1) The general detectors, GAN-FP and 2d-DCT, are highly accurate for within-distribution detection, but performance degrades significantly for cross-GAN detection, implying the risk of overfitting to unstable features; 2) By augmenting or perturbing the original frequency distribution of training images, the other four detectors all get the cross-GAN detection performance improved. By comparison, the proposed method MRCL achieves the best or second-best results for all GANs, thanks to the multi-view representation enriching the robust features.

Table 5.2: The results of cross-GAN detection in the 128 × 128 setting. **Bold** indicates the best score in each column.

	ProGAN		CramerGAN		SNGAN		MMDGAN	
	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.
GAN-FP	99.5	99.8	52.1	55.4	53.6	70.4	48.2	53.2
2d-DCT	98.9	99.1	70.2	67.1	61.9	73.5	56.0	73.1
DA	99.5	99.9	72.1	78.3	63.1	70.0	54.7	71.7
FLP	95.1	98.3	81.3	81.7	**83.6**	80.1	70.2	82.0
SRR	**100.**	**100.**	88.2	**95.1**	70.3	81.5	77.7	84.5
MRCL (Ours)	**100.**	**100.**	**91.1**	89.2	80.2	**83.3**	**85.4**	**86.1**

5.3.4.2 High-resolution setting

We conduct the high-resolution detection test following the setting in [49]: a ProGAN detector and a StyleGAN detector are trained with CelebA-HQ and the corresponding GAN images independently. Then we test the two detectors with ProGAN, StyleGAN, and StyleGAN2 test samples to evaluate the within-distribution, cross-GAN and cross-domain performances, respectively. The results are summarized in Table 5.3. Similar to the results in the low-resolution setting, all detectors perform well for within-distribution detection. In the cross-GAN group, we notice that 2d-DCT becomes more generalized when detecting high-resolution images. The reason may be that with the resolution increasing, low-frequency visual information becomes richer while high-frequency noise changes less. Thus, 2d-DCT trained directly with the spectrum input can capture more stable low-frequency features. We can also see that FLP, SRR, and MRCL improve more significantly than DA in this group because they reduce unstable frequency features in a learnable way. Regarding the cross-GAN & cross-domain group, which is the most challenging, our method remarkably outperforms all baseline methods in both sub-groups, indicating great applicability to difficult detection scenarios.

Table 5.3: The results of cross-GAN detection in the 1024 × 1024 setting. **Bold** indicates the best-in-column. P, S, S2 are short for ProGAN, StyleGAN and StyleGAN2, respectively. The right and left sides of → indicate the training and test sets, respectively.

	Within-distribution				Cross-GAN				Cross-GAN & Cross-domain			
	P→P		S→S		P→S		S→P		P→S2		S→S2	
	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.
GAN-FP	99.9	99.9	99.4	99.6	51.1	71.0	49.3	68.8	44.3	47.6	48.0	46.9
2d-DCT	99.9	99.9	99.8	99.9	90.1	91.5	93.0	92.1	62.1	60.0	93.8	90.2
DA	97.6	96.6	98.3	97.8	73.2	87.7	78.1	73.1	66.1	79.1	80.7	84.4
FLP	98.9	99.0	99.1	98.9	95.0	97.1	94.3	86.3	80.8	88.0	92.4	93.1
SRR	**100.**	**100.**	99.9	99.9	**99.1**	**99.4**	98.2	98.1	88.2	80.3	91.5	91.1
MRCL (Ours)	**100.**	**100.**	**100.**	**100.**	98.1	95.5	**99.2**	**98.9**	**95.3**	**90.0**	**97.7**	**96.0**

5.3.5 Results of Robustness

We evaluate the robustness against perturbations using the 128 × 128 ProGAN detectors. We train detectors with the CelebA and ProGAN images, and test them with perturbed ProGAN samples. Unlike prior work mainly concerning common image manipulations [39, 146, 158], we investigate a broader range of perturbations as follows:

- Common manipulations including Blurring, Cropping, Compression, Noising and a mix of all. We follow the setting in [39] to created the perturbations.
- Adversarial attacks including FGSM [46] and PGD [98]. The adversarial example are crafted based on a vanilla Xception detector with the noise amount $\epsilon = 8/255$.
- Spectrum Difference Normalization (SDN) [31], an attack specific to GAN-generated images that calibrates the spectra of fake images according to real images.

An example of samples modified by different perturbations is shown in Figure 5.5. Table 5.4 shows the results. Since most perturbations significantly modify the original frequency distribution of fake samples, the performance of general detectors degrades rapidly, while the other four are relatively more resistant given the reduction of frequency overfitting. Among the four robust methods, our method achieves the best results regarding all perturbations except for the A.P. scores for blurring and compression. In addition, it is worth noting that for much more challenging perturbations such as the adversarial attacks FGSM and PGD and the specific attack SDN, the Acc. and A.P. scores of our method are notably higher than other baselines.

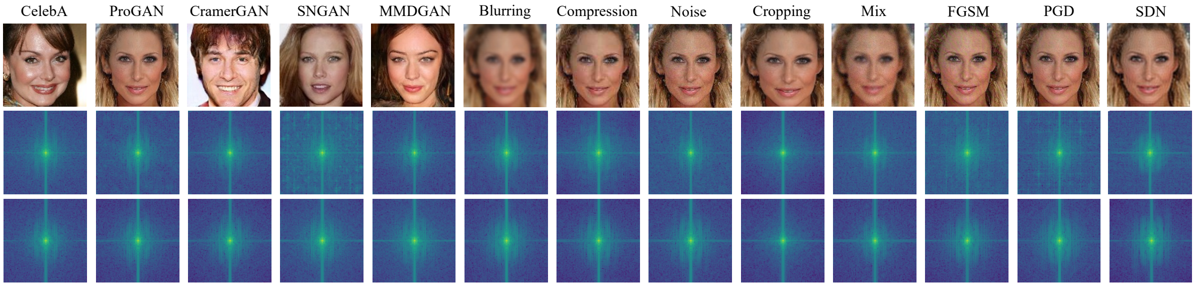


Figure 5.5: The visualization of different deepfake samples (the 1st row) and the average FFT spectra before (the 2nd row) and after (the 3rd row) the Edge-to-RGB reconstruction.

Table 5.4: The results of robustness against 8 perturbation methods. **Bold** indicates the best score in each column.

	Clean		Blurring		Cropping		Compression		Noise		Mix		FGSM		PGD		SDN	
	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.
GAN-FP	99.5	99.8	49.6	67.4	44.9	77.5	8.7	45.8	9.0	49.1	19.3	66.6	11.1	15.5	8.1	22.1	13.4	45.0
2d-DCT	98.9	99.1	60.4	77.7	80.5	76.1	67.4	80.2	46.7	74.3	61.3	61.8	34.0	45.3	23.1	41.3	21.8	56.1
DA	99.5	99.9	83.2	**98.9**	51.8	64.1	84.0	**97.3**	74.3	80.2	85.5	91.0	43.4	66.7	40.1	54.4	56.7	67.0
FLP	95.1	98.3	96.1	90.2	71.6	77.0	80.3	74.3	90.9	91.1	84.7	89.9	56.1	60.7	49.4	67.0	43.2	60.1
SRR	**100.**	**100.**	92.1	93.0	97.9	96.1	90.7	93.3	92.0	88.8	89.6	90.6	67.1	75.2	64.8	77.1	87.2	91.1
MRCL (Ours)	**100.**	**100.**	**96.4**	98.5	**98.2**	**99.1**	**93.8**	96.9	**94.7**	**94.4**	**91.3**	**94.4**	**81.6**	**80.3**	**81.3**	**81.9**	**93.2**	**95.6**

5.3.6 Discussion

5.3.6.1 Ablation study

Two ablation studies are performed to show the effects of different views and different devised modules, respectively. Evaluations are conducted using the 128 × 128 ProGAN detectors, and we report the average Acc. (mAcc.) and A.P. (mA.P.) scores for cross-model and cross-perturbation performance. Table 5.5 shows the results under different conditions. Regarding different view settings, it shows that by adding only one view, the generalization and robustness increase significantly compared with using a single view. In general, increasing the number and diversity of views will constantly improve the detection performance, indicating that the model is able to capture and fuse different types of view-specific features for generalized and robust detection. A similar trend can be observed in the ablation study of different modules. With activating more modules, the learning capacity of the model improves, enabling more effective feature representation. In addition, we highlight that the proposed framework is fully flexible and extensible with regard to the view settings, which means that it is possible to incorporate more quantities and types of views within the framework to enable stronger feature representations.

Table 5.5: The results of ablation studies with different views or modules. MFP: Multi-scale Feature Pyramid. RGA: Residual-guided Attention. ALF: Adaptive Loss Fusion.

			Within-distribution		Cross-GAN		Cross-perturbation	
			Acc.	A.P.	mAcc.	mA.P.	mAcc.	mA.P.
Masked	Gray	Edge						
✓			99.1	99.3	67.2	72.1	71.1	77.0
✓	✓		100.0	100.0	78.9	83.4	88.8	90.1
✓	✓	✓	100.0	100.0	85.6	86.2	91.3	92.6
MFP	RGA	ALF						
✓			95.1	97.3	79.1	73.0	86.8	79.9
✓	✓		97.5	98.9	81.2	76.1	90.5	82.3
✓	✓	✓	100.0	100.0	85.6	86.2	91.3	92.6

5.3.6.2 Frequency Analysis

One advantage of multi-view reconstruction learning is that it helps reduce the classifier's reliance on unstable frequency patterns by aligning the frequency distributions between real and fake samples. This is because the low-level frequency artifacts of fake samples are prior removed in the partial views. Then during reconstruction, along with the restoration of the view-missing information, the frequency pattern is calibrated according to real images. We provide a spectral analysis to confirm the effect of frequency alignment.

The azimuthal integration over each radial frequency of the center-shifted 2d-FFT spectrum [32] is used to estimate the spectral distribution. Figure 5.6 shows the averaged azimuthal integration curves of real images and images generated by different GANs before and after reconstruction. It can be observed that the original distributions differ significantly between real and fake images and between different GANs. Thus, a CNN classifier easily overfits one specific frequency pattern for detection and can not generalize to another. After reconstruction, the gaps between different frequency patterns are much closer regarding all views. The alignment is more thorough in the edge-to-RGB reconstruction than in the other two due to edge sketches containing far less information than masked and gray views. These frequency-aligned training samples will force the classifier to focus on more stable, generalized, and frequency-insensitive discernible features. Figure 5.5 additionally provides a visualization of the averaged FFT spectra of different samples before and after the edge-to-RGB reconstruction. The difference between real images and all types of fake images becomes smaller after reconstruction, except for the clipping, which changes more in semantic contents than in frequency.

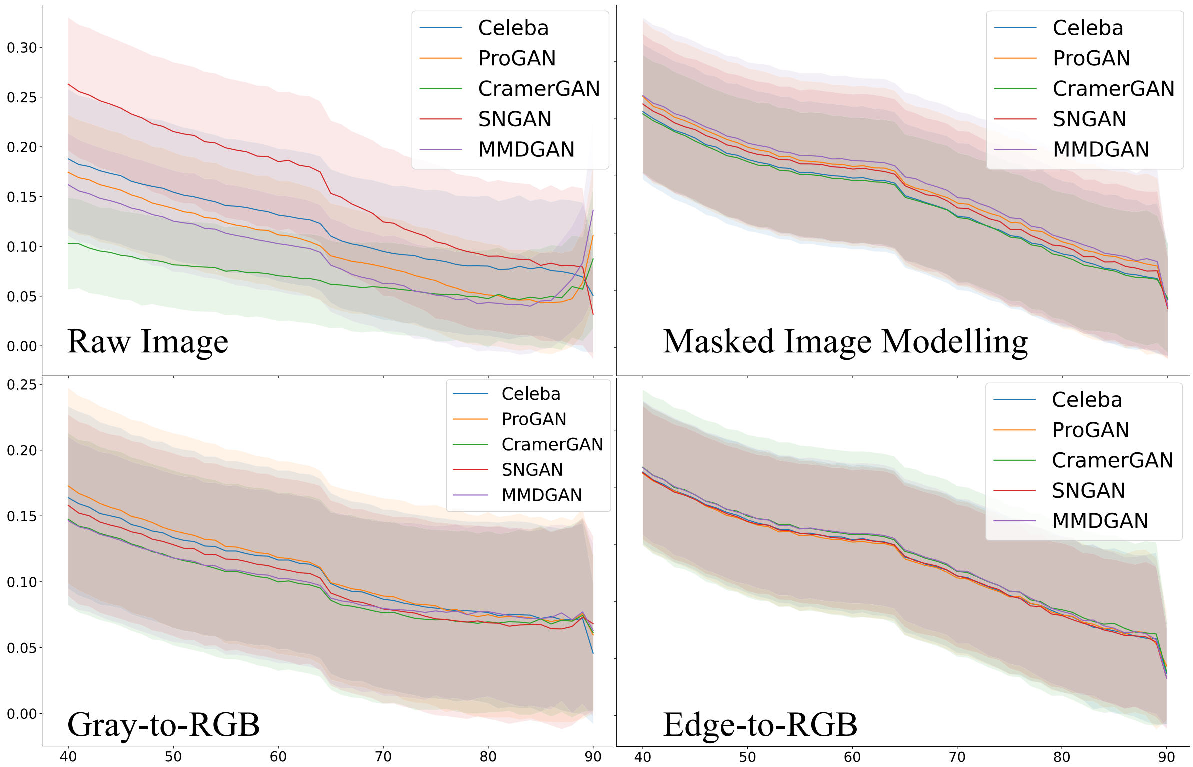


Figure 5.6: The spectral distributions of real images and fake images generated by different GANs. The distributions are successfully aligned in all reconstruction tasks.

5.3.6.3 Residual Analysis

One assumption of the low-pass residual-guided attention module of intra-view classification learning is that discriminative features are contained in the restored regional, color, and textural information of the image, which can be potentially reflected in the low-frequency reconstruction residuals. In addition to the evidence confirmed in the main experiments, we conduct a residual analysis interpreting the residual differences between real and fake images to verify the assumption. Figure 5.7 provides several visualization examples resulting from the three image completion tasks. We can see that the reconstruction residuals differ significantly between real and fake images with regard to all reconstruction models. Moreover, we compute the histograms of the average spatial amounts of reconstruction residuals of real and ProGAN images, as shown in Figure 5.8. Clear distributional gaps exist in reconstruction residuals between real and ProGAN images, which further confirms the assumption.

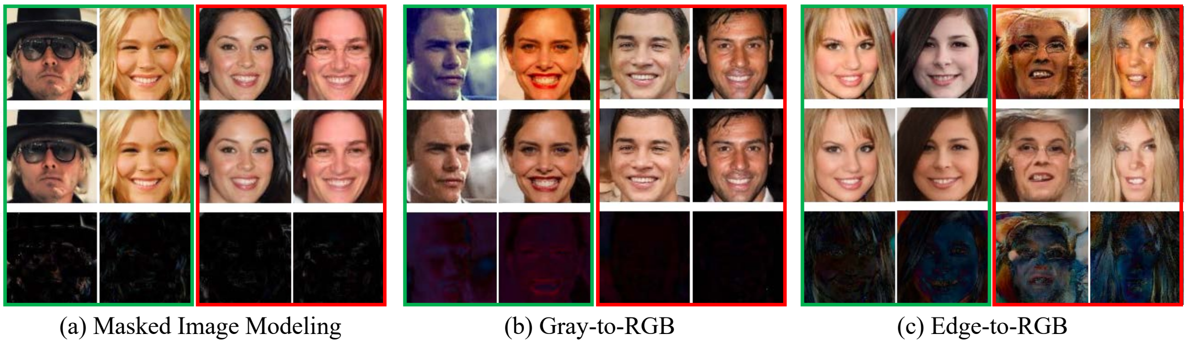


Figure 5.7: Visualization examples of residual differences between real (green box) and fake (red box) images generated by ProGAN for the three reconstruction models. From first to last row: original images, reconstructed images, and the corresponding residuals.

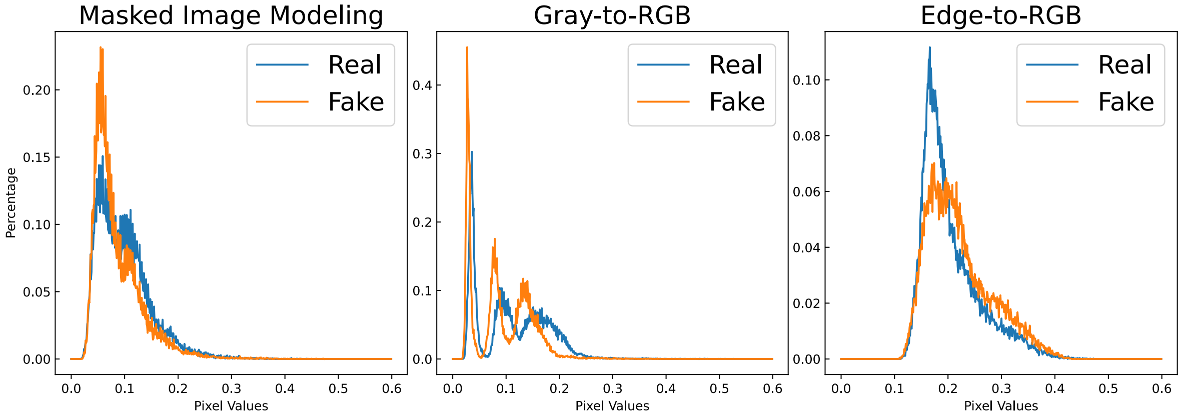


Figure 5.8: Histograms of the average spatial amounts of reconstruction residuals on CelebA. Clear margins exist between distributions of real and fake (ProGAN) images.

5.4 Summary

The generalization and robustness of detecting GAN-generated images are two critical challenges when countering unknown deepfakes outside the training dataset. Prior methods relying on unstable GAN-specific frequency features fail to generalize to other deepfake patterns. We proposed a novel detection framework, which jointly learns a reconstruction and classification streams for a robust multi-view feature representation from diverse frequency-irrelevant, view-specific distributional disparities between real and fake images. Numerous experiments with varying cross-resolution, cross-GAN, and cross-perturbation settings validated the outperforming generalization and robustness of our proposed framework compared with the current state-of-the-art detectors. We also confirmed the effect of reducing frequency reliance in deepfake detection, offering a potential route for future designs of robust deepfake detection.

Part III

Investigation on anti-forensic techniques

Chapter 6

The Trace Removal Attack

DeepFakes are raising significant social concerns. Although various DeepFake detectors have been developed as forensic countermeasures, these detectors are still vulnerable to attacks. Recently, a few attacks, principally adversarial attacks, have succeeded in cloaking DeepFake images to evade detection. However, these attacks have typical detector-specific designs, which require prior knowledge about the detector, leading to poor transferability. Moreover, these attacks only consider simple security scenarios. Less is known about how effective they are in high-level scenarios where either the detector's defensive capability or the attacker's knowledge varies. In this chapter, we aim to solve the above challenges with presenting a novel attack pattern for DeepFake anti-forensics, namely, the trace removal attack. Instead of investigating the detector side, this trace removal attack looks into the original DeepFake creation pipeline, attempting to remove all detectable natural DeepFake traces to render the fake images more "authentic". This detector-agnostic design benefits the attack to be effective against arbitrary or even unknown detectors. To implement this attack, we first perform an in-depth DeepFake trace discovery, which identifies three discernible traces: spatial anomalies, spectral disparities, and noise fingerprints. Then a trace removal network (TR-Net) is proposed based on an adversarial learning framework that involves one generator and multiple discriminators. Each discriminator is responsible for one individual trace representation to avoid cross-trace interference. These multiple discriminators are arranged in parallel, which prompts the generator to remove various traces simultaneously. To evaluate the efficacy of the attack, we crafted heterogeneous security scenarios where the detectors were embedded with different levels of defense and the attackers' background knowledge of data varies. The experimental results show that the proposed attack can significantly compromise the detection accuracy of six state-of-the-art DeepFake detectors while causing only a negligible loss in visual quality for the original DeepFake samples.

6.1 Background

Along with the recent progress in automated digital face manipulation techniques based on deep learning, deep face forgeries, also known as DeepFakes, are raising serious social concerns for information security [149]. Accordingly, the research community is dedicated to developing forensic countermeasures against DeepFakes, and many DeepFake detectors have been developed that can successfully distinguish DeepFake images from real ones [137]. However, the robustness of these detectors against malicious attacks is still in the early stages. To further understand the vulnerability of DeepFake detectors, researchers have and must continue to engage in anti-forensics against DeepFake detection [17, 29, 34, 42, 52, 53, 55, 88, 109, 111, 147]. Each novel anti-forensic attack exposed can help us to analyze these detectors more comprehensively.

Most existing attacks are based on adversarial attacks that embed imperceptible adversarial perturbations into DeepFake samples to fool machine learning-based detectors [17, 34, 42, 55, 88, 109, 147]. The development of this type of attack relies on the background knowledge of the detectors, such as the queried outputs and the detector's parameters. Even in a universal black-box attack scenario, information from surrogate detectors is always needed to imitate the behavior of the target detector. These *detector-specific* designs lead to poor transferability and a lack of stability across different detectors or unknown detectors [4, 160]. For example, the attack success rate of a typical adversarial attack FGSM will decrease significantly from 100% to only 0.8% when the target detector changes, as proved by Barni et al. [4]. Other attacks emerging in this field generally require reconstructing DeepFake samples to modify the distribution of feature-of-interest of the target detector to evade detection [29, 52, 53, 111]. This is also a detector-specific design, which means that these attacks are less transferable to detectors interested in different forgery features. Moreover, these attacks only pay attention to a single type of feature. Their efficacy may deteriorate significantly against advanced detectors that operate on hybrid features.

Another weakness of these attacks is that their studies tend to oversimplify the security scenarios. On the one hand, the attacks are often implemented and evaluated

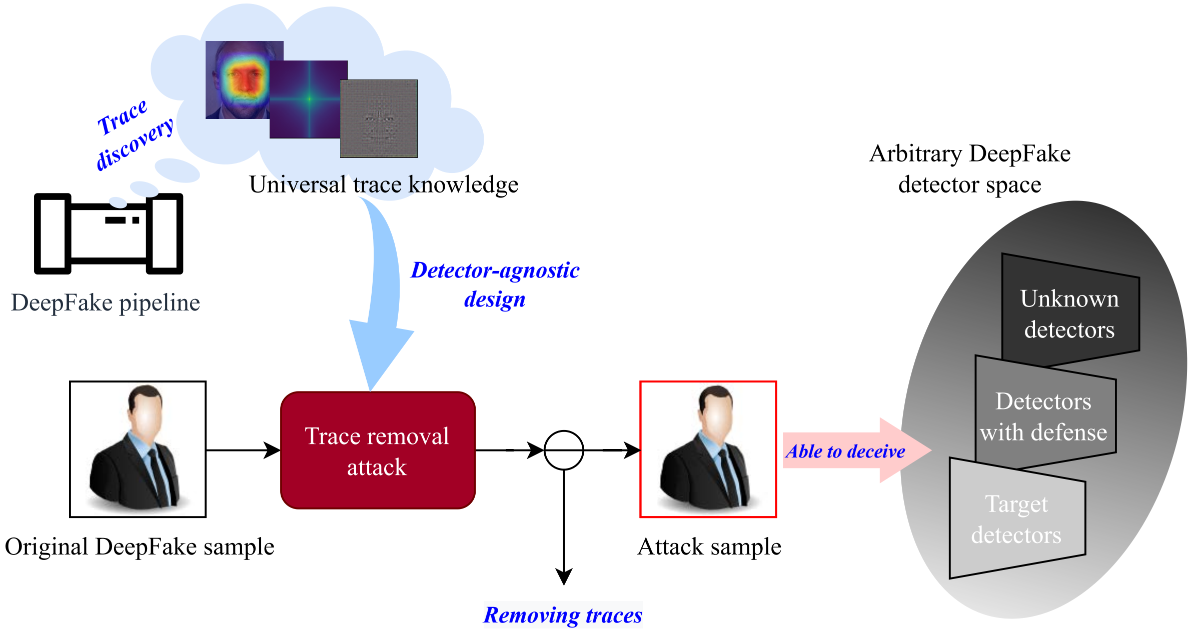


Figure 6.1: The proposed trace removal attack utilizes the universal trace knowledge distilled from the common DeepFake pipeline instead of detector-specific knowledge. Thus it is detector-agnostic and can transfer across arbitrary black-box detectors.

with ideal assumptions, e.g., the attacker has unlimited access to the target detector (or at least the surrogates) or the attacker has all the required background knowledge of the training data. On the other hand, the target detectors are often assumed to be as naked as possible, while some common and easy-to-implement defenses are left out of consideration.

In this chapter, we propose a novel attack pattern for DeepFake anti-forensics, called the *trace removal attack*, that addresses the above weaknesses. Unlike the detector-specific designs, we offer a novel *detector-agnostic* perspective. As shown in Figure 6.1, we pay full attention to the original pipeline of DeepFake image creation, identifying the discernible manufacturing traces in the DeepFake images. The DeepFake images are then refined by removing all these traces, resulting in images (i.e., attack samples) that are able to bypass any arbitrary detector. Our attack requires zero knowledge of the target detector, operating exclusively on the DeepFake images without any additional interactions with the target detector. In contrast to adding extra adversarial noise or modifying the feature distribution, removing the intrinsic detectable traces makes the DeepFake images essentially much closer to the real ones, i.e., the DeepFake images become more natural and perceptually "authentic". In this sense, the proposed method can be seen as a universal black-box attack.

To implement the trace removal attack, the first step is to conduct empirical trace discovery to thoroughly investigate what discernible manufacturing traces are naturally maintained in DeepFake images. An adversarial learning-based trace removal network (TR-Net) then removes the traces found. However, unlike a normal adversarial learning network with one generator and one discriminator, TR-Net contains a single generator and multiple discriminators, where each discriminator is responsible for distinguishing one particular type of trace. This "one-versus-multiple" structure can prompt the generator to reconstruct DeepFake images by removing all possible traces synchronously. Considering that the identified traces could exist in different signal domains, using multiple discriminators allows the representation of these traces to be effectively decoupled. We construct several heterogeneous threat scenarios to assess the efficacy of the attack, where the detectors are reinforced through various defensive strategies, and the attackers have different data background knowledge. We then evaluate the attack on a wide range of representative detectors to ensure that this detector-agnostic attack is truly universal and transferable.

Our contributions are as follows:

- We perform an in-depth DeepFake trace discovery, identifying three universal traces responsible for DeepFake images' tractability.
- We propose a novel attack concept against DeepFake detectors, namely, the trace removal attack. Benefiting from a detector-agnostic design, our attack can defeat arbitrary unknown detectors and detectors equipped with defenses. The attack is implemented via a "one-versus-multiple" adversarial learning network that erases all traces synchronously.
- The attack is tested in heterogeneous threat scenarios, where the detector's defensive capability ranges from weak to strong and the attacker's data knowledge is limited. Furthermore, performance is evaluated on a wide range of detectors, and a dataset is developed covering all typical DeepFake types to benchmark our evaluation.

6.2 DeepFake trace discovery

In this section, we investigate the original process that creates DeepFakes to provide insights into the root causes that make DeepFakes detectable. In this way, the universal forgery traces can be identified empirically.

6.2.1 The common DeepFake pipeline

DeepFakes are generated in roughly one of three ways: face synthesis, facial attribute editing, or face replacement [137]. Face synthesis means creating an entire non-existent face from random noise with an unconditional GAN, such as ProGAN [67] and StyleGAN [68]. With facial attribute editing, an image's attributes are altered. Either the appearance attributes (e.g., hair color, makeup, skin color, etc.) or the soft biometric attributes (e.g., identity, gender, age, etc.) can be modified. Conditional GANs, such as StarGAN [26] and STGAN [92], are widely employed for such tasks. Here, the target attribute serves as the extra label y in training a conditional GAN. Face replacement swaps the face of a target image with that of a source image. Factors that need to be considered include the alignment of the face in terms of size, pose, and direction. A deep rendering process then ensures the resulting image looks natural and seamless. In addition, these methods can be combined to produce high-level DeepFakes, like high-fidelity facial reenactments for fake videos.

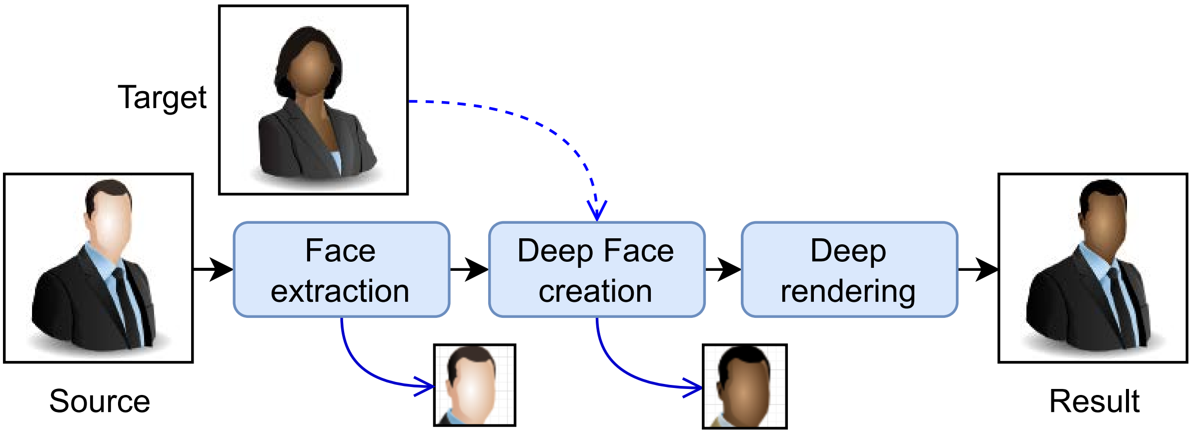


Figure 6.2: The fundamental DeepFake pipeline.

The top-level design of a DeepFake generator may vary, but underneath there is a common pipeline for producing DeepFake images that consists of three core stages: face extraction, fake face creation, and deep rendering, as shown in Figure 6.2. In the first stage, the facial region is localized and extracted from the source image. This process can be accomplished without a GAN. Next, a fake face is generated by a specific GAN or a model template according to the target face. In most cases, creating a fake face is conditioned upon some knowledge of the target face, such as the identity or attributes. In the last stage, the generated face is aligned to the source face and composed back onto the source image. Usually, a GAN or some post-processing operations are employed to render the final image to make it more natural.

This pipeline is applicable to all the aforementioned methods of generating a DeepFake, either partially or entirely. For example, end-to-end facial synthesis and attribute editing with DeepFake models are *de facto* productions from the second stage of the pipeline, while more sophisticated DeepFake models are equipped with a deep rendering process at the end of the pipeline.

6.2.2 What make DeepFakes detectable?

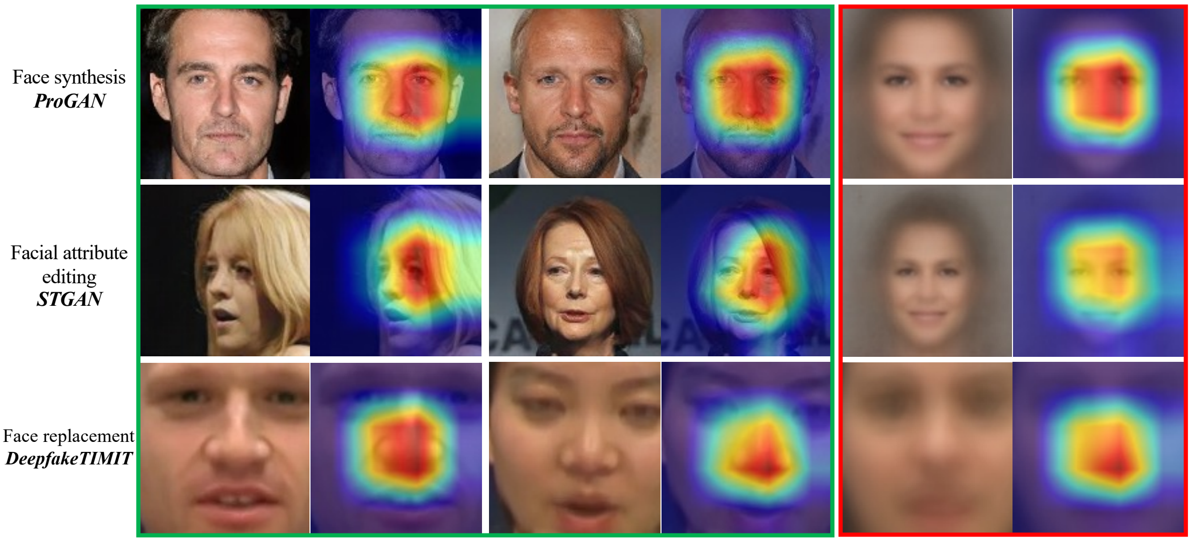


Figure 6.3: Spatial anomalies revealed by the spatial attention maps of a Xception detector. For each DeepFake type, the green box shows two random DeepFake examples and the red box shows the average results of 2000 samples.

Spatial anomalies. DeepFakes rely on deep generative models, such as GANs, to synthesize faces from real face images. Ideally, the generated faces should be visually indistinguishable from real ones. However, due to some practical limitations e.g., with the dataset or the model's capability, the fake faces may be imperfect which may show spatial abnormalities. Although the latest GANs have seen a significant improvement in visual quality over their predecessors, some subtle unnatural traces such as inconsistencies in texture or contextual discrepancies can still occur [21, 82, 94, 102, 113]. Spatial anomalies can also be found in faces generated by model templates. This can be a result of manufacturing failures during the template alignment or rendering [83, 85].

Notably, since the subtle spatial anomalies may be imperceptible to humans but can be captured by machines, we demonstrate their existence and spatial distributions in the RGB color space with the spatial attention map (SAM) of a toy Xception detector. Grad-CAM [126] is used to calculate the SAMs regarding different DeepFake types (details of these DeepFake types and the Xception detector are introduced in Section 6.4). As shown in Figure 6.3, there are evident detectable traces in the RGB space, and their distributions exhibit certain stable semantic-dependency: the ProGAN and STGAN's anomalies are around the middle-right face region, while DeepfakeTIMIT images expose traces concentrated in the nose area. These results can be seen more clearly in the averaged faces.

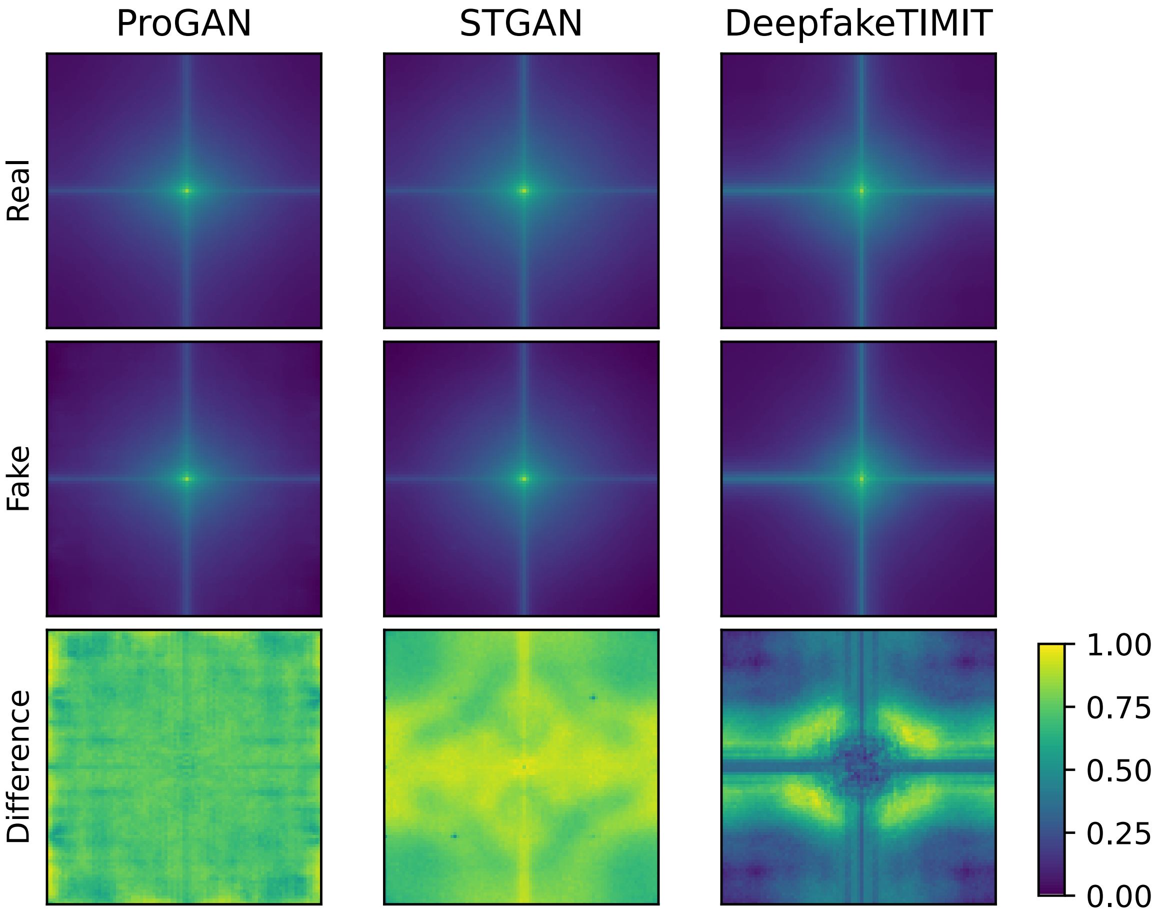


Figure 6.4: The averaged spectra of both the real images and the corresponding fake images of different DeepFake types. Each spectrum is averaged on 2000 samples. The last row shows the differences between the spectra of the real and fake images.

Spectral disparity. Detectable traces can also be revealed in the frequency domain.

This is because that CNN-based generative models, typically GANs, will create disparities in the spectra of the generated images. Some past studies attribute this phenomenon to the transposed convolution operation, a widely-used upsampling unit in CNN-based generative models for increasing feature dimensionality [20, 32, 39, 91, 159].

Claim 6.1 (1)**.** *The transposed convolution operation in upsampling layers leads to quasi-periodic high-frequency artifacts in the resulting feature maps. (The proof is in Chapter 4.)*

An illustration of the disparity between the averaged spectra of real and fake images is provided in Figure 6.4. All three types of DeepFake images have significant differences from the real ones. The disparity patterns in the DeepfakeTIMIT images are not similar to the other two because face replacement involves post-processing procedures that further change the spectral distribution.

6.2.2.2 Model traces in deep rendering

Noise fingerprint. A DeepFake may retain two types of manufacturing fingerprints through the deep rendering phase, one being the GAN fingerprint, the other one the post-processing fingerprint. The deep rendering phase usually involves a GAN-based rendering model and some post-processing operations, such as landmark alignment, color correction, splicing, and blending. It has been pointed out that GANs maintain unique and stable fingerprints in their generated images. Likewise, post-processing operations will also introduce fingerprints, due to the characteristic discrepancies in the noise space brought about through tampering the regions.

To show the fingerprints in the noise space, we estimate the fingerprints of different DeepFake types using the average noise residual. Specifically, the noise fingerprint of a DeepFake model $\mathcal{M}$ can be formulated as:

$$F^{\mathcal{M}} = \frac{1}{N}\sum_{i=1}^{N}(I_i^{\mathcal{M}} - W(I_i^{\mathcal{M}})), \tag{6.1}$$

where $I_i^{\mathcal{M}}$ is a sample generated by $\mathcal{M}$, $W(\cdot)$ is a Wiener denoising filter and $N = 2000$ in our case. For comparison, we also used the same approach to calculate the average noise residual of an equal quantity of real images, as shown in Figure 6.5. For all DeepFake types, the patterns of the average noise residuals between the real and fake samples are significantly different. Those of the real images have generally a smoother response than those of the fake images, which corroborates the fact that extra noise discrepancies are introduced into DeepFakes during their production. A distributional difference can also be demonstrated by calculating the normalized cross-correlation (NCC) between the average noise residual and the individual noise residuals from another 2000 real/fake samples:

$$\rho_i^{\mathcal{M}} = \frac{<F^{\mathcal{M}}, R_i^{\mathcal{M}}>}{\|F^{\mathcal{M}}\| \cdot \|R_i^{\mathcal{M}}\|}, \tag{6.2}$$

where $< \cdot, \cdot >$ and $\| \cdot \|$ denote the inner product and l_2-norm respectively; $R_i = I_i^{\mathcal{M}} - W(I_i^{\mathcal{M}})$. Figure 6.6 shows histograms of the individual correlations. For all DeepFake types, the NCC scores between the noise residuals of the real samples and the fingerprints of the DeepFakes are distributed around zero. This indicates that little correlation exists. By contrast, the NCC scores between the noise residuals of the fake samples and the fingerprints of the DeepFakes are remarkably larger than zero, testifying to a significant correlation with the corresponding fingerprint.

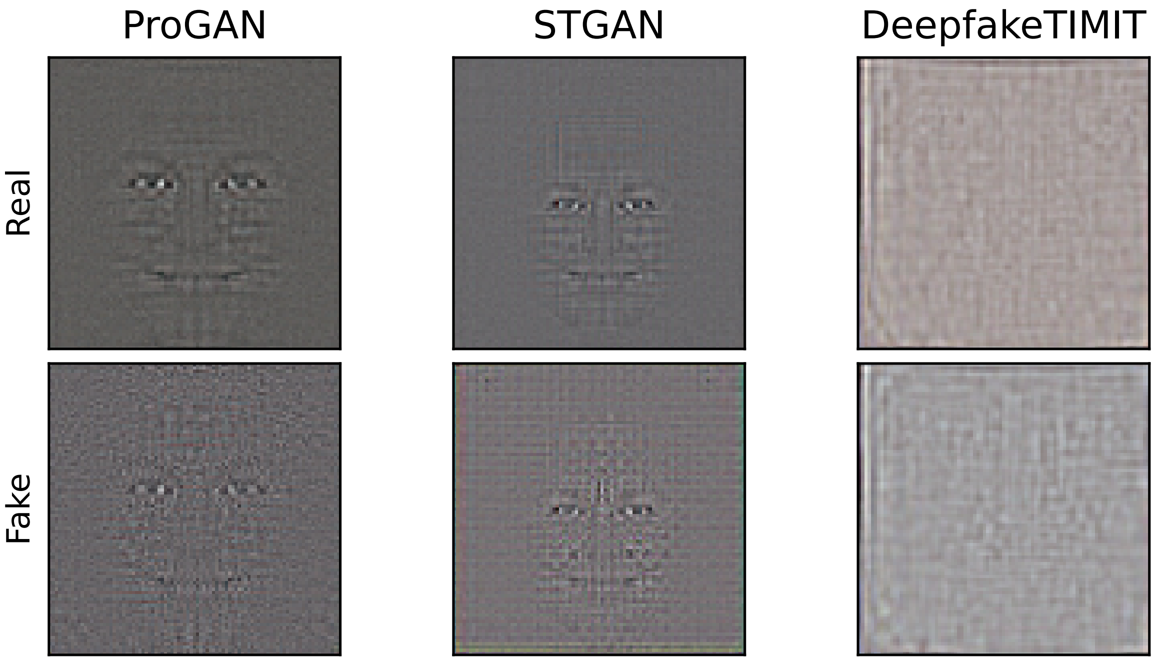


Figure 6.5: The empirical noise fingerprints of different DeepFake types estimated by average noise residual. The average noise residuals of the corresponding real images are also provided for comparison.

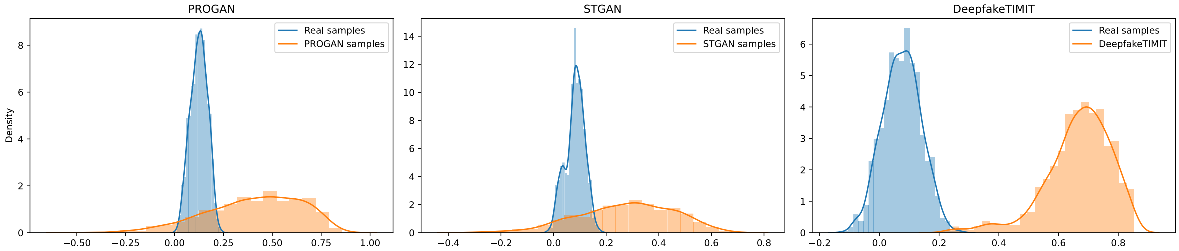


Figure 6.6: The distributions of correlations between individual samples and the noise fingerprints for different DeepFake types.

6.2.3 Discussion

The above model trace analysis identifies three typical model traces throughout the DeepFake pipeline. The interplay of these traces distinguishes DeepFake images from real ones. Hence, a solid and universal trace removal attack should be able to eliminate all these possible traces at once. In this way, the distribution of the modified DeepFake images becomes much closer to that of the real images, enabling the evasion of arbitrary detectors. Since the knowledge of DeepFake traces is derived from the fundamental pipeline shared by different DeepFake types, the trace removal attack to be presented next is applicable to all these DeepFake types.

6.3 TR-Net: trace removal attack

6.3.1 Threat model

6.3.1.1 Victim model

Assume the target victim model is an arbitrary DeepFake detector $\mathscr{C}$, which is a machine learning classifier that distinguishes trace features between real and DeepFake images. $\mathscr{C}$ takes an image I or its hand-crafted features as input and outputs a binary decision of $\{Real, Fake\}$.

6.3.1.2 Victim detector's capability

The attack is designed to defeat an arbitrary DeepFake detector $\mathscr{C}$. Therefore, there are few restrictions on $\mathscr{C}$'s capabilities. The developer of $\mathscr{C}$ can use discretionary model designs and feature engineering, and also sufficient training data. $\mathscr{C}$ is allowed to be trained with fake images from multiple DeepFake generation methods rather than a single one, so as to achieve better cross-task generalization and robustness.

In addition, more robust detectors are considered that have been embedded with defenses. As suggested in [39, 158], training data augmentation with perturbed images can significantly improve a detector's robustness against common attacks. Hence, $\mathscr{C}$ is strengthened during its training phase with two augmentation strategies that confer two different levels of defense.

Weak defense: Empirical augmentation. This method adds perturbed samples from four empirical perturbation models following the settings in [39]:

- Blurring: images are blurred with a Gaussian filter with a kernel size randomly sampled from $(3,5,7,9)$.
- Cropping: images are cropped along both sides with a random percentage sampled from $U(5,20)$ and then resized back to the original resolution.
- Compression: images are compressed with JPEG protocol with a quality factor randomly sampled from $U(10,75)$.
- Noising: i.i.d Gaussian noise is introduced into the images with a Gaussian variance randomly sampled from $U(5.0,20.0)$.

$\mathscr{C}$'s training set was augmented with a combination of these different perturbations in the order of: blurring, cropping, compression, noise. Each strategy was applied with a probability of 50%.

Strong defense: Adversarial augmentation. This method assumed that the developer of $\mathscr{C}$ had full knowledge of the attack model and could use the attack samples directly to augment the training set. This strategy was applied with a probability of 50% as well.

6.3.1.3 Attacker's background knowledge

The proposed attack requires little knowledge of $\mathscr{C}$, i.e., the attacker does not need to know the model architecture, parameters, or the features of $\mathscr{C}$'s interest. As such, there is no need to access the detector, its training set, or the query outputs.

To train the attack model, the attacker is assumed to have an auxiliary dataset containing real and fake images. Although this is a mild assumption given that there are plenty of ready-to-use DeepFake image datasets and models freely available to the public, some additional restrictions are still imposed on the attacker's data availability to simulate the worst-case scenarios:

- Limited dataset size. The attacker has limited resources with which to collect public data and, thus, the resulting dataset size is relatively small.
- Out-of-distribution DeepFake. The attacker can only collect fake images generated by some particular DeepFake methods, which means the auxiliary dataset will not include all types of DeepFakes.

6.3.1.4 Attack goals

A successful attack means that the target detector $\mathscr{C}$ will be misled into classifying the attack samples as '$Real$'. Meanwhile, the attacker may expect the attack to be stealthy with preserving the visual utility of the original DeepFake image. Therefore, the visual difference between the original DeepFake image and the attack sample is required to be small enough that it would not be perceived by humans.

Formally, let $\mathbb{I}^+$ and $\mathbb{I}^-$ be the sets of real images and DeepFake images, respectively. Given a DeepFake image $I^- \in \mathbb{I}^-$, the attack model learns a mapping $\mathscr{A} : I^- \mapsto I^*$. The attacking sample I^* satisfied the following attack goals:

(1) **Fraudulence.** The attack sample successfully deceives an arbitrary detector: $\forall \mathscr{C}, \quad p(\mathscr{C}(I^*) = \mathscr{C}(I^+)) \approx 1$;

(2) **Stealthiness.** The attack sample is perceptually indistinguishable from the original DeepFake image: $\forall I^-, \quad d(I^-, I^*) \le \epsilon$, where $d(\cdot,\cdot)$ is a distance function.

6.3.2 TR-Net

Our trace removal attack is implemented with a trace removal network (TR-Net) based on adversarial learning. As shown in Figure 6.7, TR-Net consists of a generator G and a set of discriminators $\mathbb{D} : \{D_1, D_2, D_3\}$. G takes the original DeepFake images as inputs and reconstructs them to evade trace recognition by the discriminators. Each discriminator in $\mathbb{D}$ is devised for a specific auxiliary trace recognition task. Joint training on $\mathbb{D}$ adversarially impels G to remove different traces concurrently. After the adversarial learning reaches Nash equilibrium, the optimal generator $G^\star$ is adopted as the attack model $\mathscr{A}$, i.e., $\mathscr{A} = G^\star$. Then, given a test DeepFake image I_o^-, the corresponding attack sample is $I_o^\star = A(I_o^-)$.

6.3.2.1 Generator

The generator G is a deep auto-encoder that learns to generate trace-free samples from the original DeepFake samples with an unchanged image size. The backbone of G is a

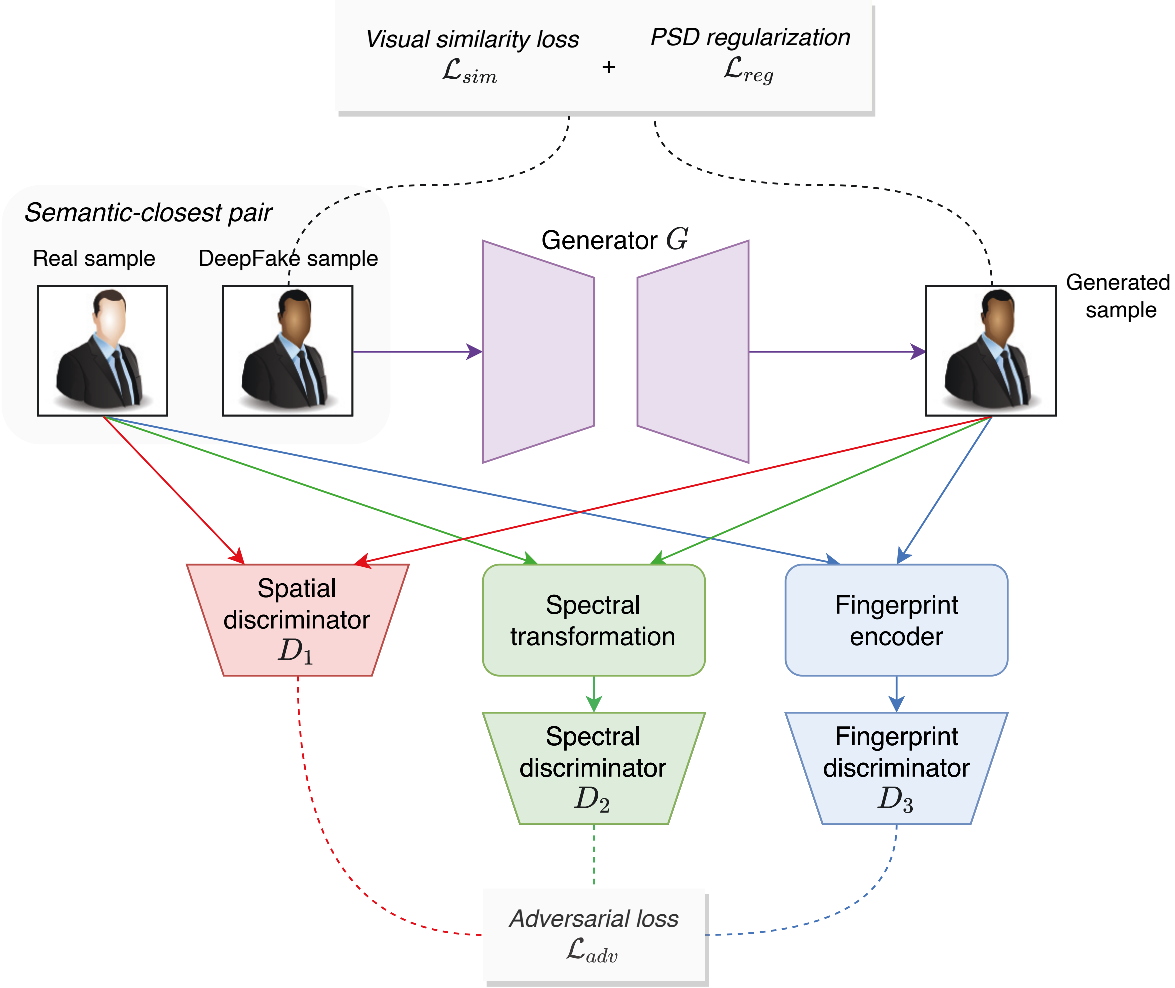


Figure 6.7: Framework overview of the trace removal network.

u-shaped network (U-Net) [120] given its remarkable capacity to reconstruct high-quality images. As shown in Figure 6.8, G consists of an encoder path and a decoder path. The encoder involves repeated convolutional layers (with $3*3$ kernels) and max pooling layers (with $2*2$ kernels), capturing features at different scales of the images while compacting the spatial information. The decoder path is a symmetric expanding counterpart. In each decoding block, the feature map is upsampled to double size while the number of features is halved. Each decoder block also concatenates the output features with the high-resolution features from the corresponding encoder block, such that the feature and spatial information can be preserved for efficient reconstruction.

An additional challenge is that, as discussed in Section 6.2, G is a CNN-based generative model. Thus, it might produce its own model traces, which may interfere with the trace removal process. The loss functions proposed in the subsequent sections effectively suppress this intrinsic noise brought about by G. In addition, we also made two structural improvements as suggested in [32] and [20] to mitigate this problem. First, we replaced the transposed convolution-based upsampling in the original U-Net with bi-linear interpolation-based upsampling. Second, we added a feature scaling layer before the last convolutional layer of G.

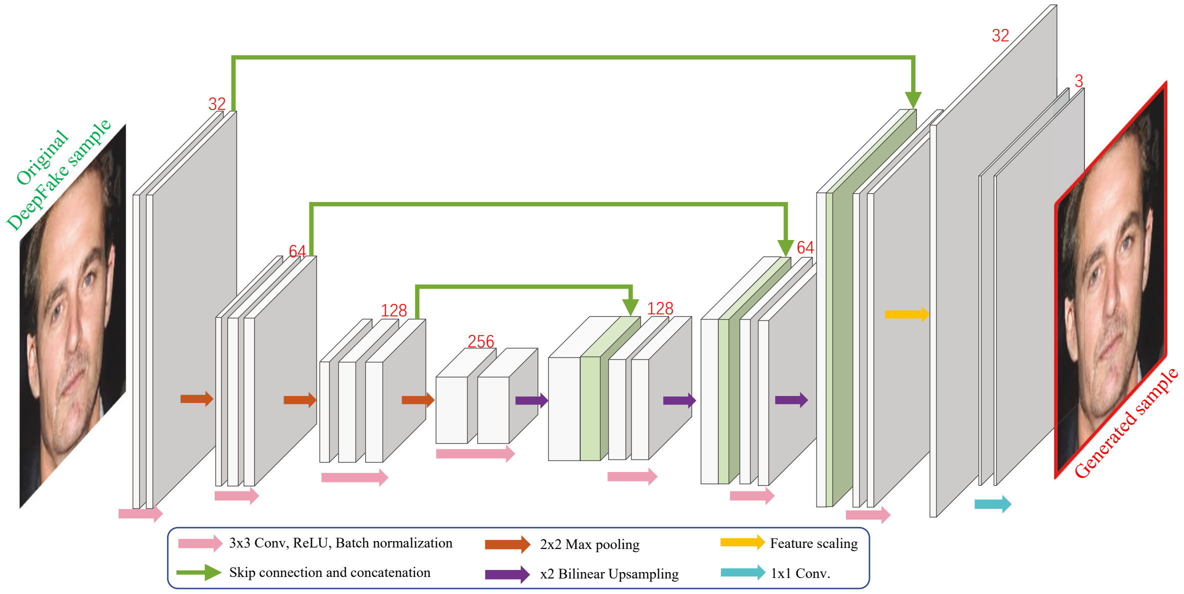


Figure 6.8: The network structure of the generator G.

6.3.2.2 Discriminators

Discriminator D impels G to produce trace-free attack samples via adversarial learning. As a result, D needs to be able to recognize accurate DeepFake trace patterns by learning to classify real and fake images in the trace space. According to our trace discovery, three types of traces are revealed in different domains, each with a unique representation. The inter-domain interference across traces makes a single discriminator learned in one feature subspace impractical to represent all traces accurately. To this end, we propose employing a set of parallel discriminators $\mathbb{D}: \{D_1, D_2, D_3\}$ to disentangle different trace representations. As shown in Figure 6.9, each discriminator is responsible for one particular input trace representation. All the discriminators have the same network structure built on a five-layer CNN. Note that using a complicated structure for the discriminator is unnecessary. In addition to extra computational cost, a complicated discriminator leads to an imbalance between the generator and discriminators during training. A shallow CNN is sufficient to capture these traces accurately in our experiments.

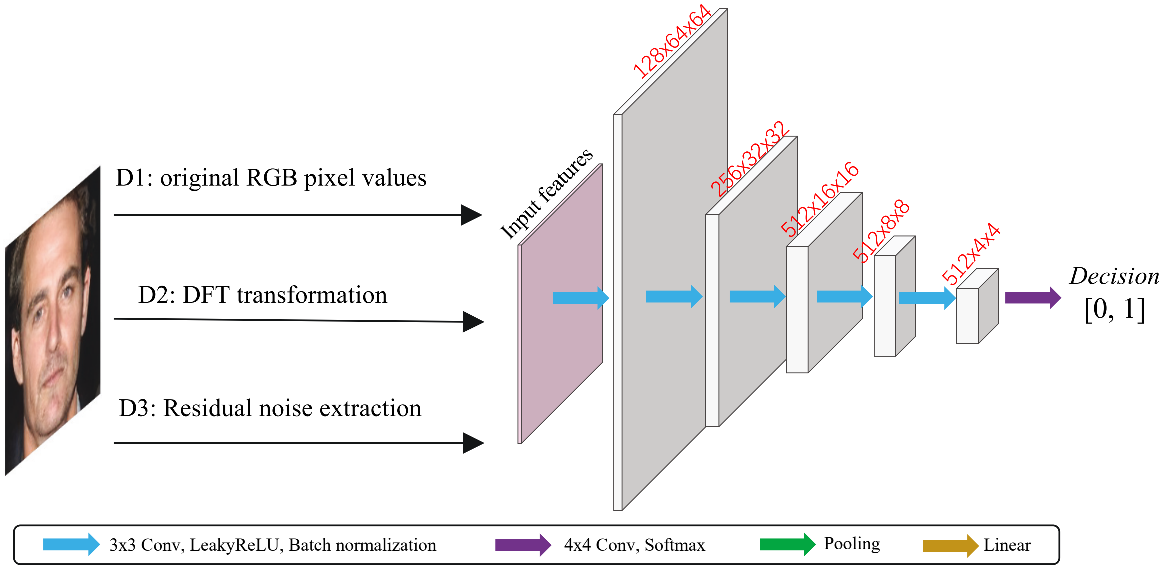


Figure 6.9: The network of the discriminators $\mathbb{D}:\{D_1,D_2,D_3\}$. The three discriminators have the same shallow structure while only differing in the input feature space.

Spatial discriminator D_1 D_1 captures potential spatial anomalies in the spatial domain, including distortions, inconsistencies, disharmony, etc. Similar to the original discriminator in a normal GAN, D_1 is trained directly with the RGB pixel values, and thus can be seen as an incremental refinement on the raw DeepFake images in terms of visual quality.

Spectral discriminator D_2 D_2 learns to recognize the spectral disparities between the real and attack samples. Unlike D_1, D_2 takes the frequency spectrum instead of RGB pixels as its input. The frequency spectrum is transformed from the pixel values by two-dimensional Discrete Fourier Transform (2D-DFT). Given a natural image $I \in \mathbb{R}^{M\times N}$, the 2D-DFT maps each pixel value of the gray-scale component of I to a frequency value $\mathscr{F}(u,v) \in \mathbb{R}^{M\times N}$:

$$\mathscr{F}(I)(u,v)=\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} I(m,n)\cdot e^{-2\pi i\cdot(\frac{um}{M}+\frac{vn}{N})}. \tag{6.3}$$

As the imaginary part is incompatible with a CNN for calculating gradients, directly applying the 2D-DFT $\mathscr{F}(I)$ to D_2 is impractical. Instead, we decompose the complex-valued matrix of $\mathscr{F}(I)$ into its amplitude response $\mathscr{F}_{am}(I)$ and phase response $\mathscr{F}_{ph}(I)$. Let the complex form of $\mathscr{F}(I)$ be $\mathscr{F}(I)=a+bi$, and we have:

$$\begin{aligned}\mathscr{F}_{am}(I) &= |\mathscr{F}(u,v)| = \sqrt{a^2+b^2} \\ \mathscr{F}_{ph}(I) &= \angle\mathscr{F}(u,v) = \arctan\frac{b}{a}.\end{aligned} \tag{6.4}$$

Then, the two components are concatenated as a 2-channel real-valued matrix as the input of D_2, denoted as $\widehat{I} = [\mathscr{F}_{am}(I), \mathscr{F}_{ph}(I)]$.

Fingerprint discriminator D_3 D_3 targets the DeepFake's model fingerprint in the noise space. A reliable fingerprint encoder is required to disentangle accurate fingerprint traces in the input feature space. Existing DeepFake fingerprint encoders include a noise-based method [100] and a learning-based method [158]. We propose to combine the two insights for a more accurate representation. First, a residual noise extraction is performed to represent the noise-level fingerprints. An SRM filter is adopted for this purpose given its effectiveness in estimating local noise distributions for image forensics [41]. The SRM filter has three layers with the following kernels:

$$k_1 = \frac{1}{4}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 2 & 4 & 2 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, k_2 = \frac{1}{12}\begin{bmatrix} -1 & 2 & -2 & 2 & -1 \\ 2 & -6 & 8 & -6 & 2 \\ -2 & 8 & -12 & 8 & 2 \\ 2 & -6 & 8 & -6 & 2 \\ -1 & 2 & -2 & 2 & -1 \end{bmatrix}, k_3 = \frac{1}{2}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The input to D_3 is then denoted as $\widetilde{I} = \mathrm{SRM}(I)$. Then, by training D_3 in the "real v.s. fake" binary classification task, fine-grained fingerprint representations can be obtained from the noise-level fingerprints.

6.3.2.3 Loss functions

We design an adversarial loss to supervise both G and $\mathbb{D}$ of the TR-Net, which can enable trace removal so as to realize the attacking goal of fraudulence. Regarding the goal of stealthiness, a visual similarity loss is imposed on G to ensure that the semantic information of the original DeepFake samples are perfectly preserved in the corresponding attack samples. In addition to achieving these attack goals, one technical challenge is that an ideal trace removal attack requires simultaneously closing the distribution gap between: the attack samples and the real samples at the trace level; and between the attack samples and the DeepFake samples at the semantic level (see Figure 6.10). However, due to the information continuity in an image, the trace features inevitably overlap the semantic features in the latent space, leading to a potential conflict in feature migration directions during optimization. Our loss function design mitigates this nontrivial problem, as shown next.

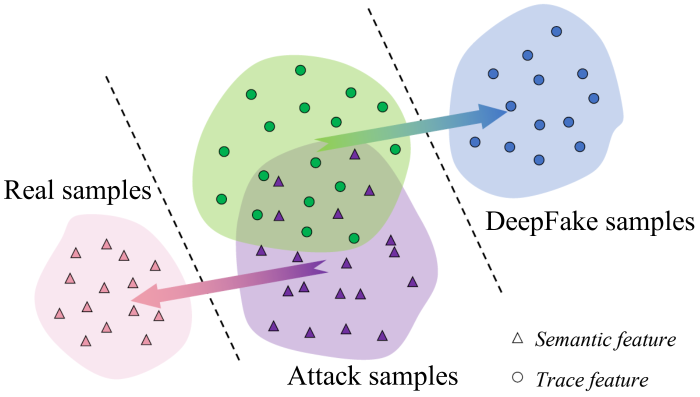


Figure 6.10: The diagram of the changes in the latent feature space of TR-Net during optimization. A conflict in feature migration directions occurs owing to the overlap of trace features and semantic features.

Adversarial loss The adversarial learning of TR-Net is performed with the input data pairs in the form of (I^+, I^-). The discriminators continuously learn to distinguish the generator's output $G(I^-)$ from I^+ in different feature spaces, while the generator tries to mislead the discriminators' judgements about $G(I^-)$. Conventionally, I^+ and I^- are randomly sampled from $\mathbb{I}^+$ and $\mathbb{I}^-$ respectively. However, the random sampling is less practical for TR-Net's optimization considering the conflicts between semantic features and trace features. The visual information between I^+ and I^- should be as consistent as possible to enforce the discriminators to focus on purer trace features while reducing their bias to semantic features. Thus, the *semantically-closest pairs* are constructed to supervise discriminators. If a fake sample is produced by a method where a real source image exists, such as facial attribute editing or face replacement, the source image is applied straightforwardly as the semantically-closest counterpart. For a face synthesis sample created out of nowhere, its nearest neighbor is retrieved from the real image set $\mathbb{I}^+$ as a counterpart.

With the semantically-closest pair (I^+, I^-) in hand, the adversarial loss for jointly training the "one-versus-multiple" framework is denoted as:

$$\mathscr{L}_{adv}(G, D_1, D_2, D_3) = \lambda_1 \mathscr{L}(G, D_1) + \lambda_2 \mathscr{L}(G, D_2) + \lambda_3 \mathscr{L}(G, D_3), \tag{6.5}$$

where

$$\mathcal{L}(G,D_i) = \mathbb{E}_{x^+,x^-}[log(D_i(x^+)) + log(1 - D_i(G(x^-)))], \tag{6.6}$$

The input x varies for different discriminators, i.e., $x = I$ for D_1, $x = \widehat{I}$ for D_2 and $x = \widetilde{I}$ for D_3. λ_1, λ_2, λ_3 are weights to balance the contribution of three discriminators, subject to $\lambda_1 + \lambda_2 + \lambda_3 = 1$.

Visual similarity loss To satisfy the stealthiness goal, a visual similarity loss is additionally imposed on the generator G. The commonly-used pixel-wise distance $||I^- - G(I^-)||_2$ is not particularly applicable to our method as it will typically lead to overfitting the visual information. In turn, this will exacerbate the conflict between the semantic features and the trace features, thus compromising the trace removal. Moreover, despite having D_2 to encourage spectra matching from the attack samples to the real images, we experimentally find that only a D_2 is insufficient to well match high-frequency components. This is because in natural images, information tends to be centralized in lower-frequency components.

Instead, we propose a novel visual similarity loss plus a power spectral density (PSD) regularization to cope with the above problem. Given an image I, first, a filter is applied to its center-shifted DFT spectrum. This decomposes I into its low frequency components I_l and high frequency components I_h:

$$\begin{cases} I_l = \mathcal{F}^{-1}(\mathcal{H}(u,v) \cdot \mathcal{F}(u,v)) \\ I_h = \mathcal{F}^{-1}(1 - \mathcal{H}(u,v) \cdot \mathcal{F}(u,v)) \end{cases}, \tag{6.7}$$

where $\mathcal{F}^{-1}$ is the reverse DFT, $\mathcal{H}(u,v) = exp(-\frac{u^2+v^2}{2\sigma^2})$ is a Gaussian filter. Then the visual similarity loss between a source fake image I^- and its reconstructed version $G(I^-)$ is computed as the VGG perceptual loss [61] on the low frequency components:

$$\mathcal{L}_{sim}(G) = \frac{1}{W * H} \left\| \mathrm{VGG}_k\left(I_l^-\right) - \mathrm{VGG}_k\left(G(I^-)_l\right) \right\|_2^2, \tag{6.8}$$

where W and H are the dimensions of the respective feature maps within the VGG network [128] and VGG_k denotes the features extracted at VGG's k-th layer.

Additionally, a PSD regularization is added to the visual similarity loss to enforce the mapping of frequency information between the attack samples and the real images. The PSD of an image I can be represented as a one-dimensional profile of the center-shifted power spectrum resulting from an azimuthal integration over each radial frequency θ:

$$\begin{aligned}\text{PSD}(\omega_k) &= \int_0^{2\pi} \|\mathscr{F}(I)(\omega_k \cdot \cos(\theta), \omega_k \cdot \sin(\theta))\|^2 \ \mathrm{d}\theta \\ \text{for} \quad & k = 0, \dots, M/2-1.\end{aligned} \tag{6.9}$$

Benefiting from the semantically-closest pair (I^+, I^-) where the lower frequency components are close to each other, the PSD regularization can operate on the high frequency components merely, which is computed as the Euclidean distance between the PSDs of I_h^+ and $G(I^-)_h$:

$$\mathscr{L}_{reg}(G) = \frac{1}{M/2-1} \left\|\text{PSD}(I_h^+) - \text{PSD}(G(I^-)_h)\right\|_2^2 \tag{6.10}$$

The final training objective of the TR-Net is:

$$T = arg \min_{\{G,D_3\}} \max_{\{D_1,D_2,D_3\}} \left\{\mathscr{L}_{adv} + \mathscr{L}_{sim} + \mathscr{L}_{reg}\right\} \tag{6.11}$$

6.3.3 Comparison with previous attacks

To date, the published adversarial attacks have had some limitations. First, searching for the optimal adversarial noise perturbations to a target detector typically requires a certain level of information about the detector itself, such as the parameters, network structure, or the outputs. Thus, there will be a transferability issue, i.e., the attacks crafted based on a specific target detector cannot work when facing another unknown or black-box detector [4, 160]. Second, the feasibility of an adversarial attack on some advanced detectors which involve sophisticated network designs will be problematic. Under these circumstances, it becomes difficult, if not impossible, to search for the optimal perturbations that will maintain a high attack success rate while being largely imperceptible.

Regarding the reconstruction-based attacks, our method is analogous to this genre of attacks, but fundamental differences exist. Similar to adversarial attacks, reconstruction-based attacks are performed in a "detector-specific" way. The attacker is assumed to know what type of forgery features are of prime interest to the target detector. What is worse, these attacks solely focused on an individual feature type in a single signal domain, irrespective of the fact that various traces exist in different domains.

By contrast, our method improves on anti-forensic attacks by removing multiple forgery traces at the same time. Additionally, they are removed in a way that is agnostic to the detector. The result is better transferablity to an unknown detector. Technically, this is more challenging than dealing with a single trace feature given the interplay between traces and the inter-domain interference, yet the proposed TR-Net is competent to meet the challenge.

6.4 Experimental Evaluations

We evaluate the proposed trace removal attack in heterogeneous security scenarios where the attacker has different background data knowledge, and the detectors' defensive capability varies. In each scenario, the attack effectiveness is assessed by verifying whether the goals of fraudulence and stealthiness have been satisfied. We also provide a closer look into the trace removal result from different dimensions to further justify its success.

6.4.1 Evaluation metrics

(1) The fraudulence goal is verified in terms of detection accuracy, calculated as the proportion of correctly classified samples out of all the samples in a single class. Attack samples with higher fraudulence result in lower detection accuracy of the test detector. (2) The stealthiness goal is verified by assessing the visual quality loss in attack samples. We used peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) between an original DeepFake image and its corresponding attack sample to evaluate the visual quality loss. PSNR quantifies the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. SSIM is a common metric for measuring the similarity between two images. A larger value in either PSNR or SSIM indicates a smaller loss in visual quality, which equates to better stealthiness of the attack sample.

6.4.2 Datasets

The proposed trace removal attack is applicable to all DeepFake types including face synthesis, facial attribute editing, face replacement. To the best of our knowledge, existing publicly-available DeepFake detection datasets fail to cover all these methods. Thus, we create a new DeepFake dataset called *All-in-One-DF* for a thorough evaluation, which consists of $66,000$ semantically-closest pairs of real and fake images (i.e., $132,000$ images in total) from four sources.

(1) CelebA: A large-scale dataset containing more than $200k$ real face images. The images are cropped and aligned to the size of $128 * 128 * 3$ with the face in the centre.

(2) Face synthesis: We employ ProGAN, one of the most popular unconditional GANs to synthesize non-existing face images. We utilize the ProGAN instance pre-trained with CelebA [158] to generate $22,000$ fake images. Then we retrieve their corresponding 1-nearest-neighbor similar counterparts from the CelebA dataset to construct semantically-closest pairs.

(3) Facial attribute editing: We select STGAN, a state-of-the-art facial attribute editing GAN for this use. We randomly sample $22,000$ real images from the remaining CelebA dataset and apply the official STGAN instance [92] to modify either the soft-biometric attribute (facial age) or the appearance attribute (hair colour), resulting in $22,000$ fake samples.

(4) Face replacement: DeepfakeTIMIT [74] is a human video dataset where faces are swapped and rendered using GAN-based approaches. There are 320 pairs of source videos and their face-swapped counterparts in DeepfakeTIMIT. We randomly select $22,000$ frames from all videos

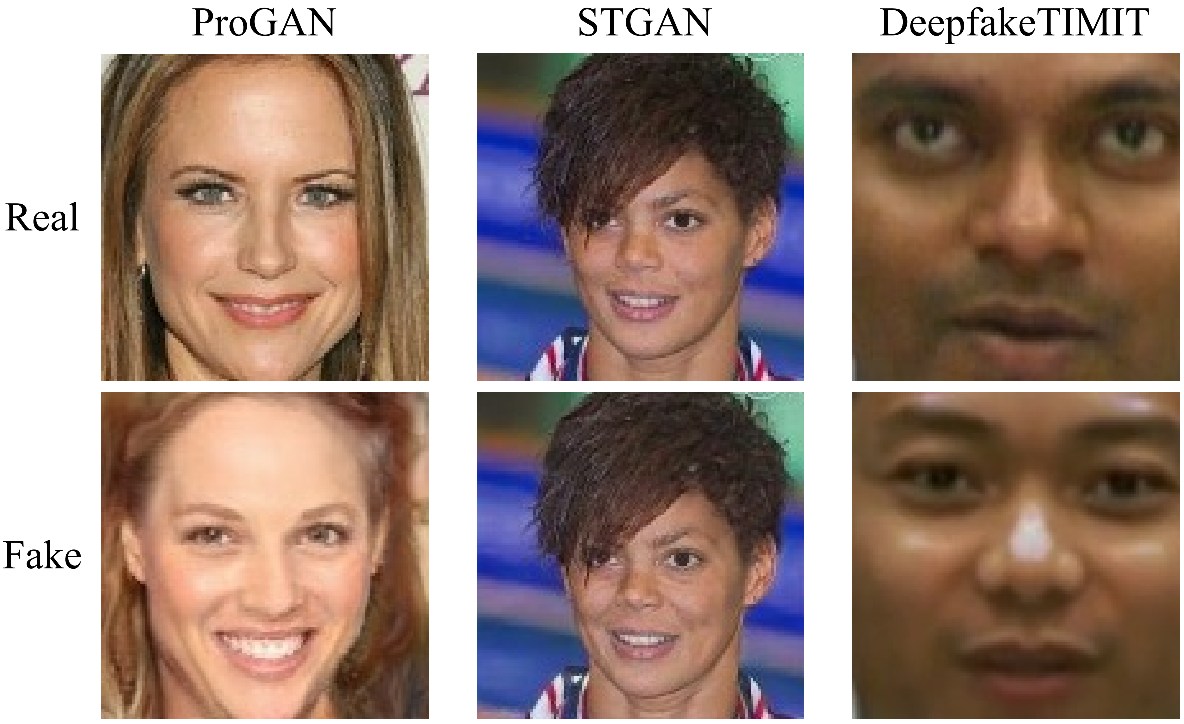


Figure 6.11: Examples of semantically-closest pairs from three different DeepFake types.

6.4.3 Selected Victim Detectors

To show the efficacy of the proposed attack model in attacking arbitrary detectors, we select six recently proposed representative DeepFake detectors, which evenly cover the three detector categories outlined in Section 2.2.2.

Spatial-based detectors: **Xception [122]** is a deep CNN widely adopted as the backbone network in face forgery forensics tasks. It has achieved leading performance in some benchmark datasets by learning directly from RGB pixel inputs. **Patch-CNN [19]** focuses on the local properties in semantic regions rather than on global semantics. It aggregates the decisions of a set of truncated Xceptions learned from image patches for the final binary decision.

Frequency-based detectors: **DCTA [39]** is a shallow CNN classifier learned from the 2D-DCT spectra of images. **F^3-Net [116]** is one of the state-of-the-arts in DeepFake detection. It involves a two-stream collaborative network that combines frequency-aware decomposition and local frequency statistics to learn frequency-aware clues.

Fingerprint-based detectors: **LF [158]** is a deep CNN that learns GAN fingerprints in a multi-source identification task. The original multi-classification results are further divided into the "real-or-fake" binary decisions. **NF [100]** is a non-trainable method that differentiates GAN images from real ones via a cross correlation score of the noise residual-based fingerprints.

In addition, we also use ensemble learning to fuse the three categories of detectors into a stronger one, denoted as **Ensemble**. **Xception**, **DCTA** and **LF** are selected as the base detectors and a random forest classifier is trained based on the features output by the final pooling layers of the base detectors.

6.4.4 Settings

The *All-in-One-DF* dataset is randomly partitioned into a training set with $60,000$ semantically-closest pairs and an evaluation set with $6,000$ pairs. For all detectors, we follow the training settings recommended in the original papers. The detectors are trained on the training set with a $9:1$ training-validation ratio. Regarding the training of TR-Net, we set the batch size to 150. Both the generator and discriminators are optimized using the RMSprop optimizer [123] with initial learning rates of $1.6e-3$ and $1.6e-4$, respectively, plus a scheduler with a decay rate of 0.5. The scheduler is executed at the end of a training epoch if the loss stopped decreasing. There are 8 training epochs in total. The weight set $\{\lambda_1, \lambda_2, \lambda_3\}$ is set to $\{0.2, 0.6, 0.2\}$. The weight decision process is detailed in Section 6.5. After training, the checkpoint with the minimal generator loss in the last epoch is nominated as the attack model and applied to the $6,000$ fake images in the evaluation set to craft attack samples.

6.4.5 Attacking with unlimited background knowledge

We first evaluate the attack performance in the scenario where the attacker has no limits on the background knowledge of data, i.e., the whole training set is available for training the attack model. For a comprehensive evaluation, the proposed trace removal attack is compared with several baseline attack methods. Also the detectors with varying abilities are considered, i.e., detectors without defense, with weak defense, and with strong defense.

6.4.5.1 Baseline attacking methods

We select four other attack methods to demonstrate baseline performance, including adding random noise (**Noise**), two classic adversarial attacks **FGSM** [77] and **PGD** [98], and a reconstruction-based attack **GANprintR** [111]. The **Noise** operation is the same as the "noising" perturbation described in Section 6.3.1.2. Both the **FGSM** and **PGD** attacks are optimized based on the **Xception** detector and then apply to all detectors so as to assess the white-box and black-box attack capacities simultaneously. The maximum perturbation ϵ is set as 0.003 for both attacks. For **GANprintR**, we follow the setting in the original paper.

6.4.5.2 Evaluating fraudulence

Attacking detectors without defense Table 6.1 details the attack results against detectors without defense. The first column shows the detection accuracy on the original clean fake samples. All detectors achieve high accuracy over 90.00%, except for the non-trainable detector **NF**. After attack, the accuracy of all detectors decreases, indicating that the state-of-the-art detectors are still vulnerable to attacks. The frequency-based detectors, especially **F^3-Net**, are relatively more robust than other individual detectors. The reason may be that all attacks lead to more significant changes in the frequency domain than in the pixel domain. In addition, the ensemble detector targeting all traces, unsurprisingly, outperforms any individual detector targeting a single trace in terms of robustness.

Regarding the attacks, the two adversarial attack methods, **FGSM** and **PGD**, are particularly destructive to **Xception**. This is not surprising because these two attacks are optimized based on **Xception** in a white-box manner. They also showed good transferability on **Patch-CNN** which has similar structural blocks to **Xception**. However, as earlier discussed, this detector-specific design leads to poor transferability on other unknown types of detectors. By comparison, **TR-Net** takes advantage of the detector-agnostic design, achieving competitive or superior results in attacking all six detectors. After the trace removal attack, the classification accuracy of all detectors has decreased markedly, and the average accuracy of the six has dropped from 92.16% to 22.85%. The results indicate the proposed trace removal attack is universal and well transferable across different detectors.

Table 6.1: Performances of five attack methods against seven detectors without defense. The bold value indicates the best attack result in each row.

Accuracy(%)	**Clean**	**Noise**	**FGSM**	**PGD**	**GANprintR**	**TR-Net**
Xception	99.86	65.44	4.43	**0.01**	58.53	17.90
Patch-CNN	92.13	53.91	12.36	**9.81**	57.31	13.06
DCTA	90.66	51.59	33.18	25.37	70.24	**20.21**
F^3-Net	99.97	85.41	49.62	45.73	80.73	**31.10**
LF	91.55	37.12	16.00	15.55	64.76	**14.75**
NF	71.12	42.65	28.21	25.70	31.88	**22.74**
Ensemble	99.80	81.23	50.11	47.20	83.33	**40.21**
Average	92.16	59.62	27.70	24.20	63.83	**22.85**

Attacking detectors with defenses Next, we test the attacks in the cases that the detectors are embedded with varying defenses as described in Section 6.3.1.2. Detectors with the weak defense, i.e., the empirical augmentation strategy, are denoted as {**model name**}**(+)** and those with the strong adversarial augmentation defense are denoted as {**model name**}**(++)**.

Weak defense. Table 6.2 shows the classification accuracy of detectors embedded with the empirical data augmentation strategy. The high accuracy values in the first column indicate that all detectors still maintain stable detection capability on clean samples after defense.

We can see that after being strengthened with empirical data augmentation, the robustness of all detectors were improved against all attack methods. Regarding the attack methods, the **Noise** attack is barely misled the detectors, and the attacking specificity of **FGSM** and **PGD** on **Xception(+)** and **Patch-CNN(+)** was no longer significant. Comparatively, the **TR-Net** maintained satisfactory, degrading the classification accuracy of almost all detectors to lower than random guess, except for **F^3-Net(+)**. Moreover, **TR-Net** surpassed all baseline attack methods in five out of the six detector groups.

Table 6.2: Performances of five attack methods against seven detectors with weak defense. The bold value indicates the best attack result in each row.

Accuracy(%)	**Clean**	**Noise**	**FGSM**	**PGD**	**GANprintR**	**TR-Net**
Xception(+)	98.86	98.00	34.73	33.35	73.37	**30.13**
Patch-CNN(+)	90.66	90.01	58.47	42.19	79.65	**33.21**
DCTA(+)	94.99	95.77	46.94	45.44	84.80	**40.56**
F^3-Net(+)	99.64	96.87	**55.79**	56.16	91.75	61.03
LF(+)	94.63	82.70	48.68	39.91	76.54	**38.60**
NF(+)	75.54	74.99	59.71	56.62	58.70	**40.21**
Ensemble(+)	99.97	99.01	66.21	64.32	92.01	**59.03**
Average	93.47	91.05	52.93	48.28	79.55	**43.25**

Strong defense. Table 6.3 shows the classification accuracy of the detectors embedded with the adversarial data augmentation strategy. Note that the adversarial augmentation strategy is specific to each attack method, thus, the results on clean (*cle*) / attacked (*att*) DeepFake samples are reported individually for each attack method.

From the table, we can see that the adversarial data augmentation strategy substantially improved the robustness of all detectors against the four baseline attack methods. Take **PGD**, the best baseline attack method in our experiments as an example, the average accuracy of strongly defended detectors only degrades from 92.19% to 73.54%, whereas the corresponding result for the weakly-defended detector is 93.47% down to 48.28%, and 92.16% down to 24.20% for the naked detector. However, the strong defense only makes a relatively small impact on our attack method. The average detection accuracy on the **TR-Net** samples is 59.74%, much lower than the accuracy on other attack samples, at merely a little higher than a random guess.

We also observe an intriguing phenomenon that unlike other augmentations, adversarial augmentation with the **TR-Net** samples can significantly compromise the detectors' ability to classify clean fake samples. The reason may be that the samples after trace removal are inherently closer to the real samples, which can confuse the detector during training. This reveals the potential to use trace removal attack to poison a DeepFake detection dataset, which remains future investigation.

Table 6.3: Performances of five attack methods against seven detectors with strong defense. The detection accuracy on clean (cle) and attacked (att) samples are shown individually for each attack. The bold value indicates the best attack result in each row.

Accuracy(%)	**Noise**		**FGSM**		**PGD**		**GANprintR**		**TR-Net**	
	cle	*att*	*cle*	*att*	*cle*	*att*	*cle*	*att*	*cle*	*att*
Xception(++)	99.37	99.64	97.69	71.22	97.36	59.32	98.34	98.37	60.17	**46.71**
Patch-CNN(++)	94.83	94.36	93.51	82.29	92.96	76.26	96.25	95.86	55.53	**43.17**
DCTA(++)	93.08	91.20	93.53	77.87	92.88	78.91	94.65	96.32	77.45	**70.12**
F^3-Net(++)	99.51	99.88	96.36	84.18	95.50	81.85	99.54	98.53	88.44	**78.01**
LF(++)	94.09	88.61	93.82	69.45	90.61	66.20	97.78	95.58	53.44	**44.86**
NF(++)	73.29	71.67	73.51	67.41	78.00	67.11	79.64	68.89	65.98	**60.76**
Ensemble(++)	99.63	99.74	98.21	90.12	98.00	85.13	99.66	99.01	88.12	**73.90**
Average	93.40	92.16	92.38	77.51	92.19	73.54	95.12	93.22	69.88	**59.65**

Discussion Figure 6.12 offers an intuitive comparison of five attack methods in three defense strategy groups. The trace removal attack is the most effective in all groups. In the circumstances where detectors are defended with data augmentation strategies, especially the adversarial augmentation strategy, the baseline attack methods generally undergo a considerable loss of efficacy, while TR-Net continues to pose a threat, and the threat is even more serious than the white-box adversarial attacks. In addition, as shown in Tables 6.1-6.3, The trace removal attack shows superior transferablity across different detectors compared to the baselines. Again, we emphasize that, unlike the baseline attacks, our attack was implemented upon all detectors being completely unknown during training. *In conclusion, the attacking goal of fraudulence is well satisfied by the proposed trace removal attack.*

6.4.5.3 Evaluating stealthiness

As mentioned, this goal was evaluated based on visual quality. To be considered stealthy, a given attack sample was required to be perceptually indistinguishable from the corresponding DeepFake image.

Table 6.4 demonstrates the visual quality differences between DeepFake samples and attack samples in the evaluation set in terms of the average PSNR and SSIM scores. **TR-Net** achieved the highest PSNR ($35.16 \pm 3.32db$) and SSIM (0.988 ± 0.004) scores, indicating that attack samples generated by **TR-Net** contain less noise and have a visual quality closer to the original DeepFake samples. Figure 6.13 also provides a

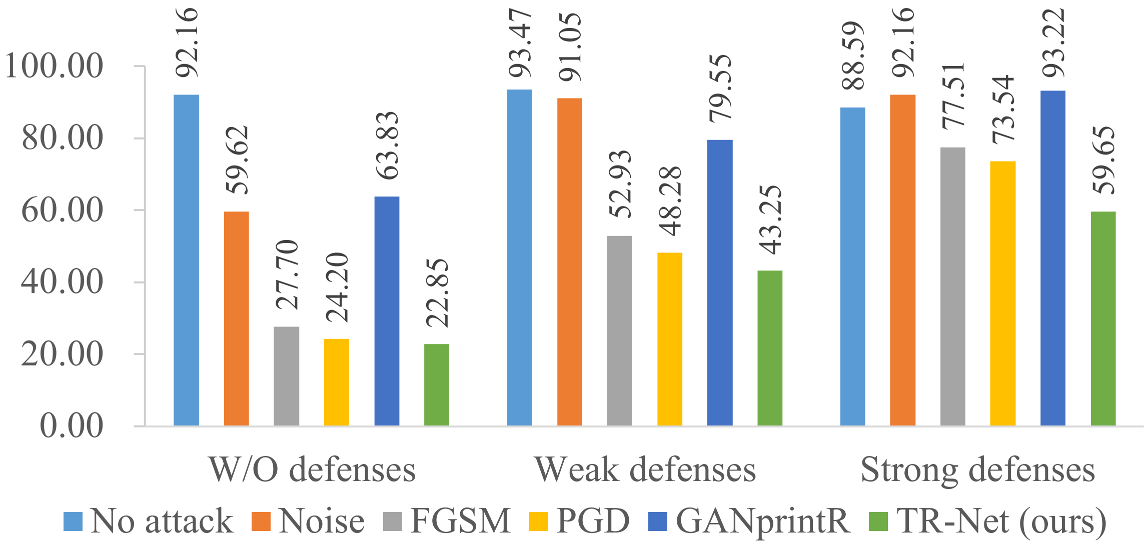


Figure 6.12: The average detection accuracy under five attack methods in three defensive strategy groups.

qualitative view of the examples of three types of DeepFake images and the corresponding attack samples from different attack methods. As shown in the figure, the methods that add noise, including including **Noise**, **FGSM** and **PGD**, bring perceptual noise or a blurriness to the attack samples that may have be potentially screened out by the forensic investigators. By contrast, the reconstruction-based methods, especially **TR-Net**, generated high-fidelity attack samples that were perceptually similar to the original ones. *Thus, we can conclude that the goal of stealthiness is well satisfied as well.*

Table 6.4: The visual quality comparison of different attack samples in terms of the average PSNR and SSIM scores. The bold value indicates the best result in each row.

	Noise	**FGSM**	**PGD**	**GANprintR**	**TR-Net** (ours)
PSNR (db)	26.86±3.24	30.13±0.08	32.23±0.20	25.80±2.58	**35.16**±3.32
SSIM	0.634±0.149	0.764±0.059	0.836±0.041	0.924±0.050	**0.988**±0.004

6.4.6 Attacking with limited background knowledge

The next evaluation scenario imposes restrictions on the attacker's background knowledge of data. Here, only the **Xception** and **F^3-Net** detectors are tested considering their generally better detection ability than other detectors.

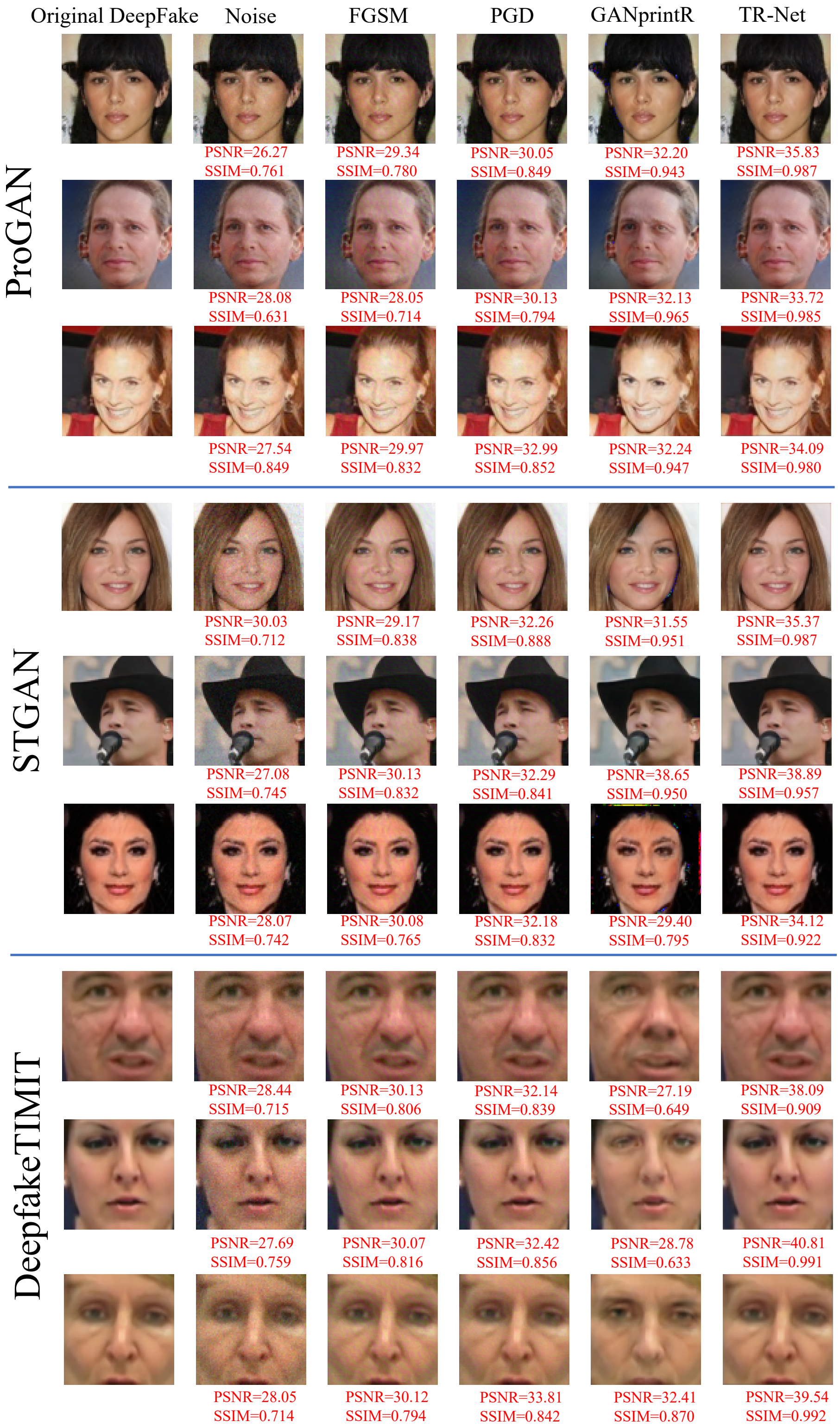


Figure 6.13: Examples of three types of DeepFake images and the corresponding attack samples from different attack methods.

6.4.6.1 Limited dataset size

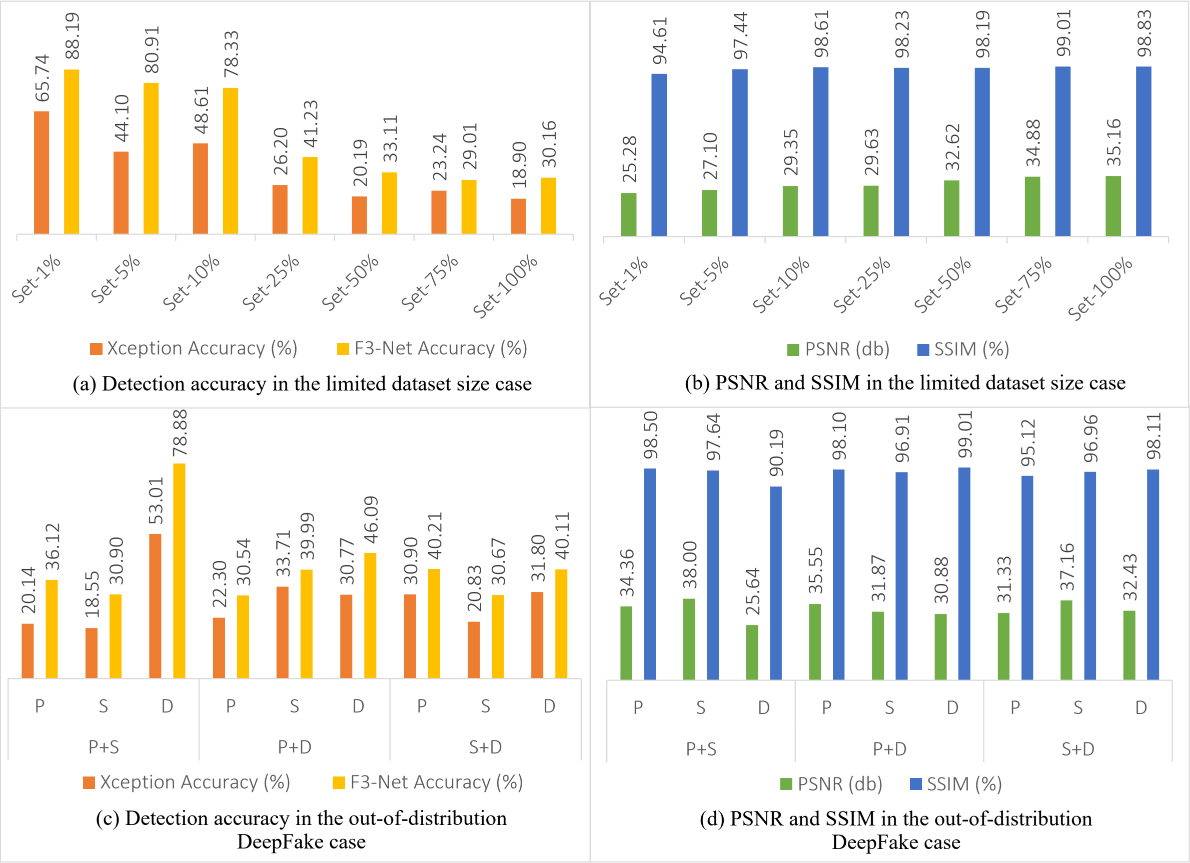
To simulate that the attacker only has access to a limited dataset, we randomly sampled six subsets from the full training set, containing 1%, 5%, 10%, 25%, 50%, 75% and 100% of data (i.e., for 1% that equates to a total of 660 semantically-closest pairs of real and fake images). Then we trained TR-Net from scratch on each subset individually and evaluated our results on the same evaluation dataset as in the previous scenario.

Figure 6.14.a and 6.14.b illustrates the detection accuracy and PSNR and SSIM scores for each subset. From the results, it appears there is a threshold for the dataset size that is within $10\% - 25\%$, under which both the accuracy and visual quality are affected. This is unsurprising since attack methods based on GAN learning are essentially data-driven. However, when the training set size equates to more than a quarter of the original data set, all metrics increase rapidly and remain relatively stable at a satisfactory level. The results indicate that TR-Net fits well even with a relatively small amount of training samples which are easily collected. This weak data volume-dependency makes TR-Net practically feasible.

6.4.6.2 Out-of-distribution DeepFake

We also assessed TR-Net's performance on out-of-distribution DeepFakes to demonstrate its domain independence. In this scenario, the attacker was restricted to train the model with only two types of DeepFake images. Yet the evaluation set still contained all three DeepFake types. Here, for example, "P+S" (short for "ProGAN+STGAN") indicates training with ProGAN and STGAN samples.

Figure 6.14.c and 6.14.d show the results for detection accuracy and visual quality when trained with different training groups. What is shown is that, when implementing the attack on the samples generated by an unknown DeepFake method that is not included in the training set, the resulting attack samples suffer from a decrease in both detection-evasive ability and visual quality. For instance, comparing the ProGAN results in the "P+S" group (where ProGAN is included in the training set) with those in the "S+D" group (where ProGAN is not included in the training set), the **Xception**'s detection accuracy increases from 36.12% to 40.21% and the **F^3-Net**'s detection accuracy increases from 20.14% to 30.90% (Figure 6.14.c). Also, the PSNR scores decreased from 34.36db to 31.33db and the SSIM scores decreased from 98.50% to 95.12% (Figure 6.14.d). The performance degradation was much more significant for unknown DeepfakeTIMIT samples than that for unknown ProGAN and STGAN samples. The reason is that both



(a) Detection accuracy in the limited dataset size case

(b) PSNR and SSIM in the limited dataset size case

(c) Detection accuracy in the out-of-distribution DeepFake case

(d) PSNR and SSIM in the out-of-distribution DeepFake case

Figure 6.14: Attacking performance in the settings where the attacker is imposed with different restrictions on data accessibility. P: ProGAN; S: STGAN; D: DeepfakeTIMIT

the source ProGAN and STGAN models were pre-trained with the CelebA dataset, while the source DeepfakeTIMIT model is developed with another dataset where a domain inconsistency exists. Our findings suggest that fine-tuning TR-Net in a domain to be consistent with the target detector helps to improve the efficacy of the attack.

6.4.7 A closer look into trace removal

In this section, we offer a closer look at the trace removal to justify the DeepFake trace discovery outlined in Section 6.2. This examination helps us to understand why and how TR-Net removes all traces.

6.4.7.1 Explanation in the feature space

The representations of different DeepFake traces are well learned by a set of discriminators, thus, we analyzed the geometrical shifting of trace features encoded by each discriminator in the latent feature space. Since that the generator and discriminators are trained in parallel, the discriminators $\mathbb{D}^*$ resulting from the same checkpoints of the optimal generator G^* are adopted as the trace feature descriptors. The trace features output by the last $512*4*4$ convolutional layer of $\mathbb{D}^*$ are analyzed. t-SNE [141] is performed to reduce feature dimensionality, so as to obtain an interpretable two-dimensional view of geometrical shifting.

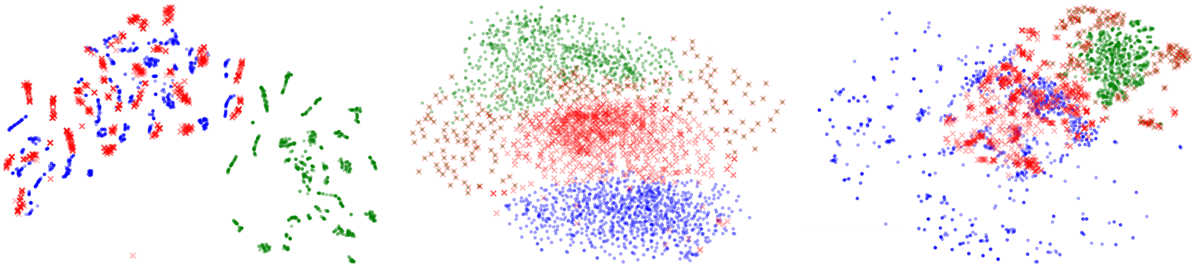


Figure 6.15: Trace features in the latent spaces learned by, from left to right, the spatial discriminator D_1, the spectral discriminator D_2 and the fingerprint discriminator D_3. t-SNE is used to project the representations of features from each discriminator's last convolutional layer onto its two principal components. • indicates real sample; • indicates fake sample; × indicates attack sample;

The result from each discriminator is shown individually in Figure 6.15. Each discriminator corresponds to a single trace type. We can see that there is a distinct trend that the attack samples' trace features are transferring towards the the real images' features. The result confirms our conjecture that TR-Net can reduce the distribution gap between the attack samples and the real samples at the trace feature level via adversarial learning. In addition, the spectral traces from D_1 and the fingerprint traces from D_3 have a more significant migration than the spectral traces from D_2. This occurs because of the aforementioned optimization conflict between the semantic features and the trace features in latent space. Since the frequency components are closely correlated with both the trace and semantic information where no distinct boundary applies, weakening the trace representations of the DeepFake samples while retaining their visual information must lead to a sub-optimum. Even so, the attack efficacy is barely affected as shown in previous experiments.

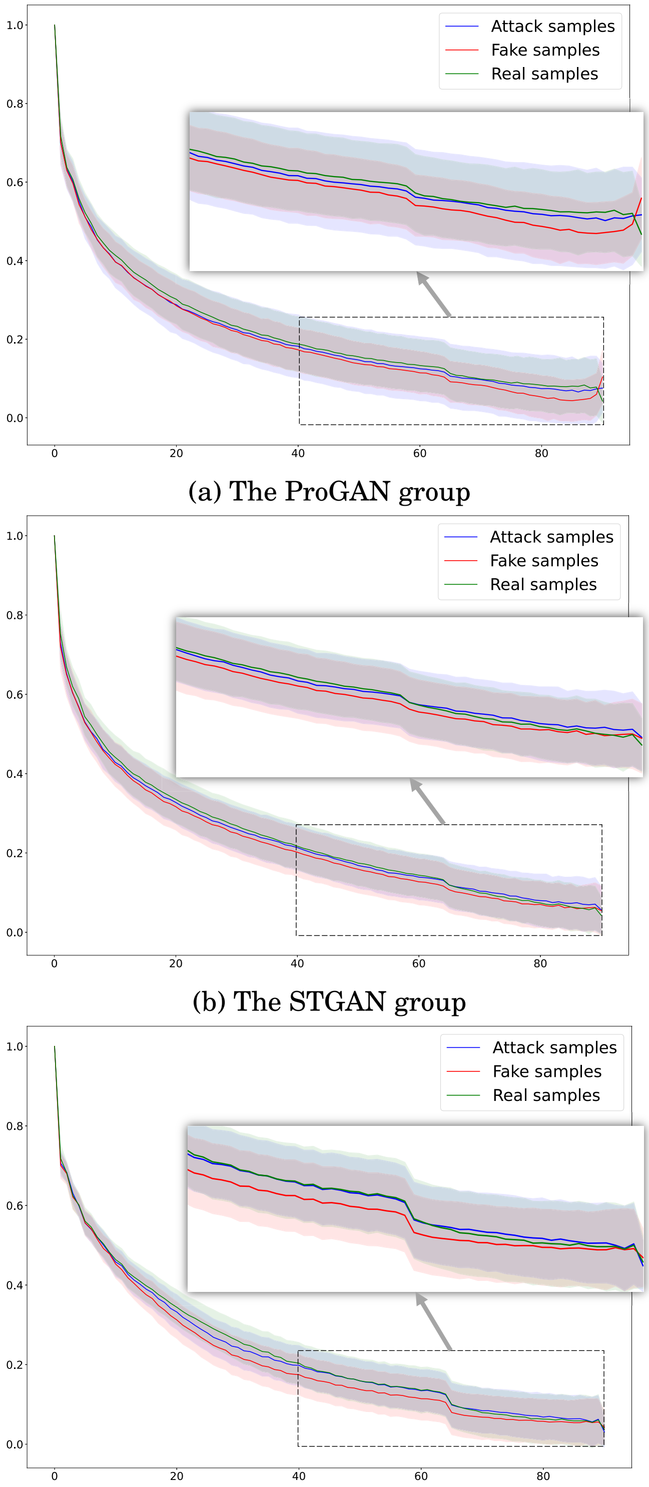
6.4.7.2 Explanation in the frequency and noise spaces

Then, we explain the trace removal in the frequency and noise space to further justify the trace removal success. First, we compare the PSD distributions of real, DeepFake and attack samples. Figure 6.16 shows the average PSD distribution along with standard deviation for different DeepFake types. The results show that for all DeepFake types, the distribution gaps between attack and real samples are significantly smaller than those between attack and fake samples. The result again showcases the wide applicability of the trace removal attack to various DeepFake types. Meanwhile, the gaps between attack and real samples are slighter in the STGAN and DeepfakeTIMIT groups than in the ProGAN group. We suppose the reason is associated with the semantically-closest pairs. For the semantically-closest pairs in both the STGAN and DeepfakeTIMIT groups, each fake sample has an exact source real image as a counterpart. In contrast, the fake samples in the ProGAN's semantically-closest pairs correspond to their nearest-neighbor similar real images where a larger visual difference exists, leading to under-fitting in the frequency domain.

We also provide some qualitative results as complementary evidence. Figure 6.17 illustrates the average spectra and noise residual differences between the real and source fake samples and between the real and attack samples. A brighter entry indicates a larger difference. As shown in the figure, the differences between the real and source fake samples are much more significant than those between the real samples and the attack samples. This result further highlights the fact that successful trace removal will refine the DeepFake images to be closer to the real ones, which they can deceive arbitrary detectors.

6.4.7.3 The effect of individual traces

We further investigate the effect of removing each individual trace instead of all. For this purpose, we evaluate three partial versions of **TR-Net** where only an independent discriminator is considered for each, namely **TR-Net-**D_1, **TR-Net-**D_2 and **TR-Net-**D_3. The partial versions are compared with the original **TR-Net** following the setting described in Section 6.4.5.2. Table 6.5 shows the detection accuracy and visual quality results. We can see that the DeepFake samples with an individual trace being removed will particularly succeed in evading the corresponding type of detectors. They can also defeat other types of detectors, but the effect becomes weaker. In comparison, the **TR-Net-**D_2 samples show better transferability across different detector types than the **TR-Net-**D_1 and **TR-Net-**D_3 samples, implying that the spectral disparity may be the most significant feature differing DeepFakes from real images. However, the good transferability of **TR-Net-**D_2 is achieved at the cost of visual quality due to the overfitting in the frequency domain will lead to visual distortion. Compared to these partial versions, **TR-Net** removing all traces at once can result in the best trade-off between transferability and visual quality.



(a) The ProGAN group

(b) The STGAN group

(c) The DeepfakeTIMIT group

Figure 6.16: The power spectral density distributions of real, DeepFake and attack samples for three DeepFake types. The zoom in box highlights the main areas of high-frequency spectral distributional gaps.

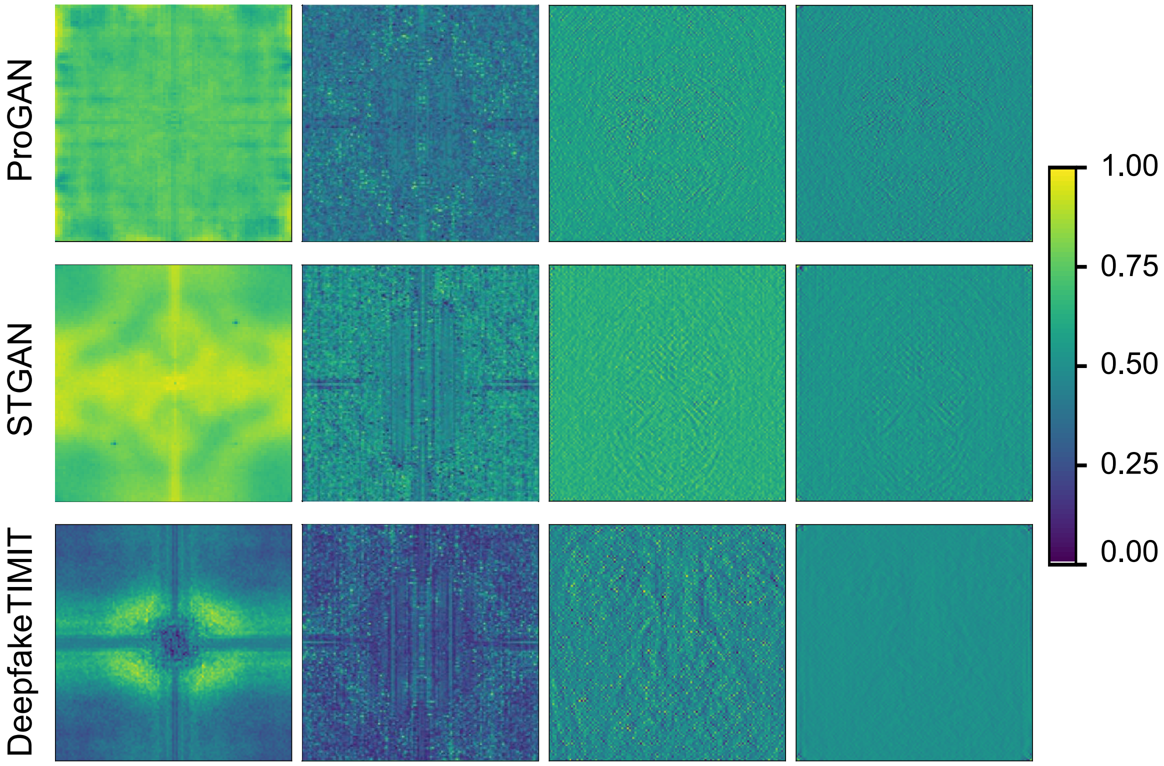


Figure 6.17: The average spectra and noise residual differences for three DeepFake types. The first two columns are spectrum differences between real and source fake samples and between real and attack samples, respectively; The last two columns are noise residual differences between real and source fake samples and between real and attack samples, respectively. A brighter entry means a bigger difference.

The above findings on the effect of each discriminator can be summarized as:

- The removals of spatial anomalies and noise fingerprint show similar attack performance, meaning that D_1 and D_3 may have equivalent importance;

- The removal of spectral disparity leads to the best attack performance, indicating that it is the most significant feature differing DeepFakes from real images, and thus D_2 should be strengthened during training.

- The removal of spectral disparity also leads to lower visual quality. Thus, a trade-off between attack success and visual quality should be concerned in deciding the best weights.

Table 6.5: The effect of individual trace removal

	Accuracy(%)	**TR-Net-D_1**	**TR-Net-D_2**	**TR-Net-D_3**	**TR-Net**
Spatial detectors	**Xception**	20.11	40.01	24.87	**17.90**
	Patch-CNN	27.02	45.10	29.00	**13.06**
Frequency detectors	**DCTA**	71.32	**14.31**	67.52	20.21
	F^3-Net	77.94	**28.99**	66.66	31.10
Fingerprint detectors	**LF**	45.17	40.23	43.10	**14.75**
	NF	35.90	38.71	40.79	**22.74**
Visual quality	PRNR (db)	**37.01**	33.12	36.67	35.16
	SSIM	0.991	0.954	**0.993**	0.988

6.5 Weight selection for discriminators

The weights $\{\lambda_1, \lambda_2, \lambda_3\}$ are imposed on the adversarial loss of the discriminators to balance the contribution of each discriminator, such that the three trace patterns can be removed in parallel. The method of weight selection is as follows:

Based on these findings on the effect of individual traces, we therefore test λ_2 in the range of $[0.1, 0.9]$ subject to $\lambda_1 + \lambda_2 + \lambda_3 = 1$ and $\lambda_1 = \lambda_3$. Figure 6.18 shows the results of detection accuracy and visual quality in terms of 1−Accuracy and SSIM scores. Note that we report the average accuracy over seven detectors in the "no defense" scenario. We can see that the best trade-off is achieved around $\lambda_2 = 0.6$. Thus, the final weight set $\{\lambda_1, \lambda_2, \lambda_3\}$ is set to $\{0.2, 0.6, 0.2\}$.

6.6 Summary

In this chapter, we focused on proposing an anti-forensics attack against DeepFake detectors. We presented a novel detector-agnostic attack, called a trace removal attack, that is capable of refining DeepFake images by removing all possible DeepFake traces via an one-versus-multiple adversarial learning network. The refined DeepFake images are closer to the real images and can therefore bypass arbitrary and even unknown detectors. We assessed the efficacy of the trace removal attack against a wide range of state-of-the-art detectors in heterogeneous high-level security scenarios where the detectors were embedded with various defensive strategies and the attacker's knowledge of data was limited. Our findings reveal that, the proposed trace removal attack achieves the highest attack effectiveness while introducing minimal visual quality loss compared

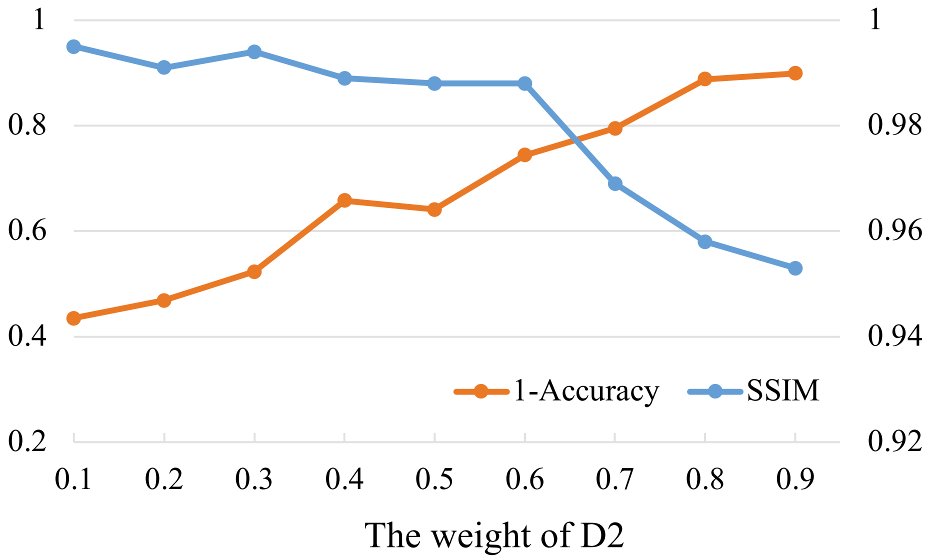


Figure 6.18: Weight selection for discriminators.

with contemporary adversarial and reconstruction-based attacks.

CHAPTER 7

FREQUENCY ALIGNMENT: A CLOSER LOOK AT FORENSICS AND ANTI-FORENSICS

As deep image forgery powered by GANs keeps challenging today's digital world, detecting GAN-generated forgeries has become a vital security topic. Generalizability and robustness are two critical concerns of a forgery detector, which in together determine the real-world reliability of a detector facing out-of-distribution forgery samples. However, the cause of the two problems has not been fully explored, and the link in between is unclear. Moreover, despite the recent achievements on the two problems from the forensic or anti-forensic aspect, a universal method that can simultaneously contribute to both sides is practically significant yet unavailable. In this chapter, we provide a fundamental explanation of the two problems from a frequency perspective. We reveal that the frequency bias of a DNN forgery detector is one dominant factor influencing generalizability and robustness. Based on the finding, we propose a two-step frequency alignment method for removing the frequency discrepancy between real and fake images, which has double-sided benefits: It can be used as a strong black-box attack against forgery detectors in the anti-forensic aspect or, inversely, in the forensic aspect, as a universal defense to improve detectors' reliability. The corresponding attack and defense implementations are also developed, and their performances, as well as the effect of frequency alignment, are evaluated in a variety of experimental settings involving ten detectors, eight forgery models, and five metrics.

7.1 Background

The recent progress in deep generative models, particularly generative adversarial network (GAN) [45], has remarkably advanced automated image processing techniques. Alongside the success, deep face forgery technologies powered by cutting-edge GAN models, such as DeepFake [135], are raising serious security concerns about individuals' safety [90, 112]. Research on countering forged face images has become a focus among security communities. One promising solution is developing deep learning-based detectors that can distinguish GAN-generated forged images from real ones [112]. Reliability is always a critical concern in developing a forgery detector, which determines whether or not the detector can apply to broader and real-world scenarios. The reliability of a detector is commonly assessed by two properties: the generalization ability to detect forged images created by unknown GANs; and the robustness against arbitrary perturbation attacks [49, 59, 146].

Existing achievements on the generalization and robustness problems can be divided into forensic and anti-forensic directions. The current forensic studies typically rely on designing sophisticated detector networks or feature engineering methods to improve on the two properties [14, 19, 49, 58, 59, 146, 159]. However, this kind of effort is incremental, outcome-driven, and case-by-case, which is insufficient to provide a fundamental solution and thus will become laborious and difficult as the technologies behind forgery GANs and perturbation attacks are continuously upgraded. The anti-forensic studies often conduct security analyses, some of which may design novel attacks, to reveal the robustness issues of a detector under attacks [4, 17, 34, 42, 55, 109, 160]. These works often end with empirical observations, underestimating the root cause of the vulnerability; and they focus solely on robustness. Moreover, the proposed attacks are often task-specific, requiring knowledge of the target detector, and poorly transferable across different detectors.

Revisiting the above challenges from both forensic and anti-forensic sides, we argue that one critical concern is that there has not been a high-level understanding that can fundamentally explain why deep neural network (DNN)-based forgery detectors easily suffer from generalization and robustness issues despite their outstanding learning capability, and it is unclear whether there is an intrinsic connection between generalizability and robustness. This knowledge, intriguingly, can benefit forensic and anti-forensic research simultaneously. For example, it could inspire a universal method to improve both the generalizability and robustness of a detector, or facilitate the design of a novel attack to evade arbitrary detectors.

To resolve the above problems, in this chapter, we step further toward the rationale underlying the detection problem of GAN-powered forgeries. First, we provide a fundamental explanation of the generalizability and robustness of GAN-generated image detectors from a frequency perspective. We establish an in-depth frequency analysis regarding the two properties, with which we point out that a specific frequency discrepancy between real images and forged images in the training dataset will lead to the *frequency bias* of DNN-based forgery detectors. The frequency bias is one dominant factor affecting generalizability and robustness and intrinsically concatenates the two properties: The frequency bias is principally associated with the higher-frequency components of the training images, making detectors much more sensitive to changes in high-frequency bands. As a result, a detector with significant frequency bias struggles to detect unknown GAN samples or attack samples, because both unknown GANs and attacks manifest different high-frequency patterns that are outside the frequency distribution of the training dataset.

Furthermore, based on the findings on the frequency bias, we propose a frequency alignment method to reduce the frequency discrepancy between an arbitrary type of forged images and real images, which can concurrently benefit the forensic and anti-forensic research on the generalization and robustness problems. The key idea of the frequency alignment method is to calibrate the frequency pattern of fake images according to real images. The method consists of two algorithms that enable a coarse-to-fine alignment. Spectral Magnitude Rescaling (SMR), the first algorithm, modifies the spectra of fake images by rescaling the magnitudes of their high-frequency components based on the estimated spectral distribution of real images. The second, Reconstructive Dual-domain Calibration (RDC), learns a functional model that maps the frequency pattern of fake images onto the real images' manifold via denoising reconstruction. The denoising reconstruction model is a self-supervised auto-encoder trained with only real images, with both image- and frequency-domain constraints to model the pixel and frequency distributions in latent space. The forensic and anti-forensic benefits of the method are demonstrated from attack and defence views respectively: It can be exploited directly as a strong black-box attack against forgery detectors. The frequency-aligned fake images are inherently closer to real images, *thus they can effectively evade arbitrary detectors without accessing the target detector*; Inversely, this method can serve as a universal defense for improving detectors,Äô generalizability and robustness via reducing their frequency bias. We accordingly devise three defense implementations based on the frequency-aligned samples, including pre-processing, data augmentation, and a novel hybrid defense, *all of which are free from detector-side modifications and thus compatible with various detectors*.

The contributions of this chapter are as follows:

- We established a comprehensive, unified frequency analysis framework for GAN-generated image detection. Through the analysis, we confirmed the frequency bias of DNN-based detectors, which can fundamentally explain several open problems related to the generalizability and robustness of DNN-based detectors.

- We proposed a universal two-step frequency alignment method for refining GAN-generated images by removing their frequency discrepancy from real images. The method can apply to fake images created by diverse forgery models, including different GANs and different perturbation attacks.

- The frequency alignment method can benefit the community from both forensic and anti-forensic sides. We proposed the corresponding attack and defense implementations, respectively, and verified the effects interactively in a wide range of settings. Ten baseline detectors, eight baseline forgery models, and five metrics are considered in the evaluation.

7.2 Frequency analysis of forgery detectors

This section presents an empirical analysis of the generalization ability and robustness of GAN-generated image forgery detection from the frequency perspective. GAN-generated forgery detection is commonly formulated as a binary "real/fake" classification problem [112]. The generalization defines the cross-GAN detection ability of the detector, i.e., whether the detector can predict accurately facing test fake samples generated by unseen GANs not included in the training set $\mathbb{D}$. The robustness measures the reliability of the detector in detecting noisy fake samples manipulated by certain perturbation attacks.

7.2.1 Frequency Analysis tools

Fourier transformation We adopt the 2D discrete Fourier transform (DFT) for image frequency analysis. Given an image $I \in \mathbb{R}^{M\times N}$, the frequency responses $\mathscr{F}(u,v)$ are computed as:

$$\mathscr{F}(I)(u,v) = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} I(x,y)\cdot e^{-2\pi i\cdot\frac{ux}{M}} e^{-2\pi i\cdot\frac{vy}{N}} \tag{7.1}$$
$$\text{for } x = 0,1,\ldots,M-1, \quad y = 0,1,\ldots,N-1$$

This transform is reversible and we denote the inverse DFT that transforms spectrum back to image as $\mathscr{F}^{-1}(\cdot)$. The DFT spectrum is typically visualized in a form of center-shifted magnitude heatmap, where lower frequency components are closer to the center of the spectrum while higher frequency components are farther from.

Frequency decomposition Our frequency analysis requires to decompose an image I into the low-frequency and high-frequency components, i.e., $I = \{I_L, I_H\}$. This can be done by applying a filter to the center-shifted DFT spectrum of the image. We use a circular mask-based ideal filter $\tau(r_0)$ with a predefined radius r_0 for decomposition, denoted as :

$$\begin{cases} I_L = \mathscr{F}^{-1}(\tau(r_0)\otimes\mathscr{F}(I)) \\ I_H = \mathscr{F}^{-1}((1-\tau(r_0))\otimes\mathscr{F}(I)) \end{cases} \tag{7.2}$$

where $\otimes$ is element-wise multiplication and each element in $\tau(r_0)$ is defined as:

$$\tau(r_0)_{u,v} = \begin{cases} 1, & \text{if} \quad \sqrt{(u-u_0)^2+(v-v_0)^2} \le r_0, \\ 0, & \text{otherwise,} \end{cases} \tag{7.3}$$
$$\text{for } u = 0,1,\ldots,M/2-1, \quad v = 0,1,\ldots,N/2-1$$

where (u_0, v_0) is the coordinate of the centroid. Figure 7.1 shows the decomposition of an image example with a certain radius.

Frequency distribution In order to straightforward observe the frequency discrepancy in a statistical view, we estimate the frequency distribution of a DFT spectrum. Given the rotation invariance of a the center-shifted DFT spectrum, the frequency distribution can be represented as a one-dimensional profile via azimuthally integrating the spectral magnitudes over the radial frequencies θ [32]. Assuming a square image $I \in \mathbb{R}^{N\times N}$, its one-dimensional profile is:

$$\mathrm{FD}(r_k) = C_0 \int_0^{2\pi} |\mathscr{F}(r_k,\theta)|\, \mathrm{d}\theta \quad \text{for} \quad k = 0,1,...,N/2-1, \tag{7.4}$$

where C_0 is a normalization constant, (r_k, θ) is the polar coordinate transformed from (u,v): $r_k = \sqrt{u^2+v^2}$, $\theta = \mathrm{atan2}(v,u)$. For ease we normalize r_k into the range of $[0,1]$ using the factor $\frac{1}{\sqrt{\frac{1}{2}N^2}}$, and use a log-scaled spectrum instead of the raw spectrum.

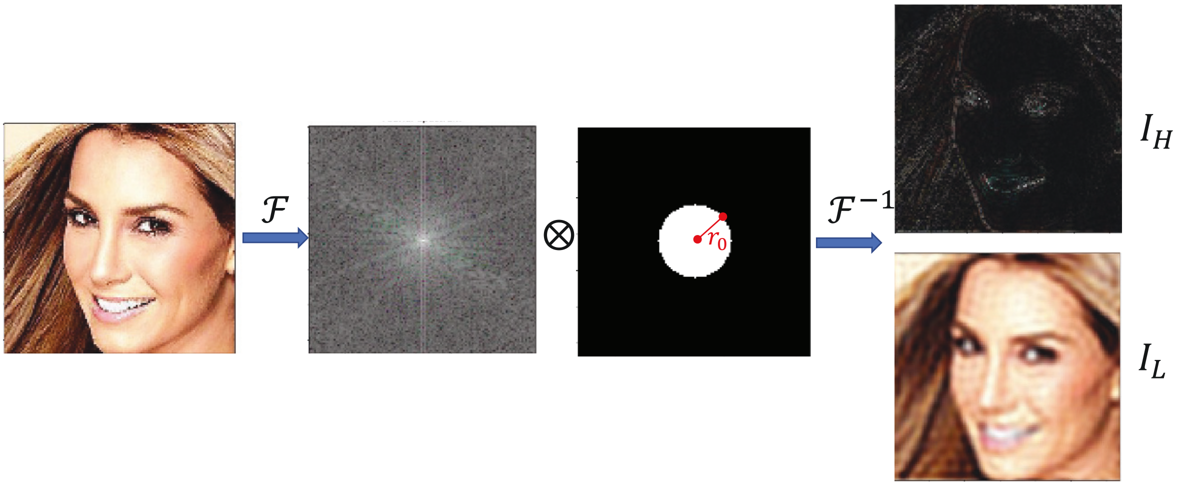


Figure 7.1: The process of frequnecy decomposition.

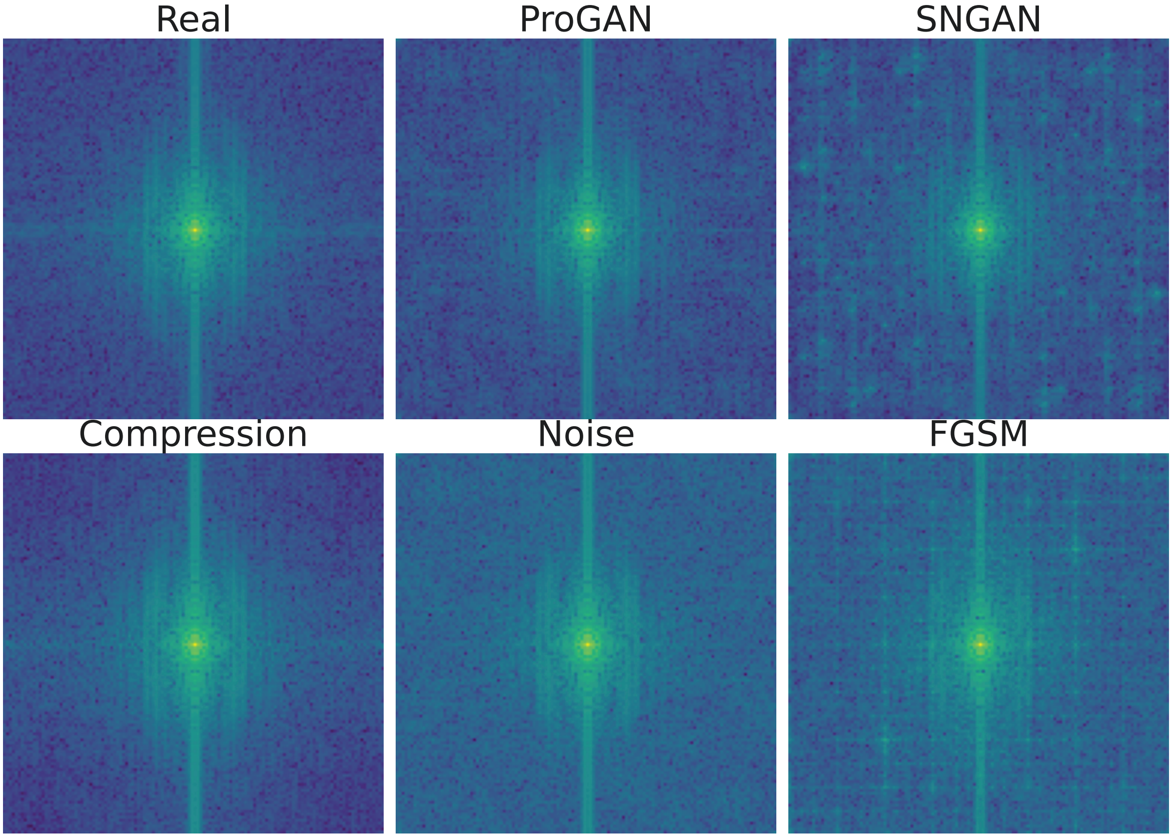


Figure 7.2: Visualization of the average DFT spectra of real images, GAN-generated images and the attack examples crafted based on ProGAN images. Specific discrepancy between real and each forgery type can be observed.

7.2.2 Visualization of frequency discrepancy

We first provide the spectral visualizations of different forgery models, including different GANs and different perturbations, to empirically figure out how the frequency discrepancies present. This information is helpful to the next analysis of detectors‚Äô generalization and robustness. The evaluations are conducted based on two popular GANs (ProGAN [65] and SNGAN [106]) and three representative perturbations crafted on ProGAN samples (Compression, Noising, and an adversarial attack FGSM ($\epsilon = 4/255$)). The settings are detailed in Section 7.4.1.

Figure 7.2 depicts the average DFT spectra of various forgery patterns. The average frequency distributions, as computed by Eq 7.4, are shown in Figure 7.3. Combining the two figures, the spectral discrepancies between real and fake images are clearly observed, along with two key findings: 1) Each forgery model has its specific frequency pattern, resulting in a unique spectral discrepancy from real images; 2) The spectral discrepancies generally become larger in higher frequency components, e.g., $r_k > 0.1$ for ProGAN. Notably, the findings hold for both GAN-generated samples and perturbed samples, implying a potential theoretical connection between generalization ability and robustness from the frequency standpoint.

7.2.3 Frequency bias of detectors

We next try to establish a unified explanation of the generalization and robustness problems of forgery detectors with the following hypothesis:

Finding 7.1 (The frequency bias of forgery detectors)**.** *A CNN detector easily overfits the specific high-frequency discrepancy between the forgery images and real images in the training set, and thus fails to detect test forgery samples with a different frequency discrepancy.*

The frequency bias hypothesis can simultaneously explain the generalization and robustness problems. This is because, whether the unseen forgery samples are generated by a different GAN or post-crafted by a perturbation attack, they consistently exhibit a specific frequency pattern distinct from the ones in the training set. As a result, if the hypothesis holds, a biased detector that has overfitted one specific pattern of frequency discrepancy will unsurprisingly fail to identify unseen forgery models with different frequency patterns. We provide two validations of the hypothesis.

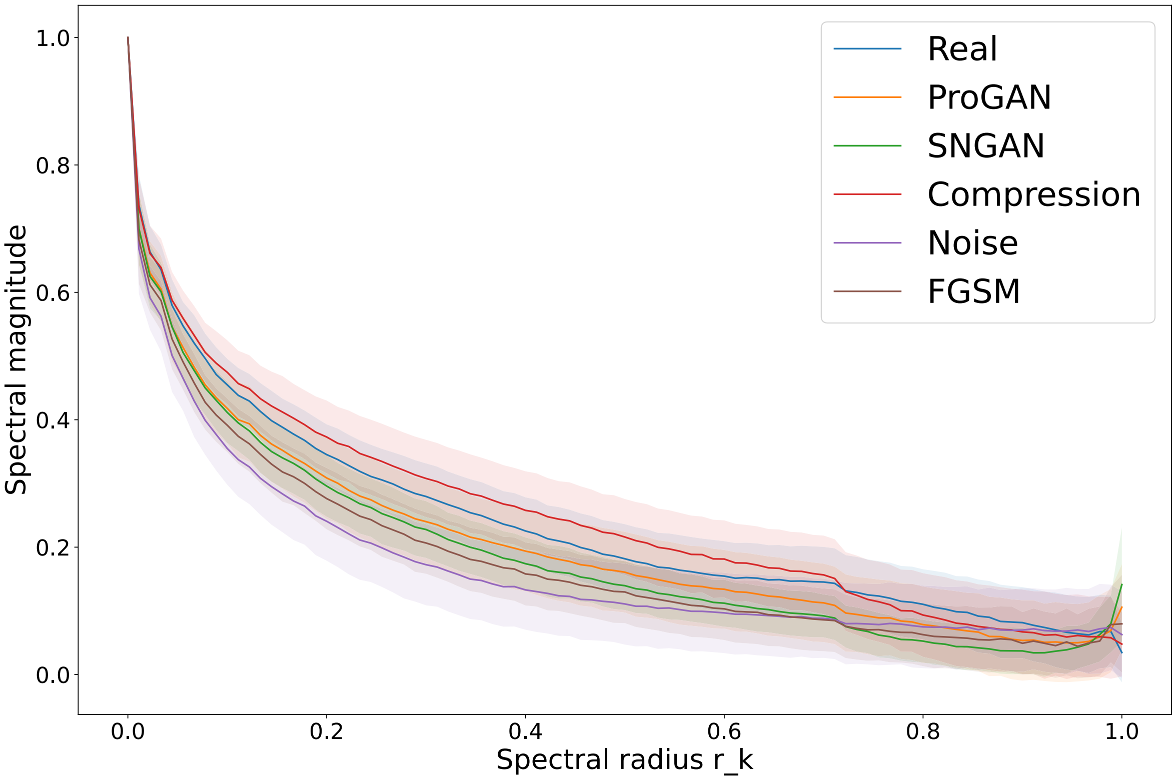


Figure 7.3: Visualization of the 1D spectral profiles real images, GAN-generated images and the attack examples crafted based on ProGAN images. The distributional gaps between real and each forgery type confirms the specific frequency discrepancy.

Validation 1: According to the observation of the high-frequency distributional characteristic of spectral discrepancy in Section 7.2.2, we aim to evaluate the responses of forgery detectors to different frequency components. Concretely, we decompose images into a set of pairs of $I = \{I_L, I_H\}_{r_0}$ by changing r_0 following Eq 7.2. Then we discard the high-frequency component I_H where the spectral discrepancies are likely to concentrate upon, and train and test the detectors with only the low-frequency components I_L. Two widely-used CNN-based forgery detector backbones, ResNet18 [48] and Xception [27], are evaluated. All detectors are trained with real and clean ProGAN images and tested on different forgery models. The settings are detailed in Section 7.4.1.

Figure 7.4 shows the results. When evaluating the raw images with full frequency information (i.e., no filter), the intra-dataset tests on the same forgery model ProGAN achieve high accuracy, while the generalization to SNGAN and the robustness against perturbations are poor. In the low-frequency groups, with decreasing the radius of the filter, which means more high-frequency components are excluded, the intra-dataset performances drop unsurprisingly due to information loss. However, the generalization

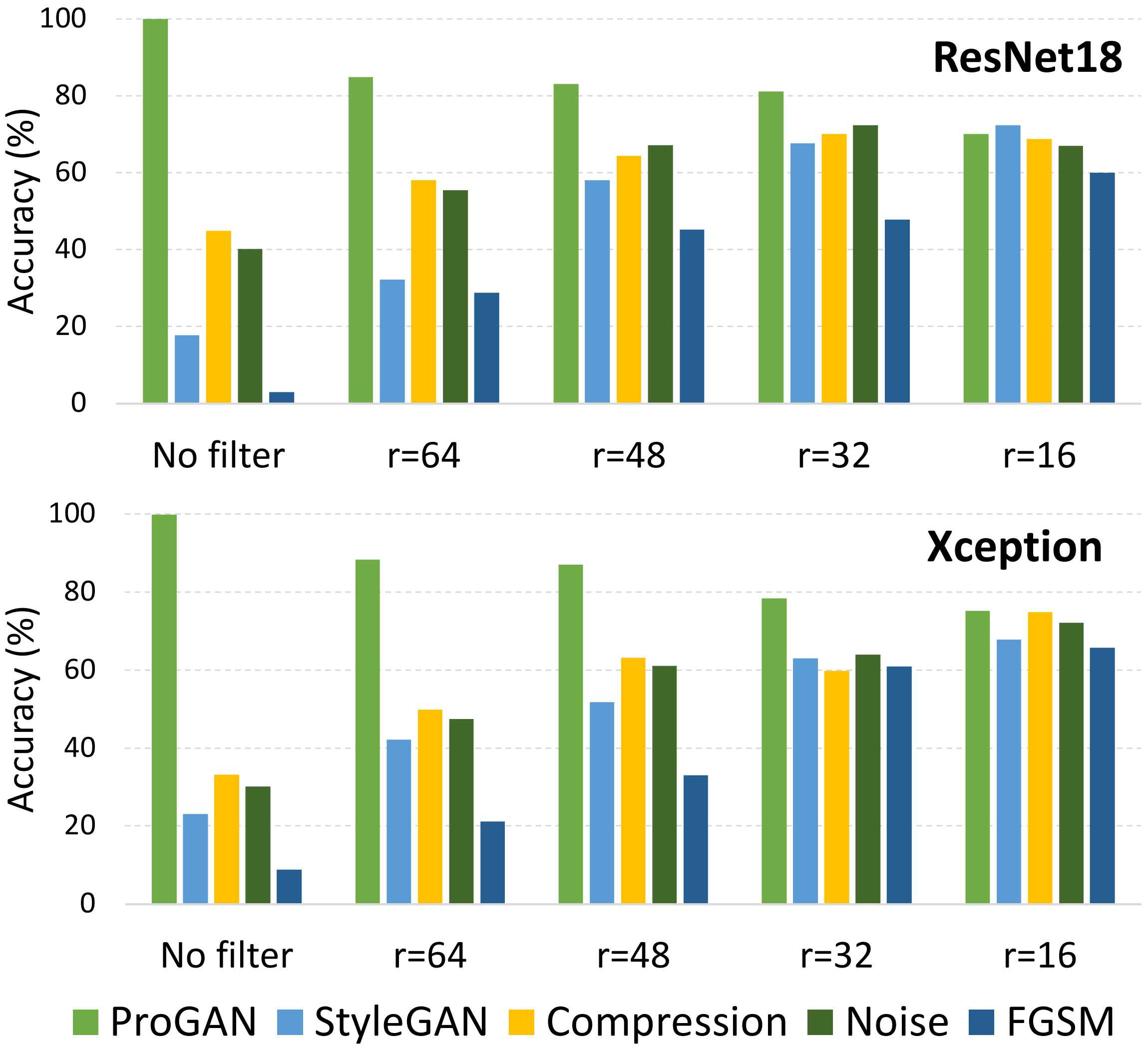


Figure 7.4: The generalization and robustness of two DNN detectors trained and tested on different frequency bands.

and robustness increase significantly for both ResNet18 and Xception, which means the detectors behave more stably after reducing the reliance on high-frequency discrepancy. The results confirm the frequency bias.

Validation 2: We further verify the hypothesis with integrating the Frequency Principle Theory of CNN classifiers.

Theorem 7.1 (Frequency Principle Theory of CNN)**.** *DNNs often fit target functions from low to high frequencies during the training process [151].*

The theory describes a CNN classifier's tendency to first pick up low-frequency information and then overfit high-frequency information when learning natural images

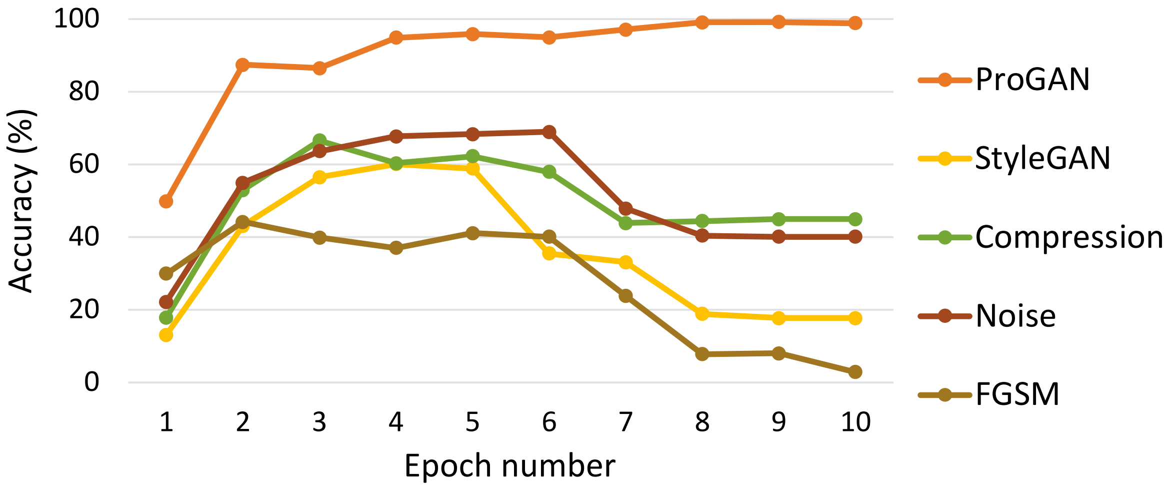


Figure 7.5: The generalization and robustness of the same DNN detector picked at different training epochs.

[144]. Applying the theory to forgery detectors, it can be deduced that detectors will exhibit a more severe frequency bias as training progresses. This is because frequency discrepancies primarily occur in higher frequency components which are mostly captured in later training phases. As a result, by evaluating the performance of the same detector at varying degrees of convergence, the influence of frequency bias can be verified.

To this end, we train a shallow CNN forgery detector using real and clean ProGAN images and test it with all forgery types at the end of each training epoch. Each epoch represents a certain convergence degree ranging from underfitting to overfitting. Figure 7.5 shows the results. Before the detector converges (i.e., epoch≤ 6), its test performances on unseen GANs or perturbations continue to improve. However, in later epochs, when the detector overfits more high-frequency information, the generalization ability and robustness both deteriorate remarkably. The outcomes again confirm the frequency bias.

7.2.4 Discussion

Through frequency analysis, we confirm the impacts of frequency discrepancy on detectors' generalization ability and robustness. The findings motivate us to rethink the forgery detection problem and develop a method to process fake images with eliminating their frequency discrepancy from real images, which will benefit forgery detection from the following opposing aspects:

The anti-forensic aspect: This method can be used as a strong black-box attack to evade forgery detectors. Unlike previous attacks fooling detectors by changing the frequency pattern of the original fake images, removing the frequency discrepancy of fake images makes them intrinsically closer to real ones. When serving as attack samples, the modified fake images will have better attack transferability across different detectors.

The forensic aspect: The method can also be used to improve detectors' generalization and robustness by retraining detectors with frequency-aligned samples. Since the frequency discrepancies are removed, the retrained detectors will become less dependent on unstable frequency patterns, reducing the frequency bias and alternatively focusing on learning more generic features.

7.3 The Frequency Alignment Method

7.3.1 Problem formulation

In this section, we propose the Frequency Alignment Method to eliminate the frequency discrepancy between real and fake images by aligning their frequency distributions. The problem can be formally formulated as follows:

Let $\mathbb{I}^+$, $\mathbb{I}^-$ be the original real and fake image datasets, respectively. We want a function $F : I^* = F(I^-)$ that can modify a given fake sample $I^- \in \mathbb{I}^-$ to I^* with satisfying the following goal:

$$\min D(q(I^*)||p(I^+)), \quad s.t. \quad \forall I^- \in \mathbb{I}^-, ||I^- - I^*|| \leq \epsilon \tag{7.5}$$

where $q(I^*)$ and $p(I^+)$ indicate the frequency distributions of fake and real samples, respectively, and $D()$ is the divergence measurement. The constraint term ensures that the modification of the original fake sample by F is small enough so that no perceptual image quality degradation is caused.

To solve the problem, we propose a two-step method to achieve a coarse-to-fine alignment. Figure 7.6 illustrates the overview. The first step is called Spectral Magnitude Rescaling (SMR). We rescale the spectral magnitudes of fake samples based on the estimated fitting function of real images' frequency distribution. The second step is called Reconstructive Dual-domain Calibration (RDC), where a denoising auto-encoder is first learned with only real images to model both the pixel and frequency distributions of real images. Then the rescaled fake samples generated by Step 1 are reconstructed by the auto-encoder with a dual-domain calibration to real images in the latent feature space.

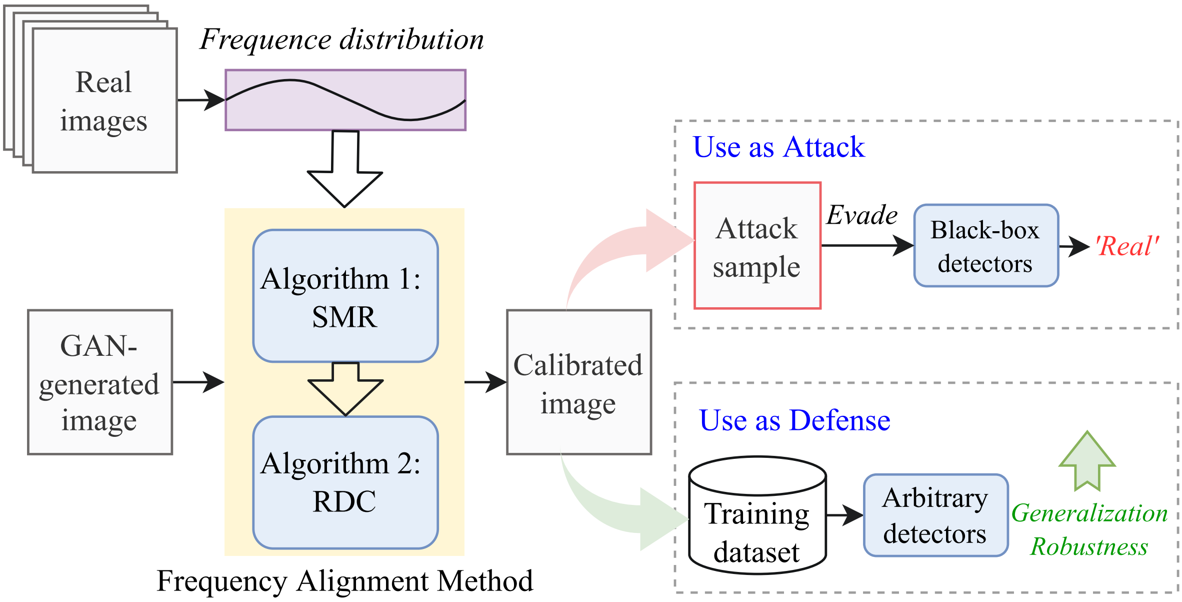


Figure 7.6: Overview of the proposed frequency alignment method and its different usages in attack and defense scenarios.

7.3.2 Spectral Magnitude Rescaling

The SMR algorithm aims to reduce the high-frequency gap between real and fake samples by rescaling fake samples' spectral magnitudes. The rescaling factor is adaptively computed at each frequency band according to the ratio of the empirical frequency distributions of real and fake images. To this end, we need to model the frequency distribution with an estimated parametric equation. As previous studies have pointed out that the spectra of natural images distribute following a power law [142], the expectation with respect to the frequency distribution can be modeled using a power law function:

$$\mathbb{E}(FD(r_k)) \approx a \cdot r_k^b \tag{7.6}$$

where the parameter a represents the spectral magnitude at the position r_k, and b represents the decay rate of the spectrum. The two parameters can be estimated by fitting the power law function with a number of images' one-dimensional spectral profiles $FD(r_k)$. Then, the spectrum of a given fake sample I^- can be rescaled as follows:

$$\hat{\mathscr{F}}(I^-)(r_k,\theta) = \mathscr{F}(I^-)(r_k,\theta)\left[\frac{a^+}{a^-}(r_k)^{b^+-b^-}\right] \tag{7.7}$$

where (a^+,b^+) and (a^-,b^-) are the parameters estimated from real images and fake images, respectively.

However, there remain two practical challenges. First, the visual contents of the given fake sample, such as facial details (e.g., profile, direction, and size), backgrounds, and color information, may significantly differ from the images sampled for fitting, leading to large visual distortions in the resulting fake sample. Second, considering the frequency discrepancy largely resides in high-frequency components, the rescaling is preferred to be performed specifically on high-frequency bands to reduce visual artifacts and computational overhead.

To overcome the challenges, we have two improvements to the algorithm. First, instead of randomly sampling image samples for fitting the function, we retrieve top-K similar samples that are visually close to the given fake sample from the real and fake image datasets individually. The retrieval is based on the Structural Similarity Index (SSIM) score. Second, we impose a threshold and smoothing factor to adjust the rescaling function in Eq.7.7:

$$
\begin{aligned}
\hat{\mathscr{F}}(I^-)(r_k,\theta) &= \mathscr{F}(I^-)(r_k,\theta)\left[1+\left(\frac{a^+}{a^-}(r_k)^{b^+-b^-}-1\right)S(r_k)\right], \\
S(r_k) &= \begin{cases} \dfrac{1}{1+e^{-(r_k-r_T)}}, & r_k \geq r_T, \\ 0, & r_k < r_T, \end{cases}
\end{aligned}
\tag{7.8}
$$

where r_T defines a fixed threshold frequency band above which the rescaling is performed to enforce the low-frequency bands unaffected, $S(r_k)$ is a sigmoid function when $r_k \geq r_T$ to smooth the rescaling. The entire SMR algorithm is shown in Algorithm 1 and the workflow is shown in Figure 7.7:

```
**Algorithm 1** Spectral Magnitude Rescaling
**Require:** The real image dataset $\mathbb{I}^+$; The fake image dataset $\mathbb{I}^-$; Sampling number $K$;
    Frequency threshold $r_T$; A given fake sample $I^-$;
**Ensure:** The spectrum-rescaled fake sample $\hat{I}^-$
    1. Retrieving the $K$ samples most similar to $I^-$ from $\mathbb{I}^+$ and $\mathbb{I}^-$ independently
    2. Computing the 1D spectral profile $FD(r_k)$ for all selected samples    ▷ following Eq. 7.4
    3. Estimating the parameters $(a^+, b^+)$ and $(a^-, b^-)$ by fitting a power law function on
    the sampled real and fake samples, respectively
    4. Transforming $I^-$ to its spectrum $\mathscr{F}(I^-)$
    5. Rescaling $\mathscr{F}(I^-)$ to $\hat{\mathscr{F}}(I^-)$    ▷ following Eq. 7.8
    6. Transform $\hat{\mathscr{F}}(I^-)$ back to the image domain: $\hat{I}^- = \mathscr{F}^{-1}\big(\hat{\mathscr{F}}(I^-)\big)$.
```

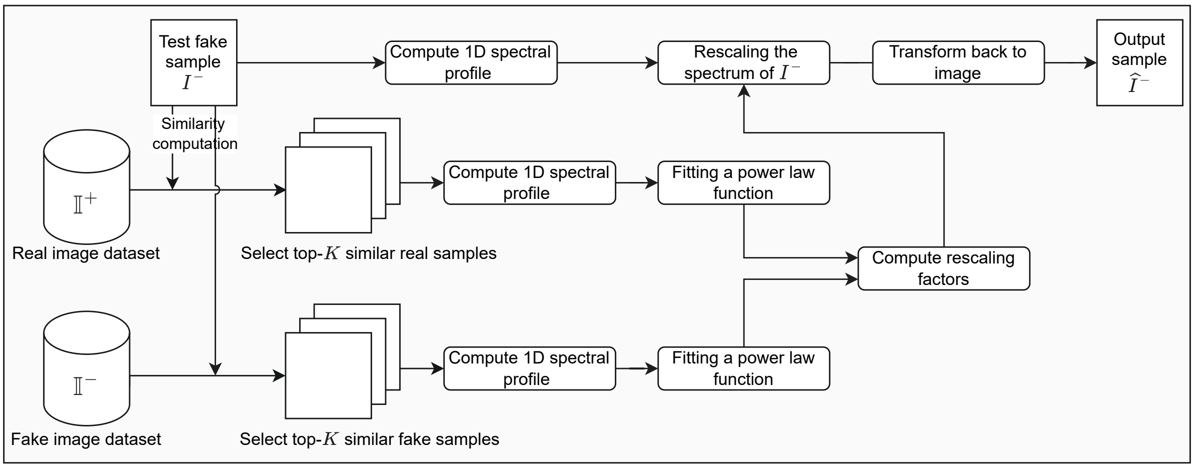

Test fake sample I^-
Compute 1D spectral profile
Rescaling the spectrum of I^-
Transform back to image
Output sample $\widehat{I}^-$
Similarity computation
$\mathbb{I}^+$
Real image dataset
Select top-K similar real samples
Compute 1D spectral profile
Fitting a power law function
Compute rescaling factors
$\mathbb{I}^-$
Fake image dataset
Select top-K similar fake samples
Compute 1D spectral profile
Fitting a power law function

Figure 7.7: The workflow of the Spectral Magnitude Rescaling algorithm.

7.3.3 Reconstructive Dual-domain Calibration

Although the SMR algorithm can reduce the high-frequency gap between real and fake samples, it will still remain several high-frequency artifacts. The reasons include that the visual contents of the real and fake samples selected for fitting do not exactly match, and the estimation of the fitting function is an empirical approximation. In order to further align the frequency patterns while satisfying the constraint of visual quality in Eq. 7.5, a more fine-grained calibration is needed. We propose the Reconstructive Dual-domain Calibration (RDC) algorithm. The key idea is to simulate both the pixel and frequency distributions of real images via a learnable model, and then use the model to calibrate the fake images resulting from the SMR algorithm in both the image and frequency domains.

7.3.3.1 Self-supervised denoising

We formulate the simulation of real images as a learning-based denoising process, i.e., try to reconstruct the original real image from its noised version by an auto-encoder $A(\cdot)$. As shown in Figure 7.8, the auto-encoder is trained with the real image dataset $\mathbb{I}^+$ only. The correct pixel and high-frequency distributions of real images are then captured by the auto-encoder through reconstruction learning. In the inference phase, the well-trained $A^*(\cdot)$ is applied to reconstruct a given fake sample. The dual-domain calibration is completed in the latent feature space formed by $A^*(\cdot)$.

To ensure an accurate calibration from the spectrum-rescaled fake samples to the real images, the noised real images, i.e., the inputs of $A(\cdot)$, should be initialized to a similar

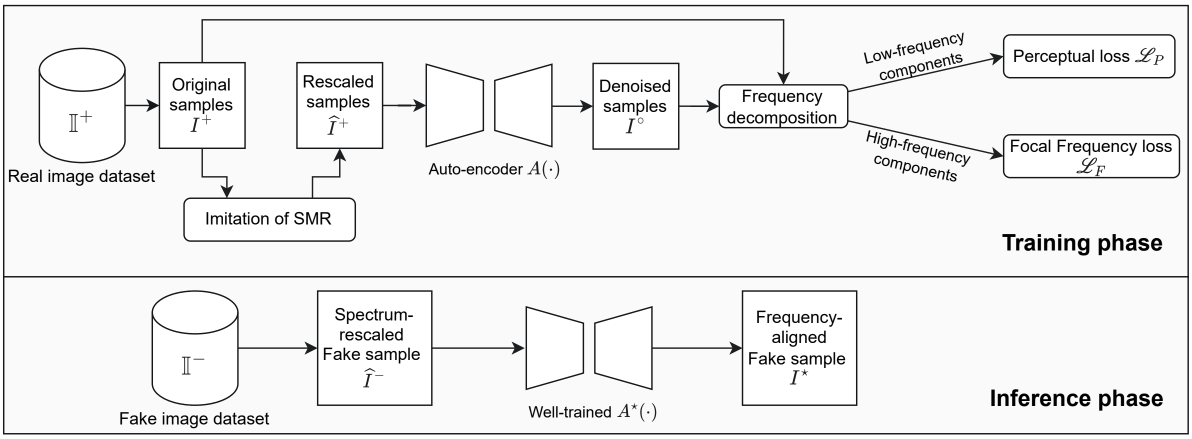

$\mathbb{I}^+$
Real image dataset
Original samples I^+
Rescaled samples $\hat{I}^+$
Imitation of SMR
Auto-encoder $A(\cdot)$
Denoised samples I°
Frequency decomposition
Low-frequency components
High-frequency components
Perceptual loss $\mathscr{L}_P$
Focal Frequency loss $\mathscr{L}_F$
Training phase
$\mathbb{I}^-$
Fake image dataset
Spectrum-rescaled Fake sample $\hat{I}^-$
Well-trained $A^*(\cdot)$
Frequency-aligned Fake sample I^*
Inference phase

Figure 7.8: The workflow of the Reconstructive Dual-domain Calibration algorithm. In the training phase, a self-supervised denoising auto-encoder A is trained with only real images, learning to model the distribution of real images in both the image and frequency domains. In the inference phase, the well-trained A^* is applied to fake samples to calibrate the frequency patterns.

pattern as the spectrum-rescaled fake samples. However, the SMR algorithm cannot be applied directly to real images because real images themselves are the ground-truth reference for power law fitting. As an alternative, we propose an approximation method to imitate the effect of SMR. Given a real image I^+, we compute its noised version $\hat{I}^+$ as follows:

$$\begin{aligned} \hat{\mathscr{F}}(I^+)(r_k,\theta) &= \mathscr{F}(I^+)(r_k,\theta)\left[1+\left(a'(r_k)^{b'}-1\right)S(r_k)\right], \\ \hat{I}^+ &= \mathscr{F}^{-1}\left(\hat{\mathscr{F}}(I^+)\right). \end{aligned} \tag{7.9}$$

where $S(r_k)$ is the same as in Eq. 7.8. a' and b' are randomly sampled from $[1/2,2]$ and $[-4,4]$ respectively to simulate the rescaling factor in Eq. 7.8. Then, $A(\cdot)$ can be trained in a self-supervised reconstruction task, as shown in Figure 7.8.

7.3.3.2 Network and Losses

We adopt a U-shape encoder-decoder [120] as the backbone of $A(\cdot)$, given its remarkable capacity to reconstruct high-quality images. As shown in Figure 7.9, the U-Net we use has an encoder and an decoder with the same number of building blocks. The encoder comprises repeated convolutional layers (with $3*3$ kernels) and max pooling layers (with $2*2$ kernels), which can learn features at different scales while compacting the spatial information. The decoder is an expanding symmetric counterpart of the encoder. Besides

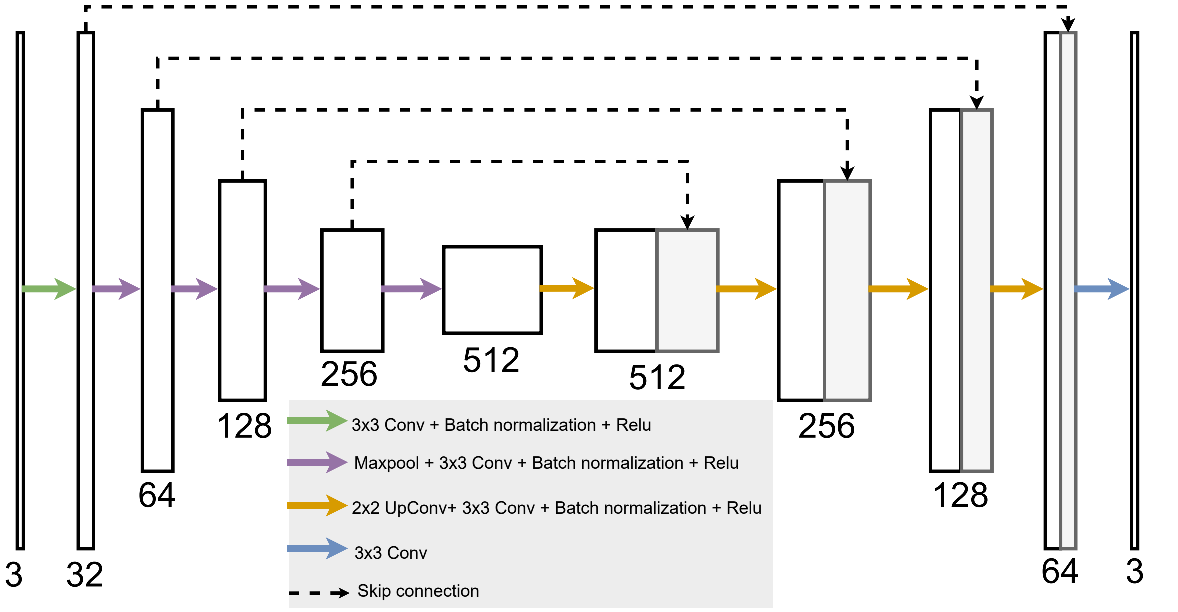

3
32
64
128
256
512
512
256
128
64
3
3x3 Conv + Batch normalization + Relu
Maxpool + 3x3 Conv + Batch normalization + Relu
2x2 UpConv+ 3x3 Conv + Batch normalization + Relu
3x3 Conv
Skip connection

Figure 7.9: The network of the denoising auto-encoder A.

forwarding the feature fro one layer to the next layer, the encoder and the decoder are also connected via skip connection, where the upstream feature of each encoder's layer is also concentrated with the downstream feature of the corresponding decoder's layer. The skip connection strengthens feature representation and preserve spatial information to facilitate reconstruction.

The loss function supervises $A(\cdot)$ to reconstruct I^+ from $\hat{I}^+$. For our dual-domain reconstruction task, the conventional pixel-to-pixel reconstruction loss, $||I^+ - A(\hat{I}^+)||$, is impractical. As the majority of pixel information in a natural image is associated with low-frequency bands, the pixel-to-pixel loss can easily lead to a suboptimum that overfits the low-frequency component. Moreover, it fails to solve the issue of spectral artifacts caused by upsampling [32]. We approach this problem by decomposing images into low- and high-frequency components and addressing each component separately, detailed as follows:

Given the original sample I^+ and its reconstructed version $I^\circ = A(\hat{I}^+)$, following Eq. 7.2, we decompose them into $(I_L^+, I_H^+)_r$ and $(I_L^\circ, I_H^\circ)_r$ respectively, with a random radius threshold r. With regard to the low-frequency components, we compute the perceptual loss [61] to measure the pixel similarity:

$$\mathscr{L}_P = \frac{1}{K}\sum_{k=0}^{K-1} \left\| \mathrm{VGG}_k(I_L^+) - \mathrm{VGG}_k(I_L^\circ)) \right\|, \tag{7.10}$$

where $\mathrm{VGG}_k(\cdot)$ is the respective feature obtained by the k-th convolutional layer of a total of K convolutional layers within a pre-trained VGG classification network. The perceptual loss can better recover the low-frequency visual details that correlate with the human visual system compared with the pixel-to-pixel loss. For the high-frequency components, we transform them into DFT spectra and compute the focal frequency loss [60] to measure the frequency similarity:

$$\begin{aligned} \mathscr{L}_F &= \frac{1}{MN}\sum_{u=0}^{M-1}\sum_{v=0}^{N-1} w(u,v)\left|\mathscr{F}(I_H^+)(u,v) - \mathscr{F}(I_H^\circ)(u,v)\right|^2 \\ w(u,v) &= \left|\mathscr{F}(I_H^+)(u,v) - \mathscr{F}(I_H^\circ)(u,v)\right| \end{aligned} \tag{7.11}$$

where $w(u,v)$ is a self-adaptive weight to force the model to focus more on higher frequency. The final training objective function is:

$$\min \mathscr{L} = \mathscr{L}_P + \lambda \mathscr{L}_F, \tag{7.12}$$

In the inference phase, the optimal model $A^*(\cdot)$ is applied to the spectrum-rescaled fake samples to create the final frequency-aligned samples with dual-domain calibration, denoted as:

$$I^* = A^*(\hat{I}^-) \tag{7.13}$$

7.3.4 Compared with other methods

Compared with low-pass filter. The low-pass filter is a straightforward way to remove the frequency discrepancies, given that the discrepancies heavily rely on high-frequency components. However, in most cases, the filter has a fixed kernel that fails to support sample-specific alignments. Also, filtering will cause unnecessary loss of high-frequency information. In contrast, our method involves a coarse-to-fine alignment based on learning the frequency pattern of real images, which is more flexible and smoother and can perfectly preserve the full-band information.

Compared with frequency regularization. Some recent studies have proposed imposing an additional frequency regularization loss [32] or frequency discriminator [25, 64] on the source GAN to suppress its frequency distortion during training. Unlike our algorithm, which is post-processing and applicable to various forgery models, these methods only work for one specific GAN, require retraining the source GAN, and cannot be used to align forgery samples post-processed by perturbation models.

Compared with adversarial learning. Another typical idea recently proposed is to train a model to directly reconstruct the fake samples via adversarial learning [29, 89].

Alongside the reconstruction generator, a discriminator is needed to distinguish the reconstructed fake images from real ones in the frequency domain during training. Compared with our RDC algorithm trained only on real images, this kind of methods requires a large number of fake samples from various forgery models for training, which is hard to acquire in practice. Moreover, the discriminator will also suffer from the frequency bias [144, 151], resulting in lower visual quality and alignment precision, as confirmed in our experiments.

7.3.5 Attack and defense implementations

As discussed earlier, the frequency alignment method enables both the anti-forensic (attack) and forensic (defense) usages. The implementations are as follows:

Attack implementation. Given an arbitrary victim detector $\mathscr{C}$, we perform the frequency alignment method to modify a fake sample I^- into I^*. The modified sample I^* is much more realistic and can directly serve as an attack sample to evade the detection by $\mathscr{C}$. Notably, as an attack, our method is detector-independent, requiring zero knowledge of $\mathscr{C}$. Thus, it works well for challenging black-box scenarios and has good cross-detector transferability.

Defense implementation. The frequency alignment method can be used to improve the detector's generalization ability and robustness by forcing the detector to mine more generic frequency-irrelevant features while reducing frequency bias. We propose three implementation methods:

(1) Implementation as a simple data augmentation in the training phase. We set the probability of a training sample being modified by our method to 0.5.

(2) Implementation as a pre-processing procedure in both the training and inference phases of the detector.

(3) Hybrid implementation: We pre-process all training and test samples with the frequency alignment method, and also employ a mix-up augmentation with a probability of 0.5. The mix-up augmentation is denoted as:

$$\begin{cases} I^+_{aug} = I^+ + \delta|I^- - I^*| \\ I^-_{aug} = I^* + \delta|I^- - I^*| \end{cases} \tag{7.14}$$

The key idea is adding the residual $|I^- - I^*|$ to the raw training samples to create hard learning samples. $\delta \sim \mathscr{N}(0,1)$ is used to scale the residual to enable various degrees of hardness.

Note that all the defense methods are free of modifying the detector's network and are therefore compatible with various detectors.

7.4 Experiments

In the experiments we aim to

(1) evaluate the performances of the proposed frequency alignment method implemented as attack and defense separately, both in a wide range of settings; and

(2) verify the success of the proposed method in frequency alignment quantitatively and qualitatively.

7.4.1 Datasets

Real-world face image dataset: The real images are from the CelebA [93], a large-scale image dataset consisting of more than $200k$ real-world celebrity face photos. We randomly sample $22,000$ images from CelebA as the real image dataset $\mathbb{I}^+$. All these images are cropped down to the resolution of $128 * 128 * 3$ with the face in the center, and the face directions are aligned.

GAN-forged face dataset: We select two powerful and representative GANs, ProGAN and STGAN as the source forgery GAN models to create fake face samples. The two GANs follow the official implementations and are pre-trained with the entire CelebA dataset, allowing them to generate high-fidelity forgery samples. For each GAN, we query $22,000$ images to construct the fake image dataset $\mathbb{I}^-$.

For each class, the $22,000$ images are randomly divided into $20,000$ and $2,000$ as training set and test set, respectively. The proposed frequency alignment method is developed with the above datasets.

Perturbed forgery images: We craft different types of perturbed forgery images based on the $2,000$ ProGAN test images as reference attack samples. We consider three common image processing perturbation [39, 158], a gradient-based adversarial attack FGSM, and two attacks specific to GAN-generated images, GANprintR [111] and the state-of-the-art method TR-Net [89] (also see Chapter 6). The configurations are as follows:

(1) Blurring: images are blurred with a Gaussian filter with a kernel size randomly sampled from $\{3,5,7,9\}$.

(2) Compression: images are JPEG-compressed with a quality factor randomly sampled from $U(10,75)$.

(3) Noising: images are embedded with i.i.d Gaussian noise with a variance randomly sampled from $U(5.0, 20.0)$.

(4) FGSM: the adversarial examples are crafted based on the gradient of a vanilla well-trained ResNet detector with two sets of noise amount constrains, i.e., $\epsilon \in \{4/255, 8/255\}$

(5) GANprintR and TR-Net: we follow the settings in the original papers to generate attack samples.

7.4.2 Evaluation metrics

Attack performance. Following previous anti-forensics studies [4, 17, 34, 42, 55, 109, 160], we evaluate an attack by error rate (ER) and image quality. The ER score is computed as the percentage that test fake samples are mis-classified into 'real' by the detector. Image quality is measured by two widely-used image quality metrics peak signal-to-noise ratio (PSNR) and structural similarity index measure (SSIM). PSNR quantifies the amount of noise that affects the fidelity of an image. SSIM measures the similarity between an original fake image and the corresponding attack sample.

Defense performance. We report the detection accuracy on fake images (Acc) computed as the percentage that test fake samples are correctly classified to show the performance of a detector.

Real-referenced Spectral Profile Distance. In order to quantify the average frequency discrepancy between the ground-truth real images and the test (fake) images, we additionally propose a novel metric called Real-referenced Spectral Profile Distance (RSPD), defined as follows:

$$\text{RSPD} = \frac{1}{N/2}\left(\sum_{k=0}^{N/2-1}\left|\overline{FD^{+}}(r_k) - \overline{FD^{test}}(r_k)\right|\right), \tag{7.15}$$

where $\overline{FD^{+}}(r_k)$ and $\overline{FD^{test}}(r_k)$ are the mean 1D spectral profile (Eq. 7.4) averaged over K real images and K test (fake) images, respectively. A lower RSPD score means a smaller frequency discrepancy.

7.4.3 Experiment configuration

Regarding the SMR algorithm, we set the sampling number K to 200, and the frequency threshold r_T to 0.2 for all experiments. For the RDC algorithm, we train the auto-encoder $A(\cdot)$ with the 20,000 CelebA training samples. The batch size is 80. We use the Adam optimizer [73] with initial learning rates of $1.6e-3$ plus a decay rate of 0.5 executed at the end of an epoch if the loss stopped decreasing. The loss weight λ is 10. We also use random Gaussian noise, color jitter, and blurring and rotation for data augmentation for training $A(\cdot)$.

7.4.4 Results of the attack implementation

7.4.4.1 Victim detectors

To demonstrate the transferablity of the attack, we consider a large variety of victim detectors:

Normal detectors: We employ three image-domain detectors based on pixel input, including a ResNet18 and a Xception which are two popular forgery detector backbones, and the GAN fingerprinting model (GF) [158] which learns model fingerprint for detection. We also employ three frequency-domain detectors, one trained with the DCT coefficients (DCT) [39], one with 1D spectral profile (1d-SP) [32], and the Spatial-Phase Shallow Learning (SPSL) which combines RGB image with phase spectrum [91].

Specific detectors: We also consider four detectors with specific design for improving generalization ability and robustness, including the spectral artifacts simulation method (AutoGAN) [159], the data augmentation-based method (DA) [146], the frequency-level adversarial perturbation method (FLAP) [59], and the re-synthesis residual method (RSR) [49].

All detectors are trained with $20,000$ CelebA and $20,000$ ProGAN images and tested with different types of perturbed ProGAN images ($2,000$ per type).

7.4.4.2 Attack performance

Table 7.1 illustrates the performances of eight attack methods in terms of ER and RSPD scores. All detectors except AutoGAN achieve fairly high accuracy in detecting clean ProGAN samples, while their performances degrade when facing attack samples. In the group of normal detectors, our attack evades all detectors, with results comparable to or better than the state-of-the-art attack TR-Net. We also emphasize that the success of FGSM against ResNet18 is not surprising because the attack samples are crafted directly based on the gradient of ResNet18.

Compared with the normal detector group, the results in the group of specific detectors are more encouraging. Although the effects of almost all attacks are diminished against detectors with strengthened generalization ability and robustness, our attack still leads to relatively high ER scores for all detectors. Moreover, its superiority over other attacks in this group is much more significant than in the normal group. This superiority is fully explicable from the frequency perspective. In this group, all the strategies used for enhancing the detectors' generalization ability and robustness can be interpreted as a kind of frequency-domain augmentation, which increases the variety of frequency patterns in the training set and thereby reduces the detector's frequency bias. For example, DA augments the training set with JPEG compression and Gaussian noise, which expands the frequency diversity of the original training samples; FLAP takes one step further by directly generating adversarial perturbations onto the spectra of the original training samples. As a result, one attack will be less effective against the frequency-augmented detectors if it simply modifies the frequency pattern of the original fake samples rather than eliminating the frequency discrepancy between real and fake images, as our method does. The phenomenon also confirms our hypothesis of frequency bias. We additionally show the RSPD scores, which directly measure the frequency gap between real and fake images, in the last row of Table 7.1 to support the conclusion. Our attack manifests an RSPD score of 0.22, which is over ten times lower than the second-best score obtained by the TR-Net attack, confirming the success of frequency alignment.

Table 7.1: The evaluation of the attack implementation in terms of error rate (ER, %, ↑) and Real-referenced Spectral Profile Distance (RSPD, %, ↓). A total of Eight attack methods against ten representative detectors are evaluated. The best result in each row is in bold. Comp.: short for Compression; GR: short for GANprintR.

	Clean	Blurring	Comp.	Noise	FGSM	FGSM	GR	TR-Net	FA
					(ϵ=4/255)	(ϵ=8/255)	[111]	[89]	(ours)
ResNet18 [48]	0.08	54.65	54.15	39.17	98.61	**100.00**	37.62	80.23	90.10
Xception [27]	0.01	46.31	41.26	50.30	78.85	**81.66**	25.12	75.50	80.31
GF [158]	0.11	64.20	50.91	47.37	55.55	67.00	40.99	**85.12**	81.32
DCT [39]	0.06	41.53	43.00	38.46	60.01	66.31	20.13	80.01	**87.06**
1d-SP [32]	2.63	84.99	61.51	65.03	53.21	54.47	49.52	95.98	**100.00**
SPSL [91]	0.06	35.17	33.85	29.60	43.60	50.12	28.88	**69.51**	64.51
Avg. ER #1	0.49	54.48	47.45	44.99	64.97	69.93	33.71	81.06	**83.88**
AutoGAN [159]	18.13	20.02	31.02	30.21	40.69	45.31	75.96	82.33	**88.16**
DA [146]	0.03	11.36	3.63	3.21	63.70	68.83	16.34	70.03	**72.11**
FLAP [59]	1.56	4.33	16.42	10.50	40.98	35.01	11.12	62.21	**76.36**
RSR [49]	0.01	8.16	10.66	8.19	30.12	29.63	6.89	40.79	**67.21**
Avg. ER #2	4.93	10.97	15.43	13.03	43.87	44.70	27.58	63.84	**75.96**
RSPD (%) ↓	4.44	9.31	4.16	13.51	8.03	12.22	5.31	2.36	**0.22**

Table 7.2: The evaluation of image quality of eight attack methods in terms of the SSIM (↑) and PSNR (↑) scores. The best result in each row is in bold.

	Blurring	Compression	Noise	FGSM (ϵ=4/255)	FGSM (ϵ=8/255)	GANprintR	TR-Net	FA (ours)
PSNR ↑	28.21	33.10	30.01	31.21	30.13	27.64	35.11	**37.91**
SSIM ↑	0.714	0.886	0.766	0.812	0.760	0.901	**0.982**	0.976

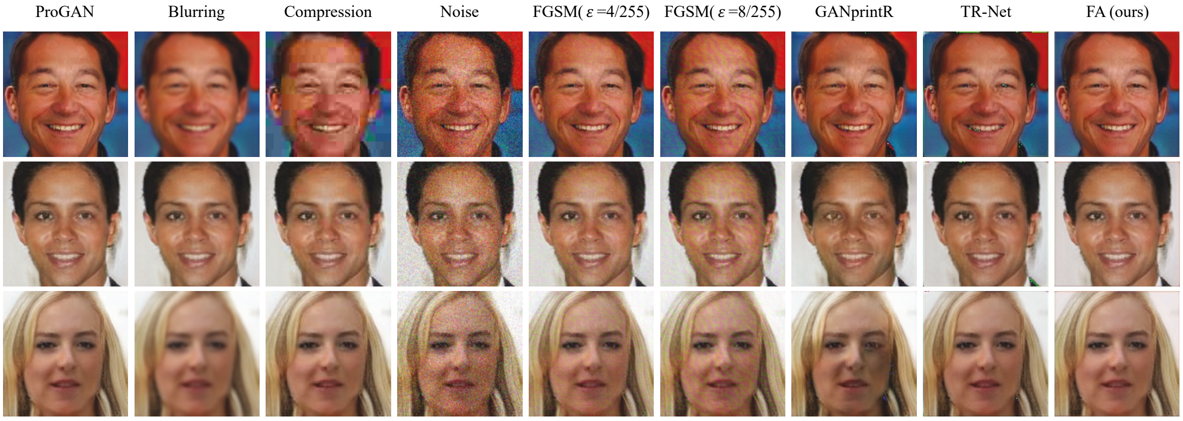


Figure 7.10: The visualization of three original ProGAN image examples and the corresponding attack samples created by eight attack methods.

7.4.4.3 Image quality

Another concern about an attack is whether it can maintain the image quality as high as the original fake sample. Table 7.2 shows the PSNR and SSIM scores of all attack methods. We can see that, alongside the pronounced attack success, the proposed method achieves the highest PSNR score (37.91) and the second-best SSIM score (0.976) compared with other attacks. Figure 7.10 additionally offers several image examples for explicit visualization. Compared with other attacks, the distortion and noise introduced to the original fake samples by our method are the smallest, which is imperceptible to human eyes. We also emphasize the comparison with TR-Net. Even though TR-Net has a slightly higher SSIM score than our method, we can see in Figure 7.10 that it leads to visible point-like noises on the attack samples due to the side effect of adversarial learning discussed in Section 7.3.4.

In summary, the results of attack implementation confirm the feasibility of the proposed frequency alignment method as a novel black-box attack. The aligned fake samples are intrinsically closer to real images by removing the frequency discrepancy while maintaining a high visual quality, resulting in great attack transferability across various detectors.

7.4.5 Results of the defense implementation

Next, we evaluate the effectiveness of the frequency alignment method as a defense strategy for improving a forgery detector's generalization ability and robustness. We evaluate the three different implementation protocols of the frequency alignment method (denoted as FA-P1, FA-P2 and FA-P3, respectively) outlined in Section 7.3.5, plus two baselines, including training with the original dataset (Original) and training with the mixture data augmentation method proposed in [146] (MDA). ResNet18 and Xception are selected as the target detectors. For each detector, we train it from scratch five times, each time with an individual strategy.

7.4.5.1 Generalization

We first evaluate the generalization ability. The detectors are trained with images from a single GAN and tested with different GANs. Table 7.3 shows the detection accuracy of two detectors in different test groups. The right and left sides of the arrow indicate the sources of training samples and test samples, respectively. For example, "P→S" means training with ProGAN images and testing with SNGAN images. If the two sides of the arrow are different, it is a cross-GAN test group.

As shown in Table 7.3, both of the original ResNet18 and Xception can achieve high detection accuracy in the intra-dataset tests P → P and S → S, even without any defense. However, when generalized to the cross-GAN tests, their performances drop considerably. For example, the Acc score of the ResNet18 trained with the ProGAN images decreases from 99.21% in the P → P group to 12.31% in the P → S group. The results of the original detectors indicate that a normal DNN can excessively learn the difference between real images and images generated by a specific GAN, which may lead to overfitting. By comparison, after implementing a defense strategy, the generalization abilities of both detectors get highly improved in all cross-GAN tests.

Among the four defense strategies, FA-P2 and FA-P3 are much more effective than MDA and FA-P1 in enhancing the cross-GAN generalization ability while maintaining the intra-dataset detection accuracy. One possible reason is that MDA and FA-P1 are both based on data augmentation, which reduces the frequency bias of the detector by improving the frequency diversity in the training set only. In contrast, FA-P2 and FA-P3 implement frequency alignment as a pre-processing module for both the training and test samples. It can pull all samples to the same distribution in the frequency domain prior to detection, so as to eliminate the frequency bias in detection.

Table 7.3: The evaluation of generalization ability in the defense implementation in terms of detection accuracy (Acc, %, ↑). A total of five defense protocols are evaluated in four test groups. The best result in each column is in bold.

Protocols		P→P	S→S	P→S	S→P
ResNet18	Original	99.92	99.19	12.31	31.36
	MDA [146]	98.78	98.12	59.21	67.23
	FA-P1 (ours)	98.91	98.22	61.41	68.45
	FA-P2 (ours)	99.61	**100.00**	81.13	**86.10**
	FA-P3 (ours)	**100.00**	**100.00**	**83.20**	85.21
Xception	Original	**100.00**	99.93	32.18	40.77
	MDA [146]	98.65	98.55	54.13	73.09
	FA-P1 (ours)	98.81	98.23	60.20	72.67
	FA-P2 (ours)	99.36	**100.00**	81.32	82.69
	FA-P3 (ours)	**100.00**	**100.00**	**87.02**	**84.03**

7.4.5.2 Robustness

We next evaluate the robustness against different attacks. The detectors are trained with the clean ProGAN images and tested with the seven types of attack samples described in Section 7.4.1. Table 7.4 demonstrates the results in terms of detection accuracy. The original DNN-based detectors suffering from severe frequency bias are vulnerable to various perturbation attacks, especially the adversarial attack FGSM, resulting in low detection accuracy scores. When being strengthened by defenses that can mitigate the frequency bias, the detectors become more reliable in classifying attack samples.

Regarding different defense strategies, similar to the results of generalization ability in Table 7.3, FA-P2 and FA-P3 are generally more effective than MDA and FA-P1 when dealing with all attack types. Note that MDA also performs well in the Compression and Noise groups. This is because MDA uses compression and noise for data augmentation; thus, it becomes a de facto white-box defense in the two groups. In comparison, the proposed FA-P2 and FA-P3 are more practical since they work evenly well for different attacks without knowing the attack setting.

In summary, the results of defense implementation showcase the potential of the proposed frequency alignment method being a universal strategy for improving generalization and robustness of a forgery detector. It is effective for various unknown forgery patterns and compatible with different detectors.

Table 7.4: The evaluation of robustness in the defense implementation in terms of detection accuracy (Acc, %, ↑). A total of five defense protocols are evaluated against seven attack methods. The best result in each column is in bold. Comp.: short for Compression; GR: short for GANprintR.

Protocols		Blurring	Comp.	Noise	FGSM (ϵ=4/255)	FGSM (ϵ=8/255)	GR [111]	TR-Net [89]
ResNet18	Original	45.35	45.85	60.83	1.39	0.00	62.38	19.77
	MDA [146]	76.79	90.11	**93.68**	49.88	49.91	70.43	31.69
	FA-P1 (ours)	70.46	74.70	77.01	65.39	60.93	71.12	58.97
	FA-P2 (ours)	86.71	87.27	87.45	**85.04**	84.99	**91.78**	79.59
	FA-P3 (ours)	**88.03**	**90.89**	90.39	81.89	**85.64**	90.55	**80.80**
Xception	Original	53.69	58.74	49.70	21.15	18.34	74.88	24.50
	MDA [146]	80.50	90.88	**91.91**	43.13	46.74	74.10	51.95
	FA-P1 (ours)	68.80	67.56	73.23	70.39	69.46	72.88	58.00
	FA-P2 (ours)	**89.98**	89.69	88.89	83.72	82.97	89.41	75.48
	FA-P3 (ours)	88.22	**91.37**	89.74	**87.10**	**86.19**	**90.78**	**83.36**

7.4.6 The effect of frequency alignment

7.4.6.1 Visualizations

The key effect of the proposed frequency alignment method is that it can align the frequency pattern of an arbitrary type of fake image to real images, eliminating the frequency discrepancy and making fake images intrinsically closer to real ones. We now provide some visualizations to confirm the effect.

First, we visualize the average DFT spectra and frequency distributions of real and frequency-aligned fake images. We select the forgery types covered in Section 7.2 for a straightforward comparison. Figure 7.11 and Figure 7.12 display the average DFT spectra and the frequency distributions, respectively. We can see that after frequency alignment, different types of fake images all manifest a spectral pattern similar to the pattern of real images, compared with Figure 7.2. Furthermore, the frequency distributions of the aligned fake images now become consistent with real images, in contrast to the substantial distributional gaps exhibited in Figure 7.3. The results confirm the success of frequency alignment as well as the broad applicability of the proposed method to various forgery types.

To further demonstrate that the frequency alignment method enables fake images to be truly closer to real images, we visualize the changes in DNN detectors' latent feature space caused by frequency alignment. Figure 7.13 shows the features of real, original ProGAN, and frequency-aligned ProGAN images extracted from the last con-

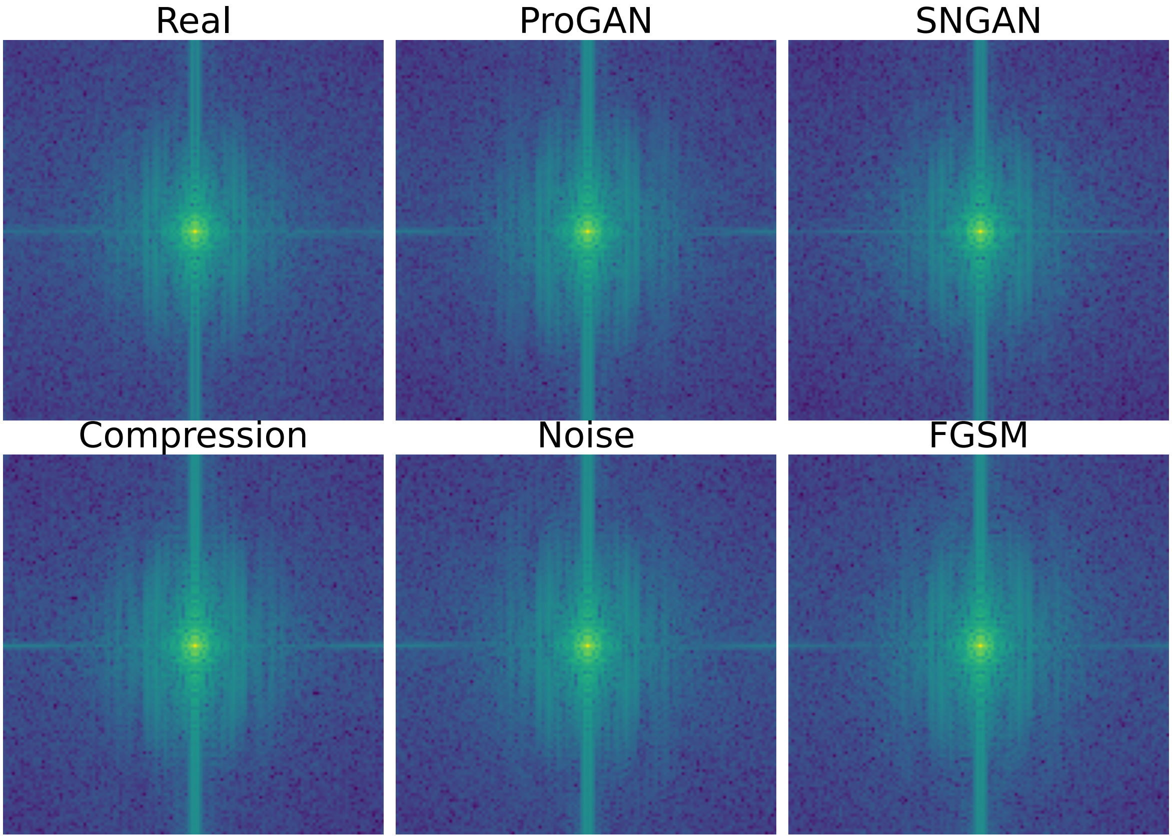


Figure 7.11: Visualization of the average DFT spectra of real images and different types of frequency-aligned fake images. A direct comparison can be made with Figure 7.2.

volutional layer of the ResNet18 and Xception detectors. Features are clustered into a two-dimensional space by T-SNE [141] for visualization. As shown in the figure, the features of frequency-aligned ProGAN images are entangled with those of real images, while far away from the features of original ProGAN images. The results also explain why the frequency-aligned fake images can be used as attack samples to evade detection.

Table 7.5: The PSNR, SSIM and RSPD scores of different frequency alignment methods.

	BLF ($r_0 = 0.2$)	BLF ($r_0 = 0.5$)	BLF ($r_0 = 0.8$)	FA-AL	FA-SMR	FA-RDC	FA-Final
PSNR ↑	28.33	29.39	37.02	36.71	32.17	36.01	**37.91**
SSIM ↑	0.721	0.749	0.961	0.959	0.903	0.955	**0.976**
RSPD ↓	8.17	5.90	4.92	4.09	1.08	0.62	**0.22**

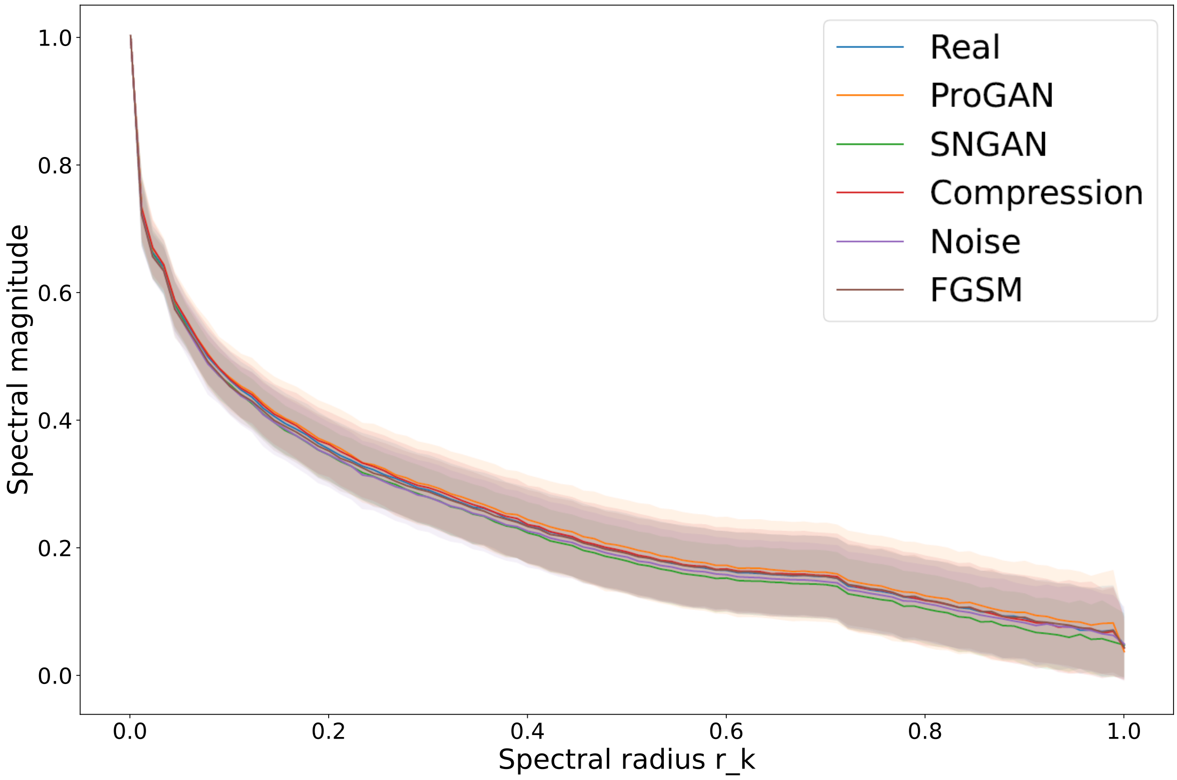


Figure 7.12: Visualization of the spectral profiles of real images and different types of frequency-aligned fake images. The distributional gaps are removed compared with Figure 7.3.

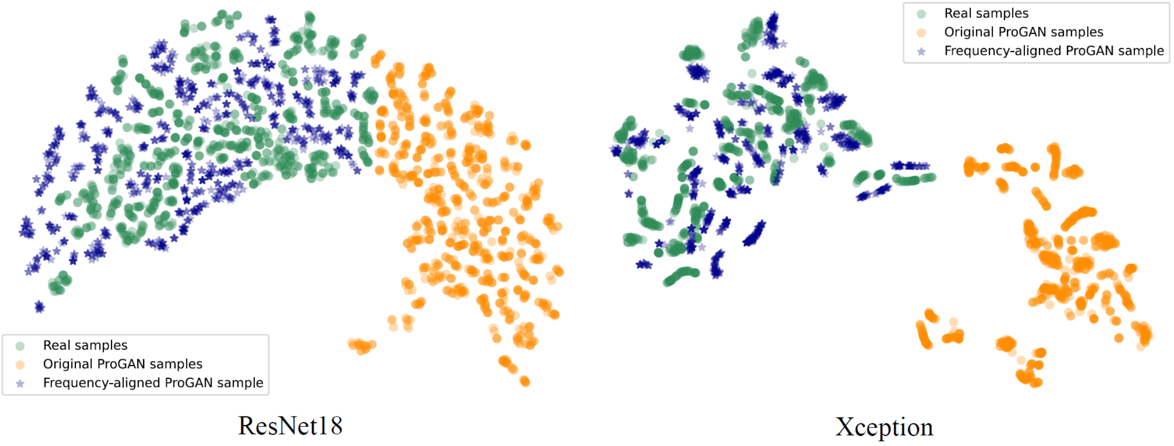


Figure 7.13: The features of real, original ProGAN, and frequency-aligned ProGAN images in the latent spaces of ResNet18 and Xception detectors.

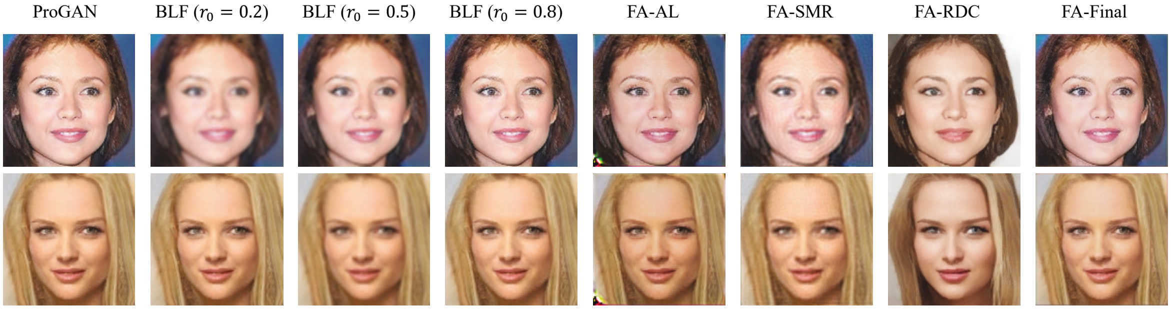


Figure 7.14: Examples of the original ProGAN images and the versions modified by different frequency alignment methods.

7.4.6.2 Comparison and ablation study

We compare the proposed frequency alignment method to other possible methods outlined in Section 7.3.4. For the low-pass filter method, we test a Butterworth Low-pass Filter (BLF) with varying cut-off bands r_0. Regarding the adversarial learning method, we redesign our RDC algorithm by adding a frequency discriminator D complement to the auto-encoder A and train them adversarially, denoted as FA-AL. The frequency regularization methods require modifying the source GAN, which is not applicable to this evaluation. We also provide an ablation study that compares the individual SMR (FA-SMR) and RDC (FA-RDC) algorithms to the final combined version (FA-Final).

We assess these methods based on the frequency alignment and image quality trade-offs. Table 7.5 shows the results in terms of PSNR, SSIM and RSPD scores. Figure 7.14 depicts some examples of the original ProGAN images and the versions modified by the above methods. Compared with other learning-based methods, BLF with a fixed cut-off band is inflexible and fails to align the frequency accurately. Moreover, when the cut-off band is small, the image quality is readily degraded. Adopting an adversarial discriminator like in FA-AL raises the SSIM and PSNR scores, but falls short in frequency alignment in terms of the RSPD score. This is because, as discussed in Section 7.3.4, the discriminator will also suffer from the frequency bias. In addition, despite the high SSIM and PSNR scores, the adversarial learning method will leave particular visual speckles on the output images, such as in the lower-left corner of the image, as shown in Figure 7.14. Comparatively, the frequency alignment effect achieved by the individual SMR or RDC algorithm surpasses the baselines remarkably. However, the image quality remains a concern: FA-SMR will produce conspicuous visual artifacts, whilst FA-RDC will cause color and local texture distortions, confirmed in Figure 7.14. The final combined version FA-Final, obtains the highest PSNR, SSIM and RSPD scores, indicating the best trade-off between frequency alignment and image quality.

7.5 Summary

Deep image forgeries powered by cutting-edge GANs, has become a new threat in this area due to their high fidelity and full automation. Reliable detection of this type of forgery is impelling. The generalization ability and robustness of the detector are critical concerns that determine the real-world reliability of the detector but have yet to be fully explored. In this chapter, we have taken a step further to fundamentally explain the two concerns and link them together from a frequency perspective. We discovered that the specific frequency discrepancy between real and fake images causes the frequency bias of DNN-based forgery detectors, which influences the detectors' generalization ability and robustness. Then, we proposed a two-step frequency alignment method to eliminate the frequency discrepancy between real and fake images. The method provides easy-to-implement solutions to benefit both forensic and anti-forensic research: it can be exploited as a strong black-box attack to evade forgery detectors; or be used as a universal defense to reduce the frequency bias of forgery detectors so as to improve their generalization ability and robustness. We also proposed the corresponding attack and defense implementations and experimentally demonstrated their effectiveness as well as the frequency alignment effect in a variety of tests. Our study lays the foundation for the reliability of deep forgery detectors.

Part IV

Conclusion

Chapter 8

Conclusion

8.1 Discussion and future directions

In this thesis, we investigate the deep image forgery problem from both the forensic and anti-forensic perspectives. From Chapter 4 to Chapter 7, four novel technical solutions are presented to mitigate existing challenges in this field. Revisiting the challenges identified in Chapter 1, we would like to emphasize that, based on our best knowledge, there are four imperative ones necessitating unremitting future research attention, which are the generalization and robustness of forensic techniques and the transferability and black-box feasibility of anti-forensic techniques.

- **Generalization**. The cross-model generalization is a vital property determining the reliability of the forgery detectors in practice because, in most real-world cases, the forged samples are from unknown GAN models that are not included in the training dataset. More importantly, the generalization ability shapes the bottleneck of overcoming the future technical update on image forgery models. In addition, although in this thesis we primarily focus on human face images as a face is a significant biometric, the generalization ability across different semantic domains is also very important, considering the wide applicability of GANs.

- **Robustness**. The robustness against perturbation attacks is another crucial property for the in-the-wild reliability of forgery detection because forged images are mostly propagated on the Internet, where they may undergo unknown perturbations such as compression and resizing for communication needs. Moreover, robustness is a key requirement to defend against smarter attackers who are able to implement some attacks to make the forgeries evasive against detectors.

- **Transferability**. Regarding the anti-forensic attacks, their transferability is a dominant concern. Similar to the generalization ability of forgery detectors, transferability defines the capability of an attack against the latest detectors. Meanwhile, an attack with good transferability can help reveal the common shortcomings of different detectors rather than detector-specific weaknesses. Good transferability can also enable the anti-forensic investigation to be cost-effective if a lot of detectors can be covered at once.
- **Black-box feasibility**. In most cases, black-box attacks are more helpful in exposing the common, fundamental weakness of detectors compared with white-box attacks. Also, black-box attacks are more practical since no knowledge of the target detector is needed; thus, they can also reduce the cost of the anti-forensic investigation.

In addition, this thesis mainly focuses on detecting GAN-generated images, while in practice, many DeepFake services also involve post-processing such as video compression and rendering. Detection of GAN-generated images is the first and foundation step of DeepFake detection, but may not be able to cover all post-processed DeepFakes. Although the performances of the proposed methods are demonstrated theoretically and experimentally, we must be aware that there is still a long way to go in combating deep image forgery, given the rapid, persistent technical iterations on deep image forgery. As mentioned earlier, this is a long-lasting battleground for the security community.

8.2 Conclusion

The widespread use of AI-generated deep image forgeries is putting the age-old adage "seeing is believing" to the test. Deep image forgery has become a major social concern due to the privacy and security threats it poses, such as misleading information spread online. Forgery detection systems that can tell fake from real images are powerful countermeasures. As the technology behind deep image forgery evolves, so must the countermeasures. With the help of two competing branches of technology, forensics and anti-forensics, we can better understand and address existing challenges in developing reliable detection systems.

In this thesis, we investigate the problem of deep image forgery detection and attempt to address some of the outstanding issues that have recently arisen in this area. To offer comprehensive solutions, we conduct research into the issue from both forensic and anti-forensic vantage points. In the forensics realm, we propose two forgery detection methods, one of which uses multi-level GAN model fingerprinting to enable task-specific forensics. The other employs a multi-view reconstruction and classification learning framework for generalized and robust detection. When it comes to anti-forensic investigation, we have developed a new black-box attack, the trace removal attack, to deceive forgery detection systems. We have also provided a frequency-based analysis of the generalization and robustness problems in deep image forgery detection, which bridges the gap between forensic and anti-forensic studies by means of a novel frequency alignment technique.

We hope the thesis can offer some new insights into the deep image forgery problem and help raise research awareness of the highlighted challenges, including the generalization and robustness of forensic techniques and the transferability and black-box feasibility of anti-forensic techniques, to further perfect the countermeasures against deep image forgery. Our aspiration is to eventually win the cat-and-mouse game, so as to build a forgery-free, mutually trusted online environment.

Bibliography

[1] S. B. A, H. T. S. B, and N. M. B, *Improvements on source camera-model identification based on cfa interpolation*, in Proc. of WG, 2006.

[2] B. Alec, *What are deepfakes & why the future of porn is terrifying*.
Highsnobiety, February 2018.
`https://www.highsnobiety.com/p/what-are-deepfakes-ai-porn/`. Accessed: 2023-01-03.

[3] V. Asnani, X. Yin, T. Hassner, and X. Liu, *Reverse engineering of generative models: Inferring model hyperparameters from generated images*, IEEE Transactions on Pattern Analysis and Machine Intelligence, (2023).

[4] M. Barni, K. Kallas, E. Nowroozi, and B. Tondi, *On the transferability of adversarial examples against cnn-based image forensics*, in ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2019, pp. 8286–8290.

[5] ——, *Cnn detection of gan-generated face images based on cross-band co-occurrences analysis*, in 2020 IEEE international workshop on information forensics and security (WIFS), IEEE, 2020, pp. 1–6.

[6] B. Bayar and M. C. Stamm, *Augmented convolutional feature maps for robust cnn-based camera model identification*, in 2017 IEEE International Conference on Image Processing, ICIP 2017, Beijing, China, September 17-20, 2017, IEEE, 2017, pp. 4098–4102.

[7] ——, *Constrained convolutional neural networks: A new approach towards general purpose image manipulation detection*, IEEE Trans. Information Forensics and Security, 13 (2018), pp. 2691–2706.

[8] S. Bayram, H. T. Sencar, N. D. Memon, and I. Avcibas, *Source camera identification based on CFA interpolation*, in Proceedings of the 2005 International

Conference on Image Processing, ICIP 2005, Genoa, Italy, September 11-14, 2005, IEEE, 2005, pp. 69–72.

[9] M. G. Bellemare, I. Danihelka, W. Dabney, S. Mohamed, B. Lakshminarayanan, S. Hoyer, and R. Munos, *The cramer distance as a solution to biased wasserstein gradients*, arXiv preprint arXiv:1705.10743, (2017).

[10] M. Bińkowski, D. J. Sutherland, M. Arbel, and A. Gretton, *Demystifying MMD gans*, in ICLR, 2018.

[11] G. K. Birajdar and V. H. Mankar, *Digital image forgery detection using passive techniques: A survey*, Digital investigation, 10 (2013), pp. 226–245.

[12] J. Brainard, J. You, et al., *What a massive database of retracted papers reveals about science publishing's 'death penalty'*, Science, 25 (2018), pp. 1–5.

[13] J. Brandon, *Terrifying high-tech porn: Creepy 'deepfake' videos are on the rise*. Fox News, June 2018. `http://www.foxnews.com/tech/2018/02/16/terrifying-high-tech-porn-creepy-deepf` `html`. Accessed: 2023-01-03.

[14] T. Bui, N. Yu, and J. Collomosse, *Repmix: Representation mixing for robust attribution of synthesized images*, in European Conference on Computer Vision, Springer, 2022, pp. 146–163.

[15] E. J. Candès, L. Demanet, D. L. Donoho, and L. Ying, *Fast discrete curvelet transforms*, Multiscale Model. Simul., 5 (2006), pp. 861–899.

[16] H. Cao and A. C. Kot, *Identification of recaptured photographs on LCD screens*, in Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2010, 14-19 March 2010, Sheraton Dallas Hotel, Dallas, Texas, USA, IEEE, 2010, pp. 1790–1793.

[17] 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.

[18] N. Carlini and D. Wagner, *Towards evaluating the robustness of neural networks*, in 2017 ieee symposium on security and privacy (sp), IEEE, 2017, pp. 39–57.

[19] 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.

[20] K. Chandrasegaran, N.-T. Tran, and N.-M. Cheung, *A closer look at fourier spectrum discrepancies for cnn-generated images detection*, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 7200–7209.

[21] B. Chen, X. Liu, Y. Zheng, G. Zhao, and Y.-Q. Shi, *A robust gan-generated face detection method based on dual-color spaces and an improved xception*, IEEE Transactions on Circuits and Systems for Video Technology, (2021).

[22] D. Chen, N. Yu, Y. Zhang, and M. Fritz, *Gan-leaks: A taxonomy of membership inference attacks against generative models*, in Proceedings of the 2020 ACM SIGSAC conference on computer and communications security, 2020, pp. 343–362.

[23] M. Chen, J. J. Fridrich, M. Goljan, and J. Lukás, *Determining image origin and integrity using sensor noise*, IEEE Trans. Information Forensics and Security, 3 (2008), pp. 74–90.

[24] W. Chen, Y. Q. Shi, and G. Xuan, *Identifying computer graphics using HSV color model and statistical moments of characteristic functions*, in Proceedings of the 2007 IEEE International Conference on Multimedia and Expo, ICME 2007, July 2-5, 2007, Beijing, China, IEEE Computer Society, 2007, pp. 1123–1126.

[25] Y. Chen, G. Li, C. Jin, S. Liu, and T. Li, *Ssd-gan: Measuring the realness in the spatial and spectral domains*, in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, issue 2, 2021, pp. 1105–1112.

[26] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, *Stargan: Unified generative adversarial networks for multi-domain image-to-image translation*, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 8789–8797.

[27] F. Chollet, *Xception: Deep learning with depthwise separable convolutions*, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 1251–1258.

[28] S. Dehnie, H. T. Sencar, and N. D. Memon, *Digital image forensics for identifying computer generated and digital camera images*, in Proceedings of the International Conference on Image Processing, ICIP 2006, October 8-11, Atlanta, Georgia, USA, IEEE, 2006, pp. 2313–2316.

[29] F. Ding, G. Zhu, Y. Li, X. Zhang, P. K. Atrey, and S. Lyu, *Anti-forensics for face swapping videos via adversarial training*, IEEE Transactions on Multimedia, (2021).

[30] M. N. Do and M. Vetterli, *The finite ridgelet transform for image representation*, IEEE Trans. Image Process., 12 (2003), pp. 16–28.

[31] C. Dong, A. Kumar, and E. Liu, *Think twice before detecting gan-generated fake images from their spectral domain imprints*, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 7865–7874.

[32] 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.

[33] 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.

[34] L. Fan, W. Li, and X. Cui, *Deepfake-image anti-forensics with adversarial examples attacks*, Future Internet, 13 (2021), p. 288.

[35] H. Farid, *Creating and detecting doctored and virtual images*, Implications to The Child Pornography Prevention Act, (2004), pp. 280–291.

[36] ——, *Digital image ballistics from jpeg quantization*, in Dartmouth College, 01 2006.

[37] ——, *Photo tampering throughout history*, [online] `http://www.cs.dartmouth.edu/farid/research/digitaltampering`, (2011).

[38] H. Farid and S. Lyu, *Higher-order wavelet statistics and their application to digital forensics*, in IEEE Conference on Computer Vision and Pattern Recognition, CVPR Workshops 2003, Madison, Wisconsin, USA, 16-22 June, 2003, IEEE Computer Society, 2003, p. 94.

[39] 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.

[40] J. Fridrich, *Digital image forensics*, IEEE Signal Processing Magazine, 26 (2009), pp. 26–37.

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

[42] A. Gandhi and S. Jain, *Adversarial perturbations fool deepfake detectors*, in 2020 international joint conference on neural networks (IJCNN), IEEE, 2020, pp. 1–8.

[43] R. C. González and R. E. Woods, *Digital image processing, 3rd Edition*, Pearson Education, 2008.

[44] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, *Generative adversarial nets*, Advances in neural information processing systems, 27 (2014).

[45] ——, *Generative adversarial networks*, Communications of the ACM, 63 (2020), pp. 139–144.

[46] I. J. Goodfellow, J. Shlens, and C. Szegedy, *Explaining and harnessing adversarial examples*, arXiv preprint arXiv:1412.6572, (2014).

[47] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, *Masked autoencoders are scalable vision learners*, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 16000–16009.

[48] 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.

[49] Y. He, N. Yu, M. Keuper, and M. Fritz, *Beyond the spectrum: Detecting deepfakes via re-synthesis*, in Thirtieth International Joint Conference on Artificial Intelligence, IJCAI, 2021, pp. 2534–2541.

[50] G. E. Hinton and R. R. Salakhutdinov, *Reducing the dimensionality of data with neural networks*, science, 313 (2006), pp. 504–507.

[51] S. Hu, Y. Li, and S. Lyu, *Exposing gan-generated faces using inconsistent corneal specular highlights*, in ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2021, pp. 2500–2504.

[52] Y. Huang, F. Juefei-Xu, Q. Guo, X. Xie, L. Ma, W. Miao, Y. Liu, and G. Pu, *Fakeretouch: Evading deepfakes detection via the guidance of deliberate noise*, arXiv preprint arXiv:2009.09213, (2020).

[53] 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.

[54] M. Huh, A. Liu, A. Owens, and A. A. Efros, *Fighting fake news: Image splice detection via learned self-consistency*, in ECCV, 2018, pp. 101–117.

[55] S. Hussain, P. Neekhara, M. Jere, F. Koushanfar, and J. McAuley, *Adversarial deepfakes: Evaluating vulnerability of deepfake detectors to adversarial examples*, in Proceedings of the IEEE/CVF winter conference on applications of computer vision, 2021, pp. 3348–3357.

[56] D. James, *Crafting digital media: Audacity, Blender, Drupal, GIMP, Scribus, and other open source tools*, Springer, 2009.

[57] H. Jeon, Y. Bang, and S. S. Woo, *Fdftnet: Facing off fake images using fake detection fine-tuning network*, in IFIP international conference on ICT systems security and privacy protection, Springer, 2020, pp. 416–430.

[58] Y. Jeong, D. Kim, S. Min, S. Joe, Y. Gwon, and J. Choi, *Bihpf: Bilateral high-pass filters for robust deepfake detection*, in Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2022, pp. 48–57.

[59] Y. Jeong, D. Kim, Y. Ro, and J. Choi, *Frepgan: Robust deepfake detection using frequency-level perturbations*, in Proceedings of the AAAI conference on artificial intelligence, 2022.

[60] L. JIANG, B. DAI, W. WU, AND C. C. LOY, *Focal frequency loss for image reconstruction and synthesis*, in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 13919–13929.

[61] J. JOHNSON, A. ALAHI, AND L. FEI-FEI, *Perceptual losses for real-time style transfer and super-resolution*, in European conference on computer vision, Springer, 2016, pp. 694–711.

[62] C. JON, *Experts fear face swapping tech could start an international showdown.* The Outline, January 2020. `https://theoutline.com/post/3179/deepfake-videos-are-freaking-experts-out`. Accessed: 2023-01-03.

[63] F. JUEFEI-XU, R. WANG, Y. HUANG, Q. GUO, L. MA, AND Y. LIU, *Countering malicious deepfakes: Survey, battleground, and horizon*, International Journal of Computer Vision, (2022), pp. 1–57.

[64] S. JUNG AND M. KEUPER, *Spectral distribution aware image generation*, in Proceedings of the AAAI conference on artificial intelligence, vol. 35, issue 2, 2021, pp. 1734–1742.

[65] 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).

[66] T. KARRAS, T. AILA, S. LAINE, AND J. LEHTINEN, *Progressive growing of GANs for improved quality, stability, and variation*, in ICLR, 2018.

[67] T. KARRAS, T. AILA, S. LAINE, AND J. LEHTINEN, *Progressive growing of gans for improved quality, stability, and variation*, in International Conference on Learning Representations, 2018.

[68] 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.

[69] 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.

[70] G. KAUR, N. SINGH, AND M. KUMAR, *Image forgery techniques: a review*, Artificial Intelligence Review, (2022), pp. 1–49.

[71] M. KHARRAZI, H. T. SENCAR, AND N. D. MEMON, *Blind source camera identification*, in Proceedings of the 2004 International Conference on Image Processing, ICIP 2004, Singapore, October 24-27, 2004, IEEE, 2004, pp. 709–712.

[72] D. P. KINGMA AND J. BA, *Adam: A method for stochastic optimization*, in ICLR, 2015.

[73] D. P. KINGMA AND J. BA, *Adam: A method for stochastic optimization*, in ICLR (Poster), 2015.

[74] P. KORSHUNOV AND S. MARCEL, *Deepfakes: a new threat to face recognition? assessment and detection*, arXiv preprint arXiv:1812.08685, (2018).

[75] A. KOT AND H. CAO, *Image and video source class identification*, Digital Image Forensics: There is More to a Picture than Meets the Eye, (2013), pp. 157–178.

[76] A. KRIZHEVSKY, I. SUTSKEVER, AND G. E. HINTON, *Imagenet classification with deep convolutional neural networks*, in Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States, P. L. Bartlett, F. C. N. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, eds., 2012, pp. 1106–1114.

[77] A. KURAKIN, I. J. GOODFELLOW, AND S. BENGIO, *Adversarial examples in the physical world*, in Artificial intelligence safety and security, Chapman and Hall/CRC, 2018, pp. 99–112.

[78] O. LECARME AND K. DELVARE, *The book of GIMP: A complete guide to nearly everything*, No Starch Press, 2013.

[79] Y. LECUN, Y. BENGIO, AND G. E. HINTON, *Deep learning*, Nat., 521 (2015), pp. 436–444.

[80] R. LETZTER, *Huawei used an image taken with $4,500 worth of camera gear to promote its smartphone camera.*
Business Insider, July 2016.
`https://www.businessinsider.com/how-a-fake-photo-got-used-in-a-\huawei-p9-ad-2016-7`. Accessed: 2023-01-03.

[81] C.-L. LI, W.-C. CHANG, Y. CHENG, Y. YANG, AND B. PÓCZOS, *Mmd gan: Towards deeper understanding of moment matching network*, Advances in neural information processing systems, 30 (2017).

[82] H. LI, B. LI, S. TAN, AND J. HUANG, *Identification of deep network generated images using disparities in color components*, Signal Processing, 174 (2020), p. 107616.

[83] 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.

[84] W. LI, T. ZHANG, E. ZHENG, AND X. PING, *Identifying photorealistic computer graphics using second-order difference statistics*, in Seventh International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2010, 10-12 August 2010, Yantai, Shandong, China, M. Li, Q. Liang, L. Wang, and Y. Song, eds., IEEE, 2010, pp. 2316–2319.

[85] Y. LI AND S. LYU, *Exposing deepfake videos by detecting face warping artifacts*, in IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2019.

[86] Z. LI, J. YE, AND Y. SHI, *Distinguishing computer graphics from photographic images using local binary patterns*, in Digital Forensics and Watermaking - 11th International Workshop, IWDW 2012, Shanghai, China, October 31 - November 3, 2012, Revised Selected Papers, Y. Q. Shi, H. Kim, and F. Pérez-González, eds., vol. 7809 of Lecture Notes in Computer Science, Springer, 2012, pp. 228–241.

[87] Z. LI, Z. ZHANG, AND Y. SHI, *Distinguishing computer graphics from photographic images using a multiresolution approach based on local binary patterns*, Security and Communication Networks, 7 (2014), pp. 2153–2159.

[88] Q. LIAO, Y. LI, X. WANG, B. KONG, B. ZHU, S. LYU, Y. YIN, Q. SONG, AND X. WU, *Imperceptible adversarial examples for fake image detection*, in 2021 IEEE International Conference on Image Processing (ICIP), IEEE, 2021, pp. 3912–3916.

[89] 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).

[90] C. Liu, T. Zhu, J. Zhang, and W. Zhou, *Privacy intelligence: A survey on image privacy in online social networks*, ACM Computing Surveys (CSUR), (2020).

[91] 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.

[92] M. Liu, Y. Ding, M. Xia, X. Liu, E. Ding, W. Zuo, and S. Wen, *Stgan: A unified selective transfer network for arbitrary image attribute editing*, in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 3673–3682.

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

[94] 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.

[95] J. Lukás, J. J. Fridrich, and M. Goljan, *Digital camera identification from sensor pattern noise*, IEEE Trans. Information Forensics and Security, 1 (2006), pp. 205–214.

[96] ——, *Digital camera identification from sensor pattern noise*, IEEE Trans. Inf. Forensics Secur., 1 (2006), pp. 205–214.

[97] S. Lyu and H. Farid, *How realistic is photorealistic?*, IEEE Trans. Signal Process., 53 (2005), pp. 845–850.

[98] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, *Towards deep learning models resistant to adversarial attacks*, in International Conference on Learning Representations, 2018.

[99] F. Marra, D. Gragnaniello, D. Cozzolino, and L. Verdoliva, *Detection of gan-generated fake images over social networks*, in 2018 IEEE conference on multimedia information processing and retrieval (MIPR), IEEE, 2018, pp. 384–389.

[100] 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.

[101] F. Marra, C. Saltori, G. Boato, and L. Verdoliva, *Incremental learning for the detection and classification of gan-generated images*, in 2019 IEEE international workshop on information forensics and security (WIFS), IEEE, 2019, pp. 1–6.

[102] F. Matern, C. Riess, and M. Stamminger, *Exploiting visual artifacts to expose deepfakes and face manipulations*, in 2019 IEEE Winter Applications of Computer Vision Workshops (WACVW), IEEE, 2019, pp. 83–92.

[103] S. McCloskey and M. Albright, *Detecting gan-generated imagery using saturation cues*, in 2019 IEEE international conference on image processing (ICIP), IEEE, 2019, pp. 4584–4588.

[104] N. Mike and L. Patrick J, *In an iranian image, a missile too many*.
The New York Times, July 2008.
`https://archive.nytimes.com/thelede.blogs.nytimes.com/2008/07/10/in-an-iranian-image-a-missile-too-many/`. Accessed: 2023-01-03.

[105] Y. Mirsky and W. Lee, *The creation and detection of deepfakes: A survey*, ACM Computing Surveys (CSUR), 54 (2021), pp. 1–41.

[106] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, *Spectral normalization for generative adversarial networks*, in International Conference on Learning Representations, 2018.

[107] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, *Deepfool: a simple and accurate method to fool deep neural networks*, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 2574–2582.

[108] 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.

[109] P. NEEKHARA, B. DOLHANSKY, J. BITTON, AND C. C. FERRER, *Adversarial threats to deepfake detection: A practical perspective*, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 923–932.

[110] P. NEEKHARA, S. HUSSAIN, X. ZHANG, K. HUANG, J. MCAULEY, AND F. KOUSHANFAR, *Facesigns: Semi-fragile neural watermarks for media authentication and countering deepfakes*, arXiv preprint arXiv:2204.01960, (2022).

[111] J. C. NEVES, R. TOLOSANA, R. VERA-RODRIGUEZ, V. LOPES, H. PROENÇA, AND J. FIERREZ, *Ganprintr: Improved fakes and evaluation of the state of the art in face manipulation detection*, IEEE Journal of Selected Topics in Signal Processing, 14 (2020), pp. 1038–1048.

[112] T. T. NGUYEN, Q. V. H. NGUYEN, D. T. NGUYEN, D. T. NGUYEN, T. HUYNH-THE, S. NAHAVANDI, T. T. NGUYEN, Q.-V. PHAM, AND C. M. NGUYEN, *Deep learning for deepfakes creation and detection: A survey*, Computer Vision and Image Understanding, 223 (2022), p. 103525.

[113] Y. NIRKIN, L. WOLF, Y. KELLER, AND T. HASSNER, *Deepfake detection based on discrepancies between faces and their context*, IEEE Transactions on Pattern Analysis and Machine Intelligence, (2021).

[114] F. PENG, L. YIN, AND M. LONG, *Bdc-gan: Bidirectional conversion between computer-generated and natural facial images for anti-forensics*, IEEE Transactions on Circuits and Systems for Video Technology, (2022).

[115] A. PIVA, *An overview on image forensics*, International Scholarly Research Notices, 2013 (2013).

[116] 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.

[117] M. A. Qureshi and E.-S. M. El-Alfy, *Bibliography of digital image anti-forensics and anti-anti-forensics techniques*, IET Image Processing, 13 (2019), pp. 1811–1823.

[118] N. Rahaman, A. Baratin, D. Arpit, F. Draxler, M. Lin, F. Hamprecht, Y. Bengio, and A. Courville, *On the spectral bias of neural networks*, in International Conference on Machine Learning, PMLR, 2019, pp. 5301–5310.

[119] E. R. S. D. Rezende, G. C. S. Ruppert, and T. Carvalho, *Detecting computer generated images with deep convolutional neural networks*, in 30th SIBGRAPI Conference on Graphics, Patterns and Images, SIBGRAPI 2017, Niterói, Brazil, October 17-20, 2017, IEEE Computer Society, 2017, pp. 71–78.

[120] O. Ronneberger, P. Fischer, and T. Brox, *U-net: Convolutional networks for biomedical image segmentation*, in International Conference on Medical image computing and computer-assisted intervention, Springer, 2015, pp. 234–241.

[121] K. Roose, *Here come the fake videos, too.*
The New York Times, March 2018.
`https://www.nytimes.com/2018/03/04/technology/fake-videos-deepfakes.html`. Accessed: 2023-01-03.

[122] A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Niessner, *Faceforensics++: Learning to detect manipulated facial images*, in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 1–11.

[123] S. Ruder, *An overview of gradient descent optimization algorithms*, arXiv preprint arXiv:1609.04747, (2016).

[124] L. Ruff, R. A. Vandermeulen, N. Görnitz, A. Binder, E. Müller, K.-R. Müller, and M. Kloft, *Deep semi-supervised anomaly detection*, in International Conference on Learning Representations, 2020.

[125] M. Schreyer, T. Sattarov, B. Reimer, and D. Borth, *Adversarial learning of deepfakes in accounting*, arXiv preprint arXiv:1910.03810, (2019).

[126] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, *Grad-cam: Visual explanations from deep networks via gradient-*

based localization, in Proceedings of the IEEE international conference on computer vision, 2017, pp. 618–626.

[127] R. R. SELVARAJU, M. COGSWELL, A. DAS, R. VEDANTAM, D. PARIKH, AND D. BATRA, *Grad-CAM: Visual explanations from deep networks via gradient-based localization*, Int. J. Comput. Vis., 128 (2020), pp. 336–359.

[128] K. SIMONYAN AND A. ZISSERMAN, *Very deep convolutional networks for large-scale image recognition*, arXiv preprint arXiv:1409.1556, (2014).

[129] J. T. SPRINGENBERG, A. DOSOVITSKIY, T. BROX, AND M. A. RIEDMILLER, *Striving for simplicity: The all convolutional net*, in ICLR, 2015.

[130] S. SUWAJANAKORN, S. M. SEITZ, AND I. KEMELMACHER-SHLIZERMAN, *Synthesizing obama: learning lip sync from audio*, ACM Transactions on Graphics (ToG), 36 (2017), pp. 1–13.

[131] SYNCED, *Barack obama is the benchmark for fake lip-sync videos.*
medium.com.
https://medium.com/syncedreview/barack-obama-is-the-benchmark-for-fake-lip-sy
Access date: 2023-01-03.

[132] A. SYSTEMS, *Adobe Photoshop 7.0*, Adobe Press, 2002.

[133] C. SZEGEDY, W. ZAREMBA, I. SUTSKEVER, J. BRUNA, D. ERHAN, I. GOODFELLOW, AND R. FERGUS, *Intriguing properties of neural networks*, arXiv preprint arXiv:1312.6199, (2013).

[134] S. TARIQ, S. LEE, H. KIM, Y. SHIN, AND S. S. WOO, *Detecting both machine and human created fake face images in the wild*, in Proceedings of the 2nd international workshop on multimedia privacy and security, 2018, pp. 81–87.

[135] J. THIES, M. ZOLLHOFER, M. STAMMINGER, C. THEOBALT, AND M. NIESSNER, *Face2face: Real-time face capture and reenactment of rgb videos*, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 2387–2395.

[136] T. THONGKAMWITOON, H. MUAMMAR, AND P. L. DRAGOTTI, *An image recapture detection algorithm based on learning dictionaries of edge profiles*, IEEE Trans. Information Forensics and Security, 10 (2015), pp. 953–968.

[137] R. TOLOSANA, R. VERA-RODRIGUEZ, J. FIERREZ, A. MORALES, AND J. ORTEGA-GARCIA, *Deepfakes and beyond: A survey of face manipulation and fake detection*, Information Fusion, 64 (2020), pp. 131–148.

[138] R. TOLOSANA, R. VERA-RODRÍGUEZ, J. FIÉRREZ, A. MORALES, AND J. ORTEGA-GARCIA, *Deepfakes and beyond: A survey of face manipulation and fake detection*, Inf. Fusion, 64 (2020), pp. 131–148.

[139] D. TRAN, *face2face-demo*.
GitHub.
`https://github.com/datitran/face2face-demo`. Access date: 2023-01-03.

[140] A. TUAMA, F. COMBY, AND M. CHAUMONT, *Camera model identification with the use of deep convolutional neural networks*, in IEEE International Workshop on Information Forensics and Security, WIFS 2016, Abu Dhabi, United Arab Emirates, December 4-7, 2016, IEEE, 2016, pp. 1–6.

[141] L. VAN DER MAATEN AND G. HINTON, *Visualizing data using t-sne.*, Journal of machine learning research, 9 (2008).

[142] V. A. VAN DER SCHAAF AND J. V. VAN HATEREN, *Modelling the power spectra of natural images: statistics and information*, Vision research, 36 (1996), pp. 2759–2770.

[143] S. WALIA AND K. KUMAR, *Digital image forgery detection: a systematic scrutiny*, Australian Journal of Forensic Sciences, 51 (2019), pp. 488–526.

[144] H. WANG, X. WU, Z. HUANG, AND E. P. XING, *High-frequency component helps explain the generalization of convolutional neural networks*, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 8684–8694.

[145] 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.

[146] 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.

[147] Y. WANG, X. DING, Y. YANG, L. DING, R. WARD, AND Z. J. WANG, *Perception matters: Exploring imperceptible and transferable anti-forensics for gan-generated fake face imagery detection*, Pattern Recognition Letters, 146 (2021), pp. 15–22.

[148] Y. WANG AND P. MOULIN, *On discrimination between photorealistic and photographic images*, in 2006 IEEE International Conference on Acoustics Speech and Signal Processing, ICASSP 2006, Toulouse, France, May 14-19, 2006, IEEE, 2006, pp. 161–164.

[149] M. WESTERLUND, *The emergence of deepfake technology: A review*, Technology Innovation Management Review, 9 (2019).

[150] D. H. WOLPERT AND W. G. MACREADY, *No free lunch theorems for optimization*, IEEE Trans. Evol. Comput., 1 (1997), pp. 67–82.

[151] Z.-Q. J. XU, Y. ZHANG, T. LUO, Y. XIAO, AND Z. MA, *Frequency principle: Fourier analysis sheds light on deep neural networks*, arXiv preprint arXiv:1901.06523, (2019).

[152] P. YANG, R. NI, AND Y. ZHAO, *Recapture image forensics based on laplacian convolutional neural networks*, in Digital Forensics and Watermarking - 15th International Workshop, IWDW 2016, Beijing, China, September 17-19, 2016, Revised Selected Papers, Y. Shi, H. Kim, F. Pérez-González, and F. Liu, eds., vol. 10082 of Lecture Notes in Computer Science, 2016, pp. 119–128.

[153] T. YANG, J. CAO, Q. SHENG, L. LI, J. JI, X. LI, AND S. TANG, *Learning to disentangle gan fingerprint for fake image attribution*, arXiv preprint arXiv:2106.08749, (2021).

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

[155] Y. YAO, W. HU, W. ZHANG, T. WU, AND Y. SHI, *Distinguishing computer-generated graphics from natural images based on sensor pattern noise and deep learning*, Sensors, 18 (2018), p. 1296.

[156] D. YIN, R. GONTIJO LOPES, J. SHLENS, E. D. CUBUK, AND J. GILMER, *A fourier perspective on model robustness in computer vision*, Advances in Neural Information Processing Systems, 32 (2019).

[157] F. Yu, Y. Zhang, S. Song, A. Seff, and J. Xiao, *LSUN: construction of a large-scale image dataset using deep learning with humans in the loop*, CoRR, abs/1506.03365 (2015).

[158] 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.

[159] 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.

[160] X. Zhao and M. C. Stamm, *The effect of class definitions on the transferability of adversarial attacks against forensic cnns*, Electronic Imaging, 2020 (2020), pp. 119–1.

[161] Y. Zhao, B. Liu, M. Ding, B. Liu, T. Zhu, and X. Yu, *Proactive deepfake defence via identity watermarking*, in Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2023, pp. 4602–4611.

[162] L. Zheng, Y. Zhang, and V. L. Thing, *A survey on image tampering and its detection in real-world photos*, Journal of Visual Communication and Image Representation, 58 (2019), pp. 380–399.

[163] P. Zhou, X. Han, V. I. Morariu, and L. S. Davis, *Learning rich features for image manipulation detection*, in CVPR, 2018, pp. 1053–1061.

[164] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, *Unpaired image-to-image translation using cycle-consistent adversarial networks*, in ICCV, 2017, pp. 2223–2232.