Adversarial Deep Learning on Digital Media Security and Forensics

by

Yongwei Wang

B.Sc, Northwestern Polytechnical University, 2014

M.Sc, Northwestern Polytechnical University, 2017

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Doctor of Philosophy

in

THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES
(Electrical and Computer Engineering)

The University of British Columbia
(Vancouver)

April 2021

The following individuals certify that they have read, and recommend to the Faculty of Graduate and Postdoctoral Studies for acceptance, the thesis entitled:

Adversarial Deep Learning on Digital Media Security and Forensics

submitted by **Yongwei Wang** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy** in **Electrical and Computer Engineering**.

Examining Committee:

Z. Jane Wang, Professor, Electrical and Computer Engineering, UBC
Supervisor

Rabab Ward, Professor, Electrical and Computer Engineering, UBC
Co-Supervisor

Panos Nasiopoulos, Professor, Electrical and Computer Engineering, UBC
Supervisory Committee Member

Cyril Leung, Professor, Electrical and Computer Engineering, UBC
Supervisory Committee Member

Laks V.S. Lakshmanan, Professor, Computer Science, UBC
University Examiner

Purang Abolmaesumi, Professor, Electrical and Computer Engineering, UBC
University Examiner

Additional Supervisory Committee Members:

Victor Leung, Professor, Electrical and Computer Engineering, UBC
Supervisory Committee Member

Abstract

Data-driven deep learning tasks for security related applications are gaining increasing popularity and achieving impressive performances. This thesis investigates adversarial vulnerabilities of such tasks in order to establish secure and reliable machine learning systems. Adversary attacks aim to extract private data from a model of a task and misguide the model so it yields wrong results or an answer desired by the attacker.

This thesis studies potential adversarial attacks that may affect an existing deep learning model of a specific task. Novel approaches that expose security vulnerabilities of four typical deep learning models in three dominant tasks (i.e., matching, classification and regression) are developed. These models include image hashing for image authentication and retrieval, fake face imagery forensic detection, image classification and single object tracking. In the first model, image hashing converts images into codes that are supposed to be non-invertible. However, we prove that this can pose image privacy concerns, and propose two deep learning de-hashing neural networks to show that we can obtain high quality images that are inverted from given image hashes. In the second model, we address fake face image detection. Fake images that can escape an adversarial attacked detector are usually degraded versions of original images. We analyze the visual degradation in such face images, and show how to design attacks that result in visually imperceptible adversarial images. For the image classification model, instead of the conventionally employed visual distortion metric, we propose the use of perceptual models as a novel measure for adversarial example generation. We then propose two sets of attack methods that can generally be incorporated into all existing gradient-based attacks. Lastly, for the single object tracking model, we propose the concept of

universally and physically feasible attacks on visual object tracking in real-world settings. We develop a novel attack framework and experimentally demonstrate the feasibility of the proposed concept.

The adversarial explorations and examples provided in this thesis show how existing deep learning tasks and their models could be vulnerable to malicious attacks. This would help researchers design more secure and trustworthy models for digital media security and forensics.

Lay Summary

Many problems in artificial intelligence (AI) deal with developing automated decision-making systems. Deep learning, a special type of AI, has achieved remarkable performance in many applications. Deep learning models however are sensitive to perturbations, giving rise to security, privacy and reliability issues in practical applications. The objective of this thesis is to address security and privacy threats that arise in four typical digital media problems. The four problems studied are: 1) how to reconstruct images with high perceptual quality from compact hashing signatures, thus causing privacy issues in image authentication and retrieval problems, 2) how to fool forensic detectors to classify fake face images as real images or vice versa, 3) how to deceive image classifiers to make wrong decisions and 4) how to misguide visual trackers during real-world object tracking by pasting on the object a printed sticker generated by our algorithms.

Preface

This dissertation is composed based on a collection of collaborative manuscripts. The majority of the research, including literature reviews, problem formulation, algorithm implementation, numerical analysis and manuscript writing were conducted by the candidate, under the supervision of Prof. Z. Jane Wang and Prof. Rabab Ward. The manuscripts were primarily written by the candidate, with technical suggestions and editorial feedbacks from Prof. Z. Jane Wang (Chapters 2-5) and Prof. Rabab Ward (Chapters 2-4).

Chapter 2 is based on the following manuscripts:

- Y. Wang, H. Palangi, Z. J. Wang, and H. Wang, "RevHashNet: Perceptually de-hashing real-valued image hashes for similarity retrieval," *Signal processing: Image communication*, vol 68, pp. 68-75, 2018.
- Y. Wang, R. Ward, and Z. J. Wang, "Coarse-to-fine image dehashing using deep pyramidal residual learning," *IEEE Signal Processing Letters*, vol 26, pp. 1295-1299, 2019.

The author was responsible for the algorithm development and implementation, numerical analysis and manuscript writing for the two works. Dr. Z. Jane Wang proposed the image de-hashing concept. These works were conducted with the guidance, extensive technical suggestions and editorial feedbacks from Dr. Z. Jane Wang and Dr. Rabab Ward. Dr. Hamid Palangi and Dr. Haoqian Wang provided many supports for the implementation and editorial feedbacks.

Chapter 3 is based on the following manuscript:

- 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," *Patter Recognition Letters*, vol 146, pp. 15-22, 2021.

The author was responsible for the algorithm development and implementation, numerical analysis and manuscript writing for the work. The work was conducted with the guidance, extensive technical suggestions and editorial feedbacks from Dr. Z. Jane Wang and Dr. Rabab Ward. Xin Ding helped with the perturbation analysis and provided insightful discussions. Dr. Yixin Yang provided insightful suggestions and editorial feedbacks in paper revision. Li Ding helped prepare the dataset and some numerical analysis.

Chapter 4 is based on the following manuscript:

- Y. Wang, M. Feng, R. Ward, Z. J. Wang, and L. Wang, "Perception Improvement for Free: Exploring Imperceptible Black-box Adversarial Attacks on Image Classification," Under Review, 2020.

The author was responsible for the algorithm development and implementation, numerical analysis and manuscript writing for the work. The work was conducted with the guidance, extensive technical suggestions and editorial feedbacks from Dr. Z. Jane Wang, Dr. Lanjun Wang and Dr. Rabab Ward. Mingquan Feng provided some numerical analysis and helpful discussions.

Chapter 5 is based on the following manuscript:

- L. Ding*, Y. Wang*, K. Yuan, M. Jiang, P. Wang, H. Huang, and Z. J. Wang, "Towards Universal Physical Attacks on Single Object Tracking," *Proceedings of the AAAI Conference on Artificial Intelligence*, Accepted, 2021 (* equal contribution).

The author contributed to the algorithm development, numerical analysis, physical experiments, and manuscript writing for the work. Li Ding contributed to the algorithm development, numerical analysis, physical experiments and algorithm implementations for the work. The work was conducted with the guidance, extensive technical suggestions and editorial feedbacks from Dr. Z. Jane Wang. Dr. Ping Wang and Dr. Hua Huang gave the data and provided insightful suggestions and

technical support. Kaiwen Yuan and Minyang Jiang helped prepare some physical attack experiments and editorial feedbacks.

Table of Contents

List of Tables

List of Figures

Glossary

ADL Adversarial Deep Learning

AGH Anchor Graph Hashing

AI Artificial Intelligence

ASR Attack Success Rate

AUC Area Under Curve

BP Back Propagation

BRE Binary Reconstruction Embedding

CNNS Convolutional Neural Networks

CS Compressed Sensing

DIM Diverse Inputs MIM

DL Deep Learning

DNNS Deep Neural Networks

DSH Density Sensitive Hashing

ERM Empirical Risk Minimization

EOT Expectation over Transformation

FC Fully Connected

FGSM Fast Gradient Sign Method

FSIM Feature Similarity

GAN Generative Adversarial Network

GPU Graphics Processing Unit

HOG Histogram of Oriented Gradients

ILSVRC ImageNet Large-Scale Visual Recognition Competition

IOU Intersection-over-Union

IQA Image Quality Assessment

ISOH Isotropic Hashing

JND Just Noticeable Difference

KLSH Kernelized Locality Sensitive Hashing

KSH Supervised Hashing with Kernels

LASOT Large-scale Single Object Tracking

LBP Local Binary Pattern

LPIPS Learned Perceptual Image Patch Similarity

LRR Long-Range Residue

LSH Locality Sensitive Hashing

MIM Momentum Iterative FGSM

MLP Multiple Layer Perceptron

MNIST Modified National Institute of Standards and Technology database

MOS Mean Opinion Score

MTD Maximum Textural Discrepancy

NAS Neural Architecture Search

NIQE Naturalness Image Quality Evaluator

NMF Nonnegative Matrix Factorization

PAC Probably Approximately Correct

PROGAN Progressive GAN

PSNR Peak signal-to-noise Ratio

RELU Rectified Linear Unit

RIP Restricted Isometry Property

RPN Region Proposal Network

SGD Stochastic Gradient Descent

SH Spectral Hashing

SIFT Scale-Invariant Feature Transform

SOT Single Object Tracking

SSIM Structured Similarity

SVM Support Vector Machine

TNR True Negative Rate

TPR True Positive Rate

TV Total Variation

VIF Visual Information Fidelity

Acknowledgments

The journey to my doctoral study is challenging yet stimulating and memorable to me. I would like to take this opportunity to express my sincere gratitude to those who have helped and supported me along the journey.

First and foremost, my deepest thanks go to my supervisors, Prof. Z. Jane Wang and Prof. Rabab Ward, for their wise guidance and invaluable inspirations throughout my doctoral study. I feel exceedingly grateful to them for giving me the freedom to explore various interesting research topics. And they are always available to provide insightful suggestions. Without their great supervision, I would not have been able to grow up as an independent researcher. I believe the precious qualities that I learned from them will have profound impacts in my life.

I owe a debt of gratitude to my supervisory committee members, Prof. Panos Nasiopoulos, Prof. Victor Leung and Prof. Cyril Leung for their care, efforts and professional comments for my research.

I would also like to acknowledge my collaborators in addition to my supervisors. I feel it a fortune to explore the scientific world and share interesting ideas with them. I want to express my thankfulness to Dr. Hamid Palangi, Dr. Lanjun Wang, Xin Ding, Prof. Will Welch, Dr. Xiao Jin, Li Ding, to name a few.

I am thankful to my labmates, friends at UBC, Dr. Ramy Hussein, Dr. Nandinee Haq, Dr. Jiayue Cai, Dr. Jianzhe Lin, Dr. Xinrui Cui, Tianze Yu, Yuheng Wang, Dan Wang, Kaiwen Yuan, Yifang Chen, Minyang Jiang and Mazen Nashat. I appreciate the delightful and supportive environment created by them. Many thanks go to my senior labmates, Prof. Liang Zou, Dr. Jiannan Zheng and Prof. Chen He for their professional supports in my early stage of research.

I also owe special thanks to my friends outside the lab. I thank them for their

persistent encouragement and care throughout my doctoral study, and I genuinely value the pleasant time that I spent with them.

Finally, I am forever thankful to my beloved parents and my family. I deeply thank them for their unconditional love and unwavering supports.

Chapter 1

Introduction

If you know the adversary and know yourself, you need not fear the results of a hundred battles. — Sun Tzu (500 BC)

We often rely on digital media to acquire information and perceive the world. Digital media (e.g., images and videos) have largely expanded our perspectives from diverse aspects. These media are however vulnerable to malicious falsification to deceive the users. The fact that "seeing is believing" no longer applies in this domain greatly motivates forensic researchers to develop secure, reliable and trustworthy approaches to decide on the authenticity of digital media.

Over the years, different techniques have been developed to manipulate digital media to misguide people. Conventionally, media attackers employ image/video editing software (e.g., Photoshop, Meitu) to create forged copies of the original data. Common manipulation operations range from semantic-level operations (e.g., copy-move, splicing, removal) to pixel-level processing ones (e.g., median filtering, resampling). Generally, such manipulation techniques are developed by highly skilled professionals with the aim that they bypass forensic analysts. Recently, with the advancement of Artificial Intelligence (AI), more convenient tools have been developed to falsify digital media by utilizing intelligent editing software. For example, the Washington Post reported that some spies created social accounts with AI-generated fake face imagery to connect with politicians, for malicious purposes [109, 110, 200]. Deepfake is another emerging technology that automatically alters human face images/videos by swapping the face attributes of one identity into

another individual in video clips [118, 238]. This technology could be used for fun, e.g., pretending to be an actress/actor by replacing the celebrity in a video clip with a selfie. Deepfake, however, can also be adopted to create fake news, hoax and financial fraud. This raises tremendous privacy and ethical concerns [164, 231]. Other advanced image forgery operations include image translation based on generative adversarial networks [99, 280], neural style transfer [106], and adversarial perturbations [69, 92, 229, 261]. In Fig.1.1, we visualize several examples of manipulated image/video frames using some typical digital media manipulation techniques.

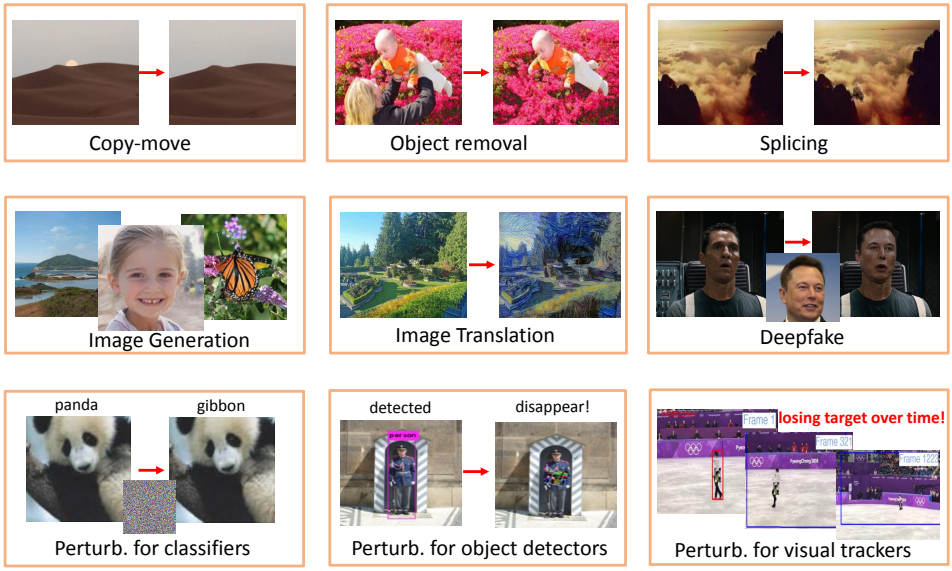


Figure 1.1: Examples of typical techniques to manipulate the digital media. The first row shows three conventional manipulation operations, copy-move [134, 255], object removal [103] and image splicing [139, 278], respectively. The second row depicts several recently emerged image/video forgery techniques which are respectively as, image generation using the generative adversarial networks [16, 66, 109, 110], pixelwise image translation [99, 106, 280], and face swapping using the Deepfake technique [118, 196, 238]. The last row visualizes some perturbation attacks targeting different deep learning models, i.e., image classifiers [46, 69, 78], object detectors [27, 92, 191, 229, 230, 257] and visual trackers [131, 252, 261, 261].

The advent of the big data era also catalyzes the evolution of favorable intelligent machine learning models. *Machine learning models help humans perceive digital media, analyze the contents and make decisions*. Some models are developed to assist forensic analysts to detect fake digital media. An example is, image hashing models are developed for image authentication and image retrieval purposes [77, 154, 219]. Diverse Deep Learning (DL) based models are carefully designed to identify the AI-generated fake face images [132, 167, 243]. In addition, researchers have explored different types of DL-based models for image recognition [78, 78, 119, 209, 225], in the presence of gigantic data from numerous categories. ResNet [78], for the first time, was reported to outperform humans in the image recognition task on the ImageNet [41] dataset. Furthermore, there exists accelerating attention developing intelligent models for high-level computer vision tasks such as object detection [145, 190, 191], image segmentation [26, 148, 194] and visual tracking [8, 81, 130, 131, 242]. Due to an increasing performance improvement, models for high-level tasks are now deployed in security-related scenarios, such as autonomous driving, intelligent surveillance and human-machine interaction.

With the superior performance of intelligent machine models in diverse digital media applications, a natural question arises, "are these models secure, reliable and trustworthy?" Or under what circumstances, will these machine models be maliciously swindled by adversaries? What can adversaries possibly infer from the deployed models? What can model designers do to counter possible adversaries? *"To know your adversary, you must become your adversary"*, said by Sun Tzu. The philosophy from the battlefield also applies to the investigation of security and reliability of machine models in the possible presence of adversaries. For one thing, we get to thoroughly know about the models by attacking them and examining their vulnerabilities. For another thing, exposure of weaknesses is crucial for algorithm designers who need to develop secure and robust machine learning models. The competing two-player game between adversaries and DL-based victim models is termed as Adversarial Deep Learning (ADL).

In the two-player adversarial game, this thesis plays the role as an "attacker". As an attacker, our overall objective is to scrutinize the vulnerabilities of three types of fundamental computer vision tasks: matching, classification and regression. For each task type, as a representative example, we target a typical security-related ma-

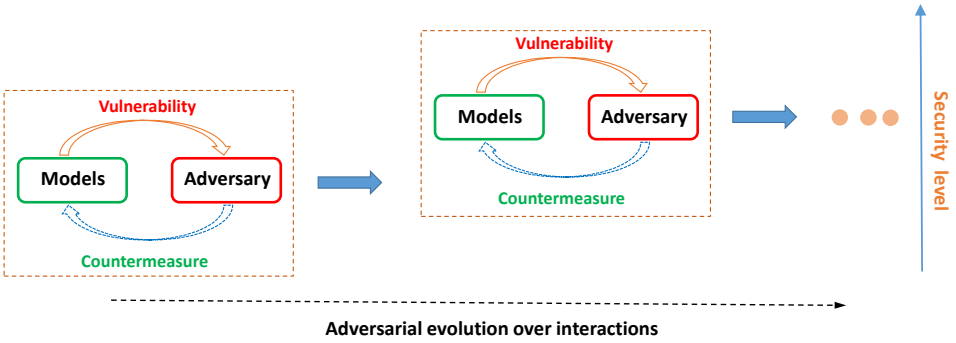


Figure 1.2: A conceptual diagram to illustrate the adversarial deep learning in progressively improving the model security during the interplay between victim models and adversaries.

chine learning model by mimicking the malicious action that attackers would do. To be more specific, this thesis develops novel attacks for four essential models belonging to three dominant tasks in digital media analysis: 1) image hashing for image retrieval and authentication, which is a typical matching task; 2) GAN-generated fake face imagery forensics, which is a representative binary classification task; 3) multi-class image classification, which is a more general multi-class classification task; and 4) single object tracking in videos, which is an essential video surveillance model involving a combination of the matching task, the classification task and the regression task. More importantly, it is worth mentioning that the proposed attacks are general methods and that they can be similarly employed (or incorporated) in investigating security threats of other related models of other tasks in matching, classification and regression.

1.1 Related work

In this section, we introduce the background knowledge and discuss existing works related to image hashing, generative adversarial networks for fake imagery generation, image classification and single object tracking in Section 1.1.1 to Section 1.1.4, respectively. In Section 1.1.5, we discuss work related to adversarial deep learning, and we summarize our research objectives, challenges and contributions in Section 1.2.

1.1.1 Image hashing

During the last decade, image hashing has been extensively studied and utilized in order to protect digital media from malicious distortions and unauthorized distributions [154, 168, 219]. Unlike conventional cryptographic hashing functions, image hashing takes a content-preserving property, namely, images that appear perceptually similar by the human vision system (HVS) will yield closer hashes, while maliciously attacked images give completely different hashes [154]. The closeness here can be readily measured using Euclidean or Hamming distances.

Image hashing is a scheme that aims to find a one-way mapping from an image to a compact hash code. There are two types of image hashing algorithms: 1) secure and robust image hashing for security purposes (e.g., image authentication, content identification etc) [154, 168, 219], and 2) image hashing methods specially designed for similarity retrieval [40, 77, 126, 143, 240]. The first type of image hashing generally consists of two steps to generate a hash: feature extraction and feature compression. Taking advantage of pseudo-randomization techniques, a secret key is incorporated in the feature compression step. In certain scenarios, another key is also added in the feature extraction step to enhance the security of image hashing algorithms. Therefore, the secure image hash behaves as a secure tag or digital fingerprint of an image.

With the advent of big data, the second type of image hashing methods are gaining increasing popularity [77]. As an early technique of this category of hashing, the Locality Sensitive Hashing (LSH) [63] exploits random projections to construct the hash functions. Without relevant information, LSH directly maps high-dimensional original images to a low dimensional hash manifold. The main drawback of LSH is its need of relatively long hash codes, as this limits its scalability to large-scale retrieval tasks. Later, the spectral hashing [250] algorithm was proposed to generate effective and compact codes leveraging machine learning techniques. To allow hashing in kernel space, much work such as Binary Reconstructive Embeddings (BRE) [120] and Supervised Hashing with Kernels (KSH) [144] were subsequently proposed. Most recent work has turned to deep learning based image hashing [21, 126, 137, 279], due to the prominent success achieved by deep learning in many vision related applications.

Some off-the-shelf image hashing methods are able to generate more compact and robust hashes for fast indexing and content-based similarity retrieval. However, security is missing in some of these image hashing methods. A shared key property by the conventional cryptographic hashing and robust and secure image hashing is that they are one-way or non-invertible functions [154, 219]. A one-way function indicates that it is computationally easy to compute the hash values given an original image and the hash function but are computationally infeasible to de-hash or to reconstruct the original images given hash values. Different from cryptographic hashing and secure and robust image hashing, the one-way function property is not explicitly addressed for similarity retrieval purposes.

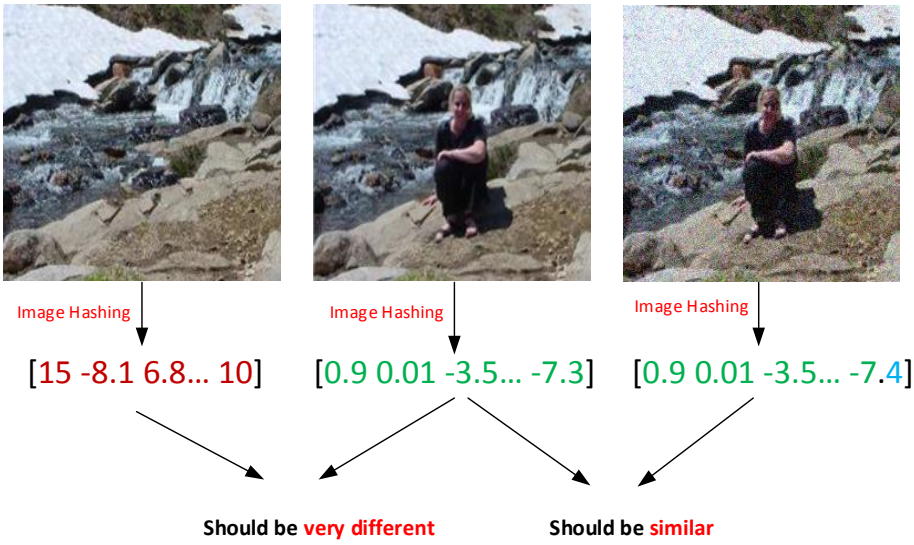


Figure 1.3: Illustration of a content-preserving property of some real-valued image hashing. The image in middle denotes the authentic image, and images in left/right represent the semantically manipulated and Guassian noise injected images on the authentic version, respectively.

In fact, this missing property could possibility result in severe security problems. Let us consider the scenario where an adversary attacks one image hash database and inverts an image which by accident contains confidential information, then information leakage becomes inevitable. Therefore non-invertibility of image hashing algorithms is of paramount importance in the context of security issues.

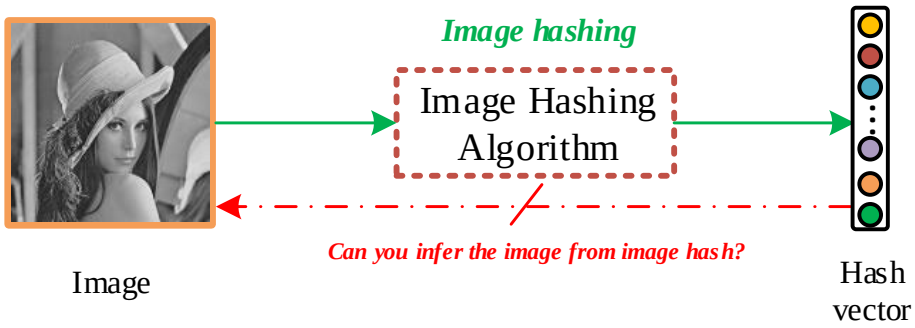


Figure 1.4: Illustration of image hashing and its inversion. Can you infer the Lena's photo from the short image hash codes?

Should there be a way to invert (or perceptually reconstruct) images to invert (or perceptually reconstruct) images from the image hashes? To our best knowledge, no work has been proposed to address the security issue in image hashing or attempt to break the one-way assumption, either implicitly or explicitly, in prevalent image hashing methods.

1.1.2 Generative adversarial networks

Generative Adversarial Network (GAN), first proposed in 2014 [66], is a type of the generative model that consists of two components: the Generator (G) and the Discriminator (D). G attempts to generate synthetic images statistically similar to the true samples to fool the discriminator. Meanwhile, the discriminator D tries to improve its discrimination capability in order to not get cheated by the generator. This two-player competing game will continue until the discriminator gives an equal probability to both the real samples and the generated fake samples.

The original GAN model [66] has limited generative capability, mainly due to the adoption of a simple neural network architecture and the vanishing gradient problem caused by its original loss function. Many GAN variants have been recently proposed to improve the original GAN's generation ability. For example, the Wasserstein GAN (WGAN-GP) with gradient penalty [72], and the Progressive GAN (PROGAN) [108] are GAN representatives that have been shown to be able to generate relatively realistic fake images. WGAN-GP adopts the Wasserstein distance as a distribution metric followed by a gradient penalty term to make the GAN

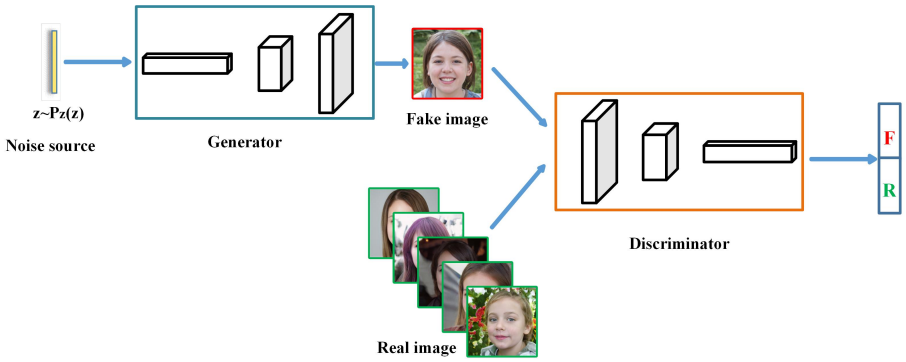


Figure 1.5: Illustration of a typical GAN model. The generator has a noise vector as its input, and tries to map the noise distribution to the real data distribution. The discriminator has the fake face image and real images as inputs and try to discriminate fake images from real ones.

training more stable. PROGAN was the first GAN model to demonstrate the capability to generate high resolution human face images. PROGAN works by progressively growing the size of the generator and discriminator. More recently, StyleGAN [109] and StyleGAN2 [110] are proposed. These models can generate human face photos with impressively realistic visual quality.

In Fig. 1.6, we depict some typical examples to illustrate recent advancements in GAN models on human face image generation [108–110, 189]. As time goes on, GAN can produce larger images with higher visual quality and more diversity. Particularly, for images from [109, 110], we can hardly discriminate the generated fake images from real ones. Therefore, given the remarkable performance of GAN, it is crucial that forensic analysts develop forensic detectors that can reliably detect GAN-generated fake face imagery from real ones.

It is worth noting that, apart from fake face imagery generation, there are plenty of GAN models available for general photo-realistic image generation on natural image datasets [16, 99, 114, 129, 280]. For example, the works in [99, 280] perform the image-to-image translation task utilizing image conditional GANs. BigGAN [16] was the first model to scale to ImageNet [41] under both the class conditional and unsupervised setting. More recently, the CcGAN [44] extended class conditional GANs to the continuous conditional setting, which was demonstrated to outperform

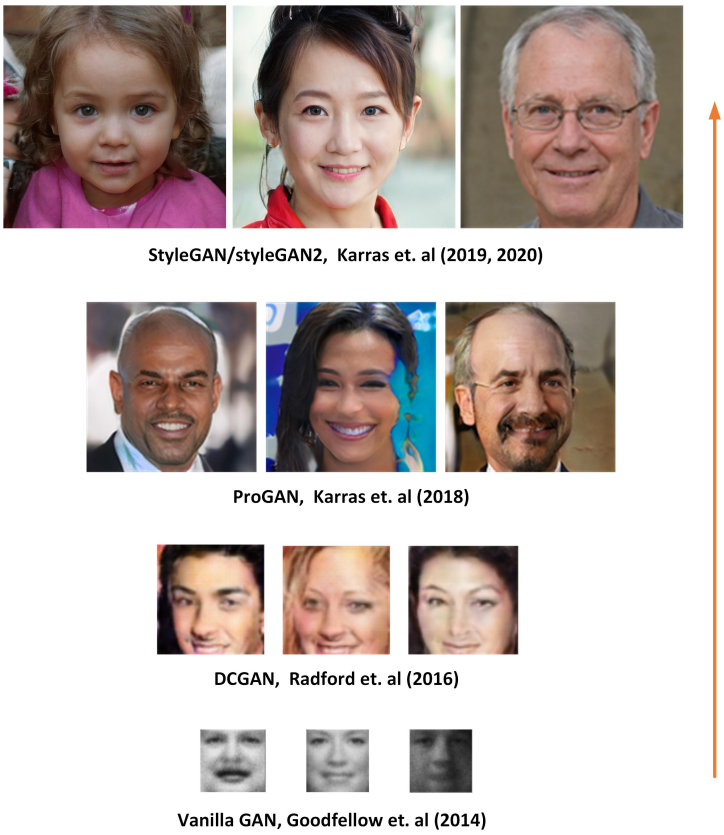


Figure 1.6: Examples of the evolution of GAN-generated fake face images from year 2014 to year 2020 [108–110, 189].

class conditional GANs both visually and quantitatively for image generation with regression labels as the condition. Despite the superior performance of such GANs, we particularly focus on the face imagery detection task because it poses a high security threat to practical applications.

1.1.3 Deep neural networks on image recognition

Deep Neural Networks (DNNS) originates from the original neural networks which were designed to model the complicated mapping between the input-output pair in the 1950s. The single-layer Perceptron is the first type of neural networks [195]. It was proposed to solve the binary classification problem given continuous-valued

inputs. The Perceptron, however, can only deal with the linear classification problem due to its limited model complexity. The Multiple Layer Perceptron (MLP) was then proposed to increase the learning capacity [197] of the network. An MLP generally consists of three layers: an input layer, an intermediate layer (hidden layer), and an output layer. In MLP, the smallest unit is called a node, each of which (except for the input nodes) is a neuron that applies a non-linear activation function (e.g., sigmoid or tanh) to enhance the nonlinear approximation capability. Although a single hidden layer is sufficient for the universal approximation, an increment in the number of hidden layers would increase its performance [87]. As for the learning method, the gradient descent was often utilized to train the neural networks with the gradient calculated by the Back Propagation (BP) method [197]. However, there are mainly two drawbacks training MLP with the BP method. Firstly, training MLP with multiple hidden layers can be very slow. Also, the solution may get stuck in some local minima. Since the 1990s, neural networks started to stagnate due to unscalability and computational issues. In 2006, Hinton et al. proposed the layer-by-layer pretraining strategy to make it possible to train neural networks that have deeper layers [82, 84].

A special research branch of MLP is the Convolutional Neural Networks (CNNS). CNNS are variants of MLP with biological inspirations from the visual cortex. In 1959, two neurophysiologists, Hubel and Wiesel [93–95], performed experiments on animals' visual cortex and revealed the mechanism of the visual system. The experiments showed that, the visual cortex contains small regions of neurons that are sensitive to specific patterns of the visual field [94]. The pattern detection process is locally invariant to the spatial location of patterns. For example, some neurons become active immediately if exposed to vertical edges, while others respond when presented with curves or horizontal or diagonal edges regardless of the exact position of these patterns in the brain. Different regions of functional neurons were organized together to form the visual perception from simple stimulus features (e.g., edges or curves) in a complex manner. The fact that specialized regions of neurons performing specific tasks has greatly inspired machine learning scientists to design the primary versions of CNNS [60, 127, 201].

CNNS were developed with two distinctly different characteristics from MLP, i.e., the sparse connectivity and the weight sharing [127]. In CNNS, neurons in

adjacent convolutional layers of a CNN network are sparsely connected by enforcing a local connectivity constraint. Sparse connectivity exploits the spatially-local correlation property, leading to a significant reduction in the number of weights. Inspired by the locality invariant property of neurons, weights are shared during the convolution process to detect corresponding patterns. The convolutional kernels are called filters, and the convolutional ouputs are named feature maps. Weight sharing further increases the learning efficiency of CNNS by reducing the number of trainable parameters. Due to the limited data and insufficient computational resources, early CNNS were only utilized in relatively simple tasks, e.g., handwritten digits recognition [127].

The first decade of the present millennium witnessed progress in computational hardware, i.e., Graphics Processing Unit (GPU), and the availability of increasingly more data from the Internet. These two factors have largely accelerated the advent of the deep learning era. In 2012, Krizhevsky et al. won the ImageNet Large-Scale Visual Recognition Competition (ILSVRC), demonstrating the superiority of CNNS by surpassing the handcrafted-feature based classification approaches by a large margin [41, 119]. Since then, different CNNS have been extensively studied and employed in numerous applications, e.g., image classification [78, 119, 209, 224, 266], image segmentation [26, 148, 194], compressed sensing [123], image super-resolution [45, 106], image hashing [21, 126, 137, 279] and image forensics [6, 25, 267]. Enhancing the field of image processing, CNNS also found successful applications in video, audio, speech and natural language processing [68, 128].

We focus on the recent development of CNNS in image recognition, a fundamental problem in computer vision. The pipeline of a basic CNN-based classifier is illustrated in Fig. 1.7 which consists of a spectrum of convolutional layers and fully-connected layers. Based on the network architectures, the exploration in CNNS of image classification can be generally categorized into two time phases. The first phase gives rise to some innovated architectural components, enabling networks deeper in depth to improve the classification accuracy. By contrast, the other trend resorts to developing more lightweight and computationally efficient architectures, preparing for model deployment in practice.

The first phase of CNN-based image classifiers: Deeper networks for improved image recognition accuracy. ***AlexNet*** is the first CNN-based image classifier

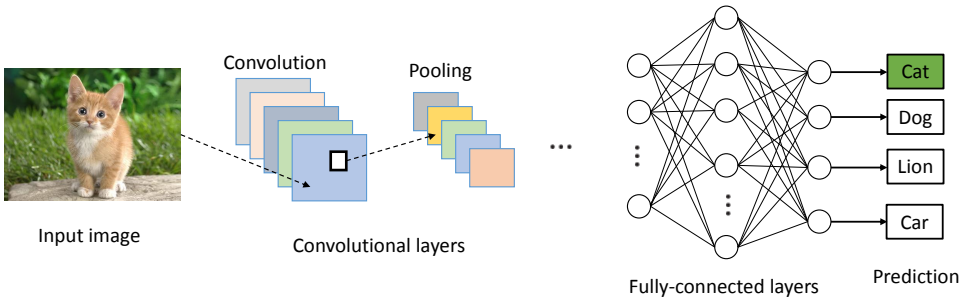
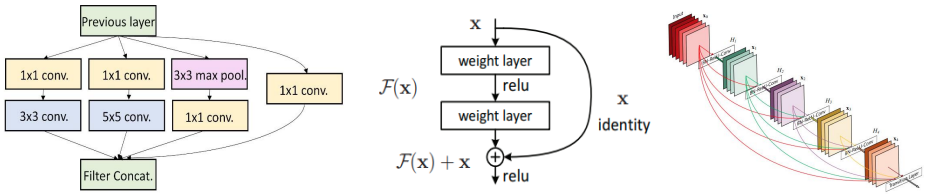


Figure 1.7: The pipeline of a basic CNN-based image classifier.

that adopts multiple GPUs to significantly increase the learning speed [119]. It also proposes the use of the Rectified Linear Units (ReLU) as the non-linear activation function to alleviate possible vanishing gradient effects in deep networks. Further, AlexNet employs "Dropout" as a regularizer to reduce the overfitting problem caused by the fully-connected neurons. ***VGG*** investigated the depth of convolutional layers on network performance [209]. This architecture utilizes 3×3 uniform-sized kernel filters to simplify the architectural design and reduce the number of parameters given the same network depth. This network configuration makes it possible to utilize much deeper layers to improve the accuracy over AlexNet. ***GoogLeNet*** proposes the *Inception* module (Fig. 1.8(a)) that allows for simultaneously increasing the depth and width of neural networks. Instead of using uniform 3×3 kernel filters, the Inception module also adopts 1×1 and 5×5 filters to capture receptive field at varied scales. Some well-known variations of GoogLeNet include Inception-v2, Inception-V3 and Inception-V4 [225, 226]. Deeper networks tend to yield improved classification accuracy; however, very deep networks are considerably challenging to optimize mainly due to the vanishing gradient issue. ***ResNet*** proposes the idea of residual learning to largely resolve the vanishing gradient problem by introducing "shortcut" connections for identity mapping. The *Residual* blocks (Fig. 1.8(b)) enable ResNet to increase its network depth to be as deep as 152 layers, and eventually enable ResNet to outperform humans in recognition accuracy on the ImageNet dataset. As a futher extension of ResNet, ***DenseNet*** was proposed to further alleviate the vanishing gradient problem, meanwhile facilitating the feature propagation and encouraging feature reuse [90]. In DenseNet, the *Dense* module (Fig. 1.8(c))

passes and reuses features from preceding layers to all subsequent layers via concatenation in a densely-connected manner. DenseNet achieves higher accuracy yet it requires less computations than its prior-art classification networks. Table 1.1 shows a more detailed performance comparison of these representative network architectures in the first research phase of CNN-based image classifiers.



(a) Inception module (b) Residual module (c) Dense module

Figure 1.8: Illustration of three typical building blocks in modern CNNS: the Inception module [224], the Residual module [78] and the Densely-connected module [90].

Table 1.1: Comparison of several representative deep CNN models on image classification. The accuracy of AlexNet is from [119], and the results of other models [78, 90, 209, 224, 225] are obtained from torchvision in PyTorch [184]. Results may fluctuate depending on hardware (e.g., initialization) and training strategies (e.g., batchsize, epoch, optimizers etc).

models	AlexNet	VGG-19	GoogLeNet	Inception-v3	ResNet-50	ResNet-101	ResNet-152	DenseNet-121	DenseNet-161
year	2012	2014	2014	2015	2015	2015	2015	2016	2016
Top 1 acc. (%)	63.3	72.38	69.78	77.45	76.15	77.37	78.31	74.65	77.65
Top 5 acc. (%)	84.6	90.88	89.53	93.56	92.87	93.56	94.06	92.17	93.80
#parameters (M)	60	144	4	24	25.6	44.5	60.2	8	28.7

The second phase of CNN-based image classifiers: Lightweight networks for more efficient deployment. Lightweight architectures are more attractive in mobile computing due to stricter requirements on the computational power, inference speed and portability in practical embedded environments. ***SqueezeNet*** proposes the *Fire* module to "squeeze" the number of parameters [96]. The Fire module consists of a "squeeze" layer and an "expand" layer. The "squeeze" layer aims to reduce the number of feature maps from a preceding layer by adopting the 1×1 convolutions. The feature maps are then expanded in channel number with 1×1 and 3×3 convolutions before they are fused via concatenation operations. SqueezeNet achieves a compa-

rable accuracy with AlexNet, yet with $\sim 50\times$ less parameters. ***MobileNet*** replaces the regular convolutional operations with depthwise separable convolution, a special type of group convolution [88]. The depthwise convolution can largely reduce the number of parameters given the same feature maps . The comparison of regular convolution and group convolution has been illustrated in Fig. 1.9. MobileNet-V2 [199], the updated version of MobileNet, proposes the *Inverted Residual Blocks* to enhance the memory efficiency (because of the directed acyclic computational graph). A concurrent work to MobileNet is ***ShuffleNet*** [274], which exploits the 1×1 group convolution with the channel-shuffle operation to reduce the computational complexity while maintaining accuracy. Apart from the manually-designed architectures, ***EfficientNet*** employs the Neural Architecture Search (NAS) technique [140] to automatically develop more efficient and more portable CNN classifiers [227]. To reduce the search space, EfficientNet maximizes the network accuracy while imposing constraints to the memory and FLOPs (i.e., multiply-adds operations), network depth, width and resolution. In Table 1.2, we list the performance comparison of seversal representative lightweight CNN models in detail.

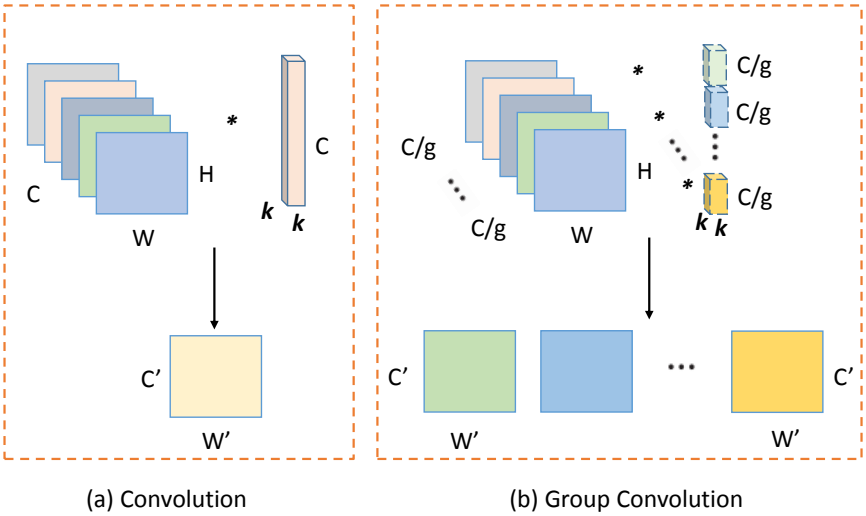
Table 1.2: Comparison of several representative lightweight CNN models on image classification [96, 199, 227, 274].

models	SqueezeNet	ShuffleNet	MobileNet-v2	EfficientNet-B0	EfficientNet-B2
year	2016	2017	2018	2019	2019
Top 1 acc. (%)	57.5	73.7	71.8	76.3	80.1
Top 5 acc. (%)	80.3	-	91.0	93.2	94.9
#parameters (M)	4.8	5.4	3.4	5.3	9.2
FLOPs (M)	833	524	300	390	1000

Given the superior performance of DL-based classifiers, before the real-world deployment, we may have natural questions including "Are these models trustworthy?", or "Under what circumstances, will the models fail?"

1.1.4 Single object tracking

Single Object Tracking (SOT) is one of the fundamental problems in computer vision, and it has attracted increasing attention in security-related applications, e.g., autonomous driving, intelligent surveillance and human-machine interaction [58, 159, 254]. SOT is the process of identifying the correspondence between an



(a) Convolution (b) Group Convolution

Figure 1.9: Comparison of the regular convolutional operation and group convolutional operations in subfigures (a) and (b), respectively. Here "$*$" denotes the convolutional operation. The regular convolution produces one feature map from C channels. By contrast, the group convolution separates kernel filters into g ($g = 1, \cdots, C$) groups and perform convolutions individually to yield g feature maps. In both operations, the number of parameters equals k^2C, and the FLOPs equal $k^2CW'H'$; however, group convolution produces $g\times$ more feature maps. Specially, group convolution degenerates to the regular convolution when $g = 1$, and it becomes depthwise (DW) convolution when $g = C$ (and DW convolution becomes the depthwise separable convolution if additionally followed by 1×1 convolutions).

arbitrary target object in the first frame and that object in subsequent frames without prior knowledge of target object categories.

SOT methods can be broadly classified into two categories: traditional handcrafted feature-based tracking methods and DL-based ones. Traditional methods focus on four building blocks: the representation scheme, the search scheme, model update and context fusion [254]. Such type of methods generally do not require high computational cost and they can achieve high tracking speed even on CPU devices. However, traditional tracking methods cannot produce satisfactory tracking performances mainly due to the employment of handcrafted features [37, 149]. By

contrast, deep neural networks can extract hierarchical level of features which have superior representation capability than that of handcrafted ones. Therefore, recent tracking works generally focus on developing more advanced DL-based approaches [8, 130, 131, 242]. In Fig. 1.10, we illustrate the pipeline of DL-based single object tracking methods.

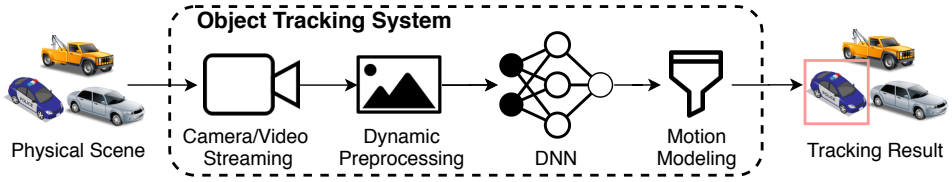


Figure 1.10: Illustration of the pipeline of DL-based single object tracking system.

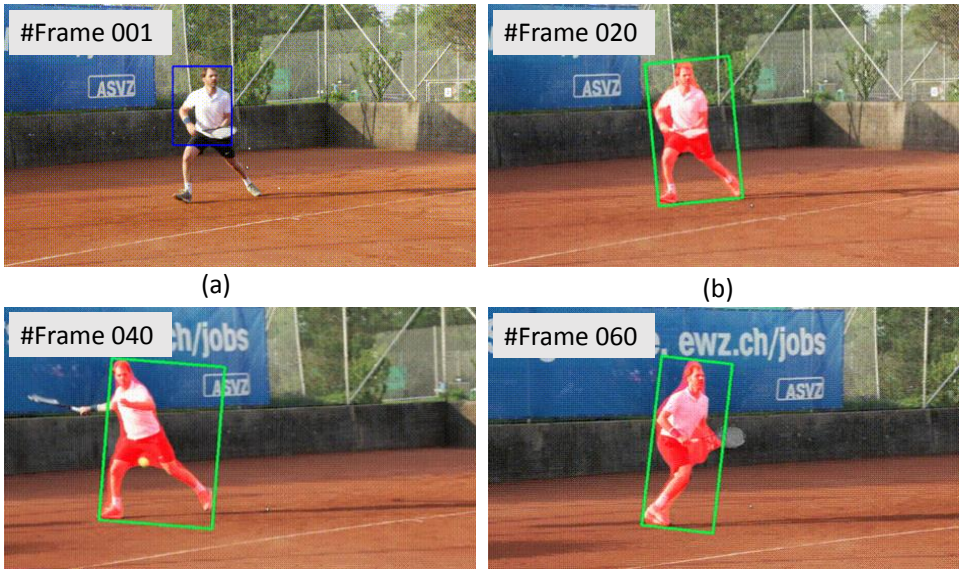


Figure 1.11: Example tracking results in single object tracking. Given the location of an object in the first frame, i.e., subfigure (a) in the blue box, the SiamMask tracker [241] accurately tracks the target over time, i.e., subfigures (b)-(d) in green boxes.

In DL-based tracking methods, mainstream explorations include: development

of deeper backbone networks, designing novel network objective functions and proposing new learning objectives. Among DL-based methods, the Siamese-based family has become the predominant research direction, mainly because of its good balance between the tracking performance and the tracking speed. In general, the Siamese networks have their inputs as a template (i.e., target object in the first frame) and search frames (i.e., subsequent video frames), and their output is a probability map indicating the location of the target in corresponding search images. The target position is estimated by computing the maximum probability from the probability map. Since the seminar work in [8] which is based on a fully-connected Siamese architecture (SiamFC), different representative methods [130, 131, 242] have been proposed in recent years. In Fig. 1.11, we show an example of tracking results from a State-of-the-art (SOTA) Siamese-based tracking method [241] that tracks the target object with satisfactory performance. The high precision and success performances achieved by SOTA trackers motivated us to investigate the security and reliability issue of existing visual tracking models.

1.1.5 Adversarial deep learning

In this section, I will firstly introduce the Probably Approximately Correct (PAC) learning, a foundation of machine learning theory, then summarize the characteristics of threat modeling. Finally, I will give an overview of several typical research directions within the general framework of adversarial deep learning in digital media security and forensics.

PAC learning

The task of machine learning is to design intelligent approaches for automatic decision making given a large volume of data [14, 172, 203]. A machine learning system generally consists of two components: the data and the model. As shown in Fig. 1.12, a machine learning system can also be divided into the *training phase* and the *inference phase* based on timing. In the training phase, a learning algorithm aims to select an optimal model, based on observed data samples, and the loss function from the hypothesis space. Assuming that training/test samples follow an independent and identical distribution (i.i.d.), a well-learned model is expected to

have similar performance in the inference phase with unseen test samples.

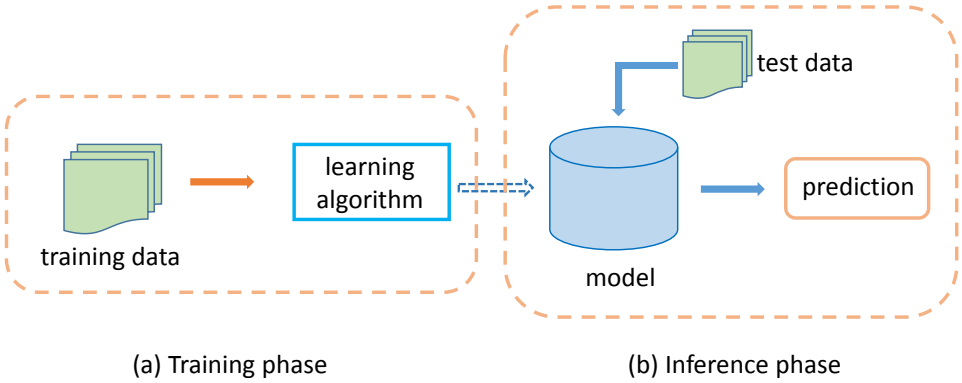


Figure 1.12: A conceptual flowchart of machine learning system in the training phase and test phase in subfigures (a) and (b), respectively.

The PAC learning is a foundation of the machine learning models, which describes the probability of learning a model from the hypothesis space with quantified error bounds [237]. In a supervised setting, assume that data samples (x, y) are independently and identically sampled from the data space $\mathcal{D}$, i.e., $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$, where x and y denote the data and associated label, respectively. Assuming there exists a mapping function $f : \mathcal{X} \mapsto \mathcal{Y}$, and a learner defines a loss function $l_f : \mathcal{X} \times \mathcal{Y} \mapsto \mathbb{R}$, based on some criteria. The goal of a learner is to capture an f^* which achieves the minimal theoretical risk $r(f, z) \triangleq \mathbb{E}_{x,y\sim\mathcal{D}}\Big[l_f(x, y)\Big]$, i.e., $f^* \in \arg\min_{f\in\mathcal{F}} r(f, z)$ where $z = (x, y)$.

In general, f^* cannot be obtained since the data distribution is usually unknown, and the learner only has finite training data observations $\vec{z} = \{\vec{z}_i\}_{i=1}^n$ where $\vec{z}_i = (x_i, y_i)$. In practice, the learner approximates f^* by adopting the Empirical Risk Minimization (ERM) formulation $\widetilde{r}(f, \vec{z}) \triangleq \frac{1}{n}\sum_{i=1}^n l_f(x_i, y_i)$, i.e., $\widetilde{f} \in \arg\min_{f\in\mathcal{F}} \widetilde{r}(f, \vec{z})$. Under these assumptions, PAC learning states that: For arbitrary data space $\mathcal{D}$, hypothesis space $\mathcal{F}$, ϵ, $\delta \in (0, 1)$, then with probability at least $1-\delta$,

$$P_r\left(\left|r(f^*) - \widetilde{r}(\widetilde{f})\right| \leqslant \epsilon\right) \geqslant 1 - \delta$$

where $r(f^*) \triangleq \arg\min_{f \in \mathcal{F}} r(f, z)$, and $\widetilde{r}(\widetilde{f}) \triangleq \arg\min_{f \in \mathcal{F}} \widetilde{r}(\widetilde{f}, \vec{z})$. Two pre-conditions to hold the PAC guarantees are: a proper ERM algorithm and uniform convergence. The first pre-condition states that the empirical risk should be close to $\widetilde{f}$ with high probability. This condition is often assumed to exist in statistical machine learning. The second pre-condition addresses that, for $\forall f \in \mathcal{F}$, the difference between empirical risk and theoretical risk should be close with high probability. This pre-condition can be satisfied by a given adequate data complexity, i.e., $m(\epsilon, \delta) = poly(\frac{1}{\epsilon}, \frac{1}{\delta})$.

To build a machine learning system, the ERM step is conducted in the *training phase* (see Fig. 1.12(a)), and PAC provides the generalization bound for the *inference phase* (see Fig. 1.12(b)). Please note that the generalization bound relies on the assumption that ***the training data and test data are independently and identically sampled from the same unknown data space*** $\mathcal{D}$. This assumption, however, is often exploited by adversaries to intentionally fool machine learning models. For example, an adversary may maliciously perturb some test samples such that the test samples no longer follow the i.i.d. assumption with the training data, which could give significantly high prediction error. Moreover, the generalization bound is given from the statistical perspective where the model pays little attention to the low-probability regions. For adversaries, however, they can always utilize data samples from such low-probability region (unseen in training) to find samples to deceive models (though low-probability region has low impacts on the generalization bound) [65, 181, 210]. These underlying assumption in PAC learning gives chances for the adversarial machine/deep learning in the presence of adversaries.

Threat modeling

Attacks to security-related models naturally arise, and the PAC theory allows the existence of such malicious manipulation. Threat modeling, originated in cyber security. It is a structured process by which potential threats or vulnerabilities can be identified and compromised beforehand. Threat modeling helps model designers to answer questions such as, "*Is the model secure*?", "*What are possible vulnerabilities*?", "*What can adversaries infer from the model*?", and "*What should I do to countermeasure such threats*?"

Inspired by works [12, 13, 24, 38, 91, 142, 181, 239, 265], we summarize threat

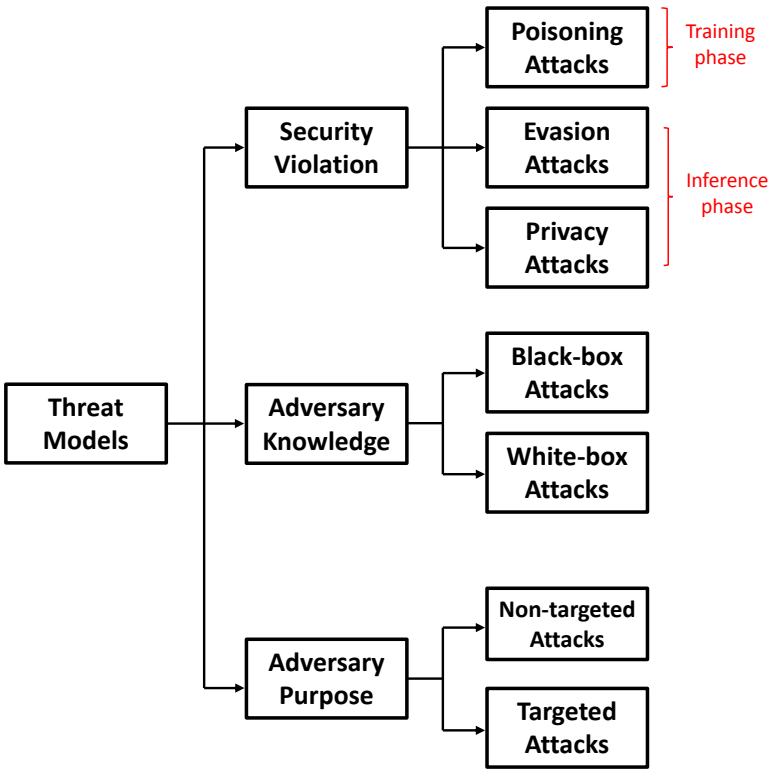


Figure 1.13: Categories of threat models from different perspectives.

models and categorize them from different perspectives, as in Fig. 1.13. These categories and taxonomies are described individually as follows.

- **Security violation.** For the security violation, threat models can be categorized into three types: poisoning attacks, evasion attacks and privacy attacks, wherein poisoning attacks take place in the *training phase* (see Fig. 1.12(a)) while evasion attacks and privacy attacks often take place during the *inference phase* (see Fig. 1.12(a)).

  Poisoning attacks (a.k.a causative attacks) target the learning algorithm by manipulating the training data in the training phase [11, 13, 100, 142, 181]. This type of attacks may happen in scenarios where malicious users can conveniently interact with models. For example, in graph classification on social networks or recommendation systems, certain nodes can be manipulated by

adversaries [218, 281]. Such attacks may also exist in federated learning systems in which a non-colluding agent can conduct data or model poisoning attacks [9, 232].

Evasion attacks often target security-critical deployed systems. In evasion attacks, adversaries do not alter the training process, rather they attempt to manipulate the test samples to mislead the deployed models. For example, in spam filtering, spam emails often fool spam filters by inserting "non-spam" words or characters to evade being detected [38, 91]. Evasion attacks also frequently take place in applications such as malware detection [71, 217], image classification [46, 69], object detection [27, 230], visual tracking [76, 259], forensic tasks [5, 177] etc. The challenge in developing evasion attacks is how to create perturbation to successfully evade detection and preserve unnoticeability (e.g., inserting texts with semantic-preserving words or perturbing images with imperceptible patterns) simultaneously.

Privacy attacks aim at uncovering information from a deployed model [161, 193, 208, 275]. Such information was not supposed to be shared, e.g., revealing knowledge about the private training data or replicating a copyright-protected model. In general, there are four categories of privacy attacks: model extraction attacks, model inversion attacks, membership inference attacks and property inference attacks. *Model extraction* attacks intend to replicate a substitute model from the target model. *Model inversion* attacks attempt to reconstruct sensitive data or attributes based on the correlation between model outputs and the training data. The process is titled *membership inference* when an adversary tries to determine whether or not a given record is from the training set of a target model. *Property inference* attacks try to infer properties of datasets which are not directly related to the learning task, e.g., inferring ratios of people wearing glasses from a training dataset for gender classification.

- **Adversary knowledge.** Based on the knowledge degree of adversaries to a target model, threat modeling can be broadly categorized into black-box attacks and white-box attacks. In black-box attacks, adversaries have no knowledge of model parameters. Rather, adversaries are permitted to access

model outputs (e.g., predicted labels or confidence scores in classification). In general, an adversary conducts black-box attacks utilizing queries [30, 48, 97] or based on the adversarial transferability property from substitute models [46, 147, 169]. In white-box attacks, an adversary is assumed to have full knowledge of the target model, i.e., training dataset, hyperparameters, model architecture and parameters. In general, an adversary achieves higher attack confidence with increasing knowledge of a target model. Yet both black-box and white-box attacks are important in security evaluations despite having varying difficulty levels.

- **Adversary purpose.** Depending on the purpose of adversaries in manipulating a target model, threat models can be divided into non-targeted attacks and targeted attacks. In *non-targeted attacks*, an adversary can mislead a target to make arbitrary decisions, but not the original one. Such attacks may happen in dodging attacks on facial biometric systems where an adversary can pretend to be any other arbitrary persons to evade detection [48, 204]. Other potential scenarios on non-targeted attacks such as multi-class image classification [46, 69], object detection [27, 92] and visual tracking [259, 261]. In *targeted attacks*, an adversary attempts to deceive the target model to produce a desired prediction. For example, in face verification, an attacker may impersonate the security system by having a face incorrectly recognized as that from a designated identity [13, 48]. Generally, compared with targeted attacks, non-targeted attacks are easier for an adversary to perform, since adversaries often have a larger search space from which to find a valid attack than in targeted situations.

Typical ADL examples

After having described learning-based models and summarized threat models, in this section, I will introduce several typical examples that fall under the umbrella of adversarial deep learning.

With the resurgence of machine learning, particularly neural networks-based DL methods, "adversarial" related terminology emerged and gained increasing attention. In 2013, Szegedy et al. designed perturbations injected in natural images

Table 1.3: Several typical examples of adversarial deep learning in digital media security and forensics. "*" denotes that some of the works are model-driven and some are data-driven.

model description	violation type	data modality	model role	adversary role	literature
watermarking	evasion*	image, video	robust watermark embedding	watermark removal	[1] [173] [220]
steganography	evasion*	image	statistical undetectability	secret message exposure	[4] [89] [59]
spam filering	evasion	text	spam detection	evading detection	[38] [91] [36]
JPEG forensics	evasion*	image	JPEG detection	evading detection	[213] [187] [153]
resampling forensics	evasion*	image	resampling detection	evading detection	[17] [185] [7]
median filtering forensics	evasion*	image	median filtering detection	evading detection	[25] [114] [256]
face verification	poisoning, evasion, privacy	image	face recognition	dodging, impersonation, privacy leakage	[204] [48] [11]
image classification	evasion	image	high accuracy	reducing accuracy	[69] [46] [124]
object detection	evasion	image	object localization	missing localization	[27] [230] [276]
visual tracking	evasion	video	object tracking	missing tracking	[259] [261] [76]
reinforcement learning	evasion	sequential data	correct policy	wrong policy	[20] [64] [28]
graph classification	evasion, poisoning	structured data	high accuracy	reducing accuracy	[218] [281] [146]

which can fool DL-based image classifiers with high success rates. Such manipulated images are named as "adversarial examples" [223]. Concurrently, Biggio et al. independently proposed gradient-based evasion attacks against classifiers based on Support Vector Machine (SVM) and neural networks [10, 12]. In the machine learning field, the first work proposing the adversarial concept dates back to 2004 when Dalvi et al. introduced an *adversary* to spam detection, termed as "adversarial classification" [38].

Indeed, the adversarial idea has existed before the pioneering work above in several security-critical disciplines, such as electronic countermeasures [15], robust and optimal control [277], cryptography [107], and data hiding [59, 220]. For example, Servetto et al. computed the watermark capacity by optimizing a minimax two-player game between a signal (i.e., watermark) and a jammer (i.e., adversarial noise) [202]. Katzenbeisser et al. defined the security of steganography from the complexity-theoretic perspective, and established the stegosystem's security as a probabilistic game between a judge and an adversary to decide whether an object is a plain cover or a stego-object [59, 86, 112]. Stamm et al. proposed the JPEG anti-forensic idea by perturbing JPEG images with carefully designed noises to hide JPEG compression artefacts and confuse forensic detectors [214]. Although such earlier works were mostly model-driven, the success of data-driven approaches also fosters the extension of the original adversarial ideas to related topics. Table 1.3 shows a literature study of typical ADL topics on digital media security and forensics.

1.2 Research objectives, challenges and contributions

In the last section (i.e., Section 1.1.5), I have introduced the concept and characteristics of adversarial deep learning. Then I surveyed several typical research examples which fall into the general framework of adversarial deep learning on digital media security and forensics. Following previous sections, firstly I will introduce research objectives in this thesis, then I will discuss challenges and the roadmap to my research topics, and finally close this chapter with a summary of the novelty and methodology contributions of each exploration separately.

Under the paradigm of adversarial deep learning, as an attacker, the major objective of this thesis is to explore and propose novel approaches to scrutinize

potential vulnerabilities of machine/deep learning models in digital media security and forensics. Specifically, this thesis studies adversarial vulnerabilities of four typical security-critical models from three dominant computer vision tasks for digital media analysis. The adversarial vulnerability of each victim model is examined under a proper threat modeling. The roadmap of this thesis is depicted in Fig. 1.14, where key contents and methodology contributions are briefly summarized.

First, Chapter 2 considers the matching task type. The victim model is image hashing models (i.e. a typical matching task) for image retrieval and authentication. Image hashing models are supposed to be non-invertible. In this topic, we perform model inversion attacks under the white-box attack assumption. Suppose that an attacker can obtain a set of image and image hash pairs (or the attacker can generate such pairs based on the victim image hashing model), the attacker's objective is to perceptually invert images from their image hashes. Our study explores the feasibility of the image hashing inversion, and fills this gap by firstly proposing a deep learning based framework, titled RevHashNet. Given real-valued image hashes generated by certain image hashing methods, the proposed RevHashNet can automatically reconstruct images that are similar to the original ones and have high visual quality. Experiments and simulations on real image datasets support the de-hashing effectiveness of the proposed RevHashNet. Despite the success of RevHashNet, the perceptual quality of dehashed images is challenged when real-valued hashes are quantized using a limited number of bits (i.e. shorter image hash codes). Besides, the scalability to larger or to color image dehashing is limited in the RevHashNet dehashing network. To address such concerns, we then propose a Pyramidal Long-Range Residual-learning Network (PyLRR-Net). PyLRR-Net is a pyramidal image reconstruction network that dehashes images in a progressive manner. At each image scale, we design and insert a Long-Range Residual (LRR) block to refine the coarse image reconstruction by leveraging deep residual learning. Experiments on both grayscale and color image datasets show that the proposed PyLRR-Net outperforms RevHashNet in terms of image dehashing quality, scalability and flexibility for large and color image dehashing problems.

Next, Chapters 3 and 4 consider the classification task type. This thesis studies the evasion vulnerability of image classification tasks in the black-box setting, where we examine two typical classification model examples: GAN-generated

fake face imagery forensic models (to represent the category of binary classifiers) and general image classification models (to represent the category of multi-class classifiers) in Chapter 3 and Chapter 4, respectively. On the classification task, we demonstrate that such models are prone to making wrong classification decisions when we inject carefully designed imperceptible perturbations to the input images. More specifically, in Chapter 3, we investigate more imperceptible and transferable attacks for GAN-generated fake face imagery forensic models, i.e., perturbing GAN-generated fake face images such that forensic detectors would mis-classify them as real ones, or vice versa. Since facial and background regions are often smooth, even small perturbations could cause noticeable perceptual impairment in fake face images. Therefore it makes existing adversarial attacks ineffective as an anti-forensic method. We analyze the perturbation residues from existing attacks. This perturbation analysis reveals the intuitive reason of the perceptual degradation issue resulting from applying existing attacks. We then propose a novel adversarial attack method, better suitable for image anti-forensics, in the transformed color domain, by considering visual perception. Simple yet effective, the proposed method can fool both deep learning and non-deep learning based forensic detectors, achieving higher attack success rate and significantly improved visual quality. Specially, we have shown that when adversaries consider imperceptibility as a constraint, the proposed anti-forensic method can improve the average attack success rate over two baseline attacks on two benchmark datasets by around 30% on fake face images. More imperceptible and more transferable, the proposed method raises new security concerns to GAN-generated fake face imagery detection.

Following the evasion attack on binary classifiers in Chapter 3, Chapter 4 studies evasion attacks on general multi-class image classifiers in the non-targeted setting. The objective is to fool such classifiers by perturbing images while keeping high visual quality. E.g., a cat image will be wrongly classified as a class label from some other classes (e.g. a panda) after perturbation. However, keeping successful adversarial perturbations imperceptible is especially challenging in black-box adversarial attacks. Often such adversarial examples can be easily spotted due to their unpleasantly poor visual quality. To improve the image quality of such attacks perceptually, we propose structure-aware adversarial attacks by generating adversarial images based on psychological perceptual models. Specifically, we

allow higher perturbations in perceptually insignificant regions, while assigning lower or no perturbation on visually sensitive regions. In addition to the proposed spatial-constrained adversarial perturbations, we also propose a novel structure-aware frequency adversarial attack method in the discrete cosine transform (DCT) domain. Since the proposed attacks are independent of the gradient estimation, they can be directly incorporated into existing gradient-based attacks. Experimental results show that, with a comparable attack success rate (ASR), the proposed methods can produce adversarial examples with considerably improved visual quality without the need of sacrifice of attack success rates. With the comparable perceptual quality, the proposed approaches achieve higher attack success rates.

Finally, this thesis explores evasion attacks on the composite task type which involves the matching, classification and regression tasks jointly. In this composite task type, we use the single object tracking model as a challenging case study, where we perform white-box evasion attacks in the real-world tracking scenes. The objective is to "blind" a single object tracker by pasting an adversarial sticker somewhere on the surface of a tracked object, i.e., the tracker/patch can no longer accurately estimate the position/size of its target object (using a bounding box) over time. Indeed, physical attack on such trackers is a very challenging task since it involves a combination of three tasks (using three sub-networks). To fool the matching sub-network, we especially design the maximum textural discrepancy (MTD), a resolution-invariant and target location-independent feature de-matching loss. MTD loss distills global textural information of the template and search images at hierarchical feature scales prior to performing feature attacks. To fool the classification and regression sub-networks, we propose motion model-incorporated shape attacks. The shape attacks can manipulate the estimated position/size of the target object in a controllable way. Furthermore, we employ a set of transformations to simulate diverse visual tracking scenes in real life natural settings. Experimental results show the effectiveness of the physically feasible attacks on SiamMask and SiamRPN++ visual trackers both in digital and physical scenes.

Chapter 6 concludes the thesis and discusses the future work.

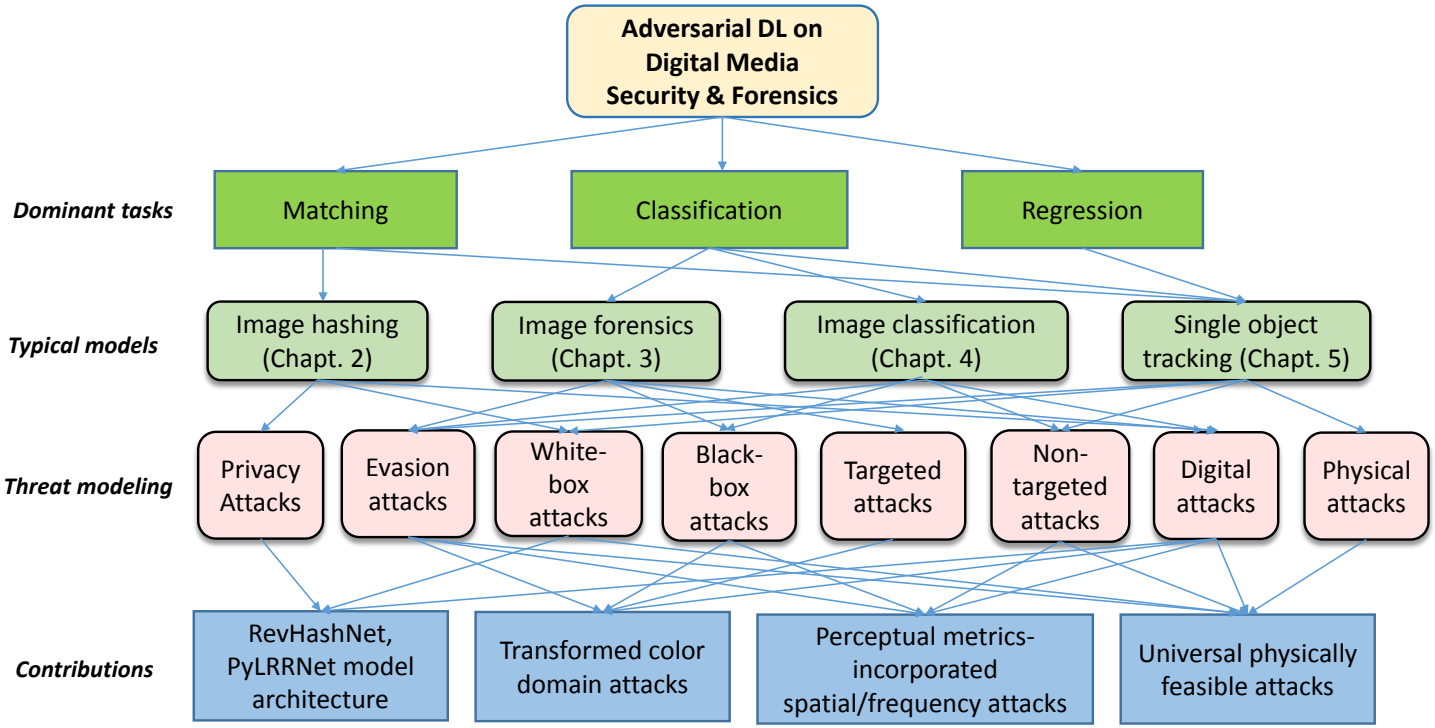


Figure 1.14: The roadmap of this thesis. This thesis plays the role as an attacker, which studies the vulnerabilities of three dominant types of computer vision tasks in digital media security and forensics: matching, classification and regression. With diverse threat modeling, four representative security-related models are selected as victim model examples of these three dominant types of tasks. The attacking approaches developed in this thesis can be generally employed to examine the vulnerabilities of other security-related models in matching, classification and regression tasks.

Chapter 2

A Case Study of Matching Task: Image De-hashing (Model Inversion Attacks on Image Hashing)

2.1 Introduction

With an overwhelmingly large volume of images generated in Internet every day, image hashing has attracted increasing attention for massive data indexing, fast image retrieval and image authentication [63, 77, 144, 154, 176, 244]. Image hashing is a scheme to generate unique image signatures that are supposed to be both compact and robust to non-content modifications. To efficiently retrieve images in big data era, a compact representation of images plays a pivotal role to deal with data deluge [40, 240]. Another desirable property of image hashing is perceptual robustness to image contents [154, 240]. Specifically, image hashing algorithms should yield similar image hashes with a higher probability for perceptually similar images of human vision systems than dissimilar ones.

There has been extensive research on image hashing. Generally, existing image hashing methods can be broadly divided into two classes: robust and secure image

hashing for content authentication, and image hashing specifically designed for similarity retrieval. Robust and secure image hashing methods generally incorporate pseudo-randomness to image hashes both secure and robust to content-based modifications [77, 154, 168, 219, 260]. In this Chapter, we mainly focus on the latter class and are particularly interested in its inversion issue (a.k.a model inversion attacks), a relatively less studied problem.

In image hashing for similarity retrieval, in general, there are two categories of learning to hash methods — unsupervised learning based methods, and supervised learning based ones. In unsupervised learning based methods, image hashing functions can be trained without semantic labels or relevance information. One of the representative unsupervised hashing methods is Spectral Hashing (SH) [250]. The SH method explores data distribution to learn the image hashing function. Some other efficient unsupervised hashing methods include Isotropic Hashing (ISOH) [117], Anchor Graph Hashing (AGH)[143], and Density Sensitive Hashing (DSH) [104].

In contrast, supervised learning based hashing methods exploit semantic similarity information. This class of image hashing techniques have been intensively studied in recent years. A popular algorithm in this category is the Binary Reconstruction Embedding (BRE) [120], which was proposed to minimize the squared errors between the distances of the original data points and those of the corresponding hashes. Other supervised learning based hashing schemes are Supervised Hashing with Kernels (KSH) [144], Hamming distance metric learning [176], and some deep learning based image hashing approaches [126, 279].

Despite a large number of image hashing methods based on similarity retrieval generating short and robust image hashes, one important aspect of image hashing, i.e., the *privacy* or *security* issue, has been largely neglected. Similar to a cryptographic hashing function for image security protection, image hash generation is also supposed to inherit the security property like the *one-way* function [154, 219, 251]. Ideally, the *one-way* function indicates that hash generation should be non-invertible. Specifically, it is easy to generate the image hash values; however, it is computationally infeasible to calculate or reconstruct the original images. Given a vectorized image $\mathbf{x}$, and the hashing function $h(\cdot)$, this property can be

mathematically expressed as follows:

$$\mathbf{x} \mapsto h(\mathbf{x}) \tag{2.1}$$

In fact, if attackers can revert $\mathbf{x}$ from $h(\mathbf{x})$, then they would probably break the *one-way* property, and raise *privacy* or *security* concerns. Suppose that an adversary attacked one image hash database and reverted images which contain private or confidential information, then information leakage would be inevitable. Therefore, in this scenario, the *one-way* property of image hashing algorithms is of essential importance in the context of information security. Particularly for the image case, only perceptual inversion (which reconstructs a highly perceptually similar image) is required. In this Chapter, we address one security concern in image hashing for similarity retrieval – reverting real-valued image hashing using deep Convolutional Neural Networks (CNNS).

Please note that this work is distinct from reconstructing images from their feature descriptors (e.g., Scale-Invariant Feature Transform (SIFT), Local Binary Pattern (LBP), Histogram of Oriented Gradients (HOG)) [49, 111, 248], since our proposed *RevHashNet* reverts image hashes directly. To our best knowledge, this is the first work to consider reverting the image hashing, thus possibly break the underlying *one-way* function assumption in certain image hashing methods. Indeed, if image de-hashing is not a preferred property by the image hash designer, the straightforward impact (or our primary motivation) of our proposed image de-hashing serves to raise the security awareness towards several image hashing methods for similarity retrieval and some security-enhancing rules can be incorporated into the hash design phase when designing image hashing schemes designed for similarity retrieval in real-life scenarios (e.g., to make the currently proposed image de-hashing techniques fail). To make retrieval-based methods more secure, we suggest adopting similar security techniques (e.g., pseudo-randomness, random projection) as for Robust and Secure image hashing methods. With secure image hashes, we could largely prevent our (confidential) data suffering from information leakage issues.

Our contributions in this part are summarized as follows:

1. *We address one missing property – one-way function in several image hashing*

methods for similarity retrieval. We then propose the new concept of perceptually invertible image hashes. Through our exploration of real-valued image hashing inversion, we hope our work can raise the potential security concern during designing new image hashing algorithms, or we can introduce an additional perceptually invertible property of image hashing when security is not a concern.

2. *We propose the RevHashNet architecture, a deep learning framework to de-hash images from some real-valued image hashes for similarity retrieval. This compact network consists of one fully connected layer, followed by six convolutional layers, yet is powerful to reconstruct perceptually similar images to the original ones from much lower dimensional image hash vectors.*

3. *We show the possibility of the previously assumed impossible image de-hashing problems using deep neural networks. Our work provides insights to exploit deep learning methods to tackle certain challenging de-hashing problems in image processing field.*

4. *We further develop the PyLRR-Net architecture, a novel image de-hashing network based on deep residual learning. The proposed PyLRR-Net learns to reconstruct the images in a progressive way. PyLRR-Net improves over RevHashNet in terms of image dehashing quality, scalability, and flexibility for large and color image dehashing problems.*

To our best knowledge, this is the first work proposing the image de-hashing concept and illustrating it by reconstructing the images from certain image hashes [104, 120, 121, 168, 250]. The rest of the work is organized as follows: in Section 2.2 we formulate the image hashing inversion as a nonlinear mapping optimization problem. We then propose *RevHashNet* in Section 2.3. Extensive experiments and evaluations are reported in Section 2.4. Moreover, we develop *PyLRR-Net* by leveraging deep residual learning in Section 2.5 to further improve the image de-hashing performance. Experimental comparisons are reported in Section 2.6 between the proposed *RevHashNet* and *PyLRR-Net*. In Section 2.8, we conclude the image de-hashing work and point out some open problems for future work in this direction. Through our explorations on model inversion attacks, we hope to

we could raise the security or privacy awareness of model designers in developing more secure image hashing methods.

2.2 Problem formulation

In the image de-hashing problem, we resort to finding a mapping function from an image hash to an image perceptually similar to the original image. Mathematically, the image inversion process can be expressed as:

$$g(h(\mathbf{x}), \mathbf{w}_{rec}) = \hat{\mathbf{x}} \tag{2.2}$$

where $g(\cdot)$ denotes an image hashing inversion function which maps hash values to the de-hashed image, $\mathbf{w}_{rec}$ means the reconstruction weight, $\hat{\mathbf{x}}$ represents the output of the image de-hashing algorithm. $\hat{\mathbf{x}}$ is desired to be perceptually similar to the original image $\mathbf{x}$ as much as possible.

We employ the supervised training to approximate the inversion function $g(\cdot)$, and parameterize it with trainable reconstruction weights $\mathbf{w}_{rec}$. In supervised training, the input data is set as image hash values $h(\mathbf{x})$ and the output is chosen to be the corresponding original image $\mathbf{x}$.

Given N training samples $\left\{\left(h(\mathbf{x}_i), \mathbf{x}_i\right)\right\}_{i=1}^{N}$, firstly, as a proof-of-concept, we define the objective function as the Euclidean distance between de-hashed images their original (ground truth) images:

$$\mathbf{w_{opt}} = \arg\min_{w_{rec}} \ell(\mathbf{w}_{rec}), \quad \ell(\mathbf{w}_{rec}) = \frac{1}{N}\sum_{i=1}^{N} ||g(h(\mathbf{x}_i), \mathbf{w}_{rec}) - \mathbf{x}_i||^2 \tag{2.3}$$

where $\mathbf{w_{opt}}$ denotes the optimized reconstruction weights of the image hashing inversion function $g(\cdot)$.

Now the de-hashing problem converts to solving the optimization problem in Eq.(2.3). The mini-batch stochastic gradient descent method is employed to minimize the cost function in Eq.(2.3):

$$\mathbf{w}_{rec}^{t+1} \leftarrow \mathbf{w}_{rec}^{t} + \alpha \nabla_{\mathbf{w}_{rec}^{t}} \ell(\mathbf{w}_{rec}^{t}) \tag{2.4}$$

where α is the learning rate, $\nabla_{(\cdot)}$ is the gradient operator, and the superscript t denotes the iteration number.

In practice, however, it is very challenging to give an analytical expression of the image de-hashing function $g(\cdot)$ due to its highly non-linear and non-convex complexity. Even worse, we do not even know whether or not there exists such a de-hashing function, mathematically. To resolve this problem, we leverage the functional expressiveness of deep neural networks to approximately learn such an image de-hashing function.

CNNs have gained popularity in the past few years. Starting from 2012, the ImageNet Large-Scale Visual Recognition Competition (ILSVRC) [41] has greatly fostered the popularity of CNNs in varieties of vision tasks. AlexNet [119] was the first deep CNNs that achieved significant improvements in classification accuracy over traditional machine learning methodologies. The powerful deep neural network was composed of five convolutional layers followed by three fully connected layers. Importantly, AlexNet alleviated the vanishing gradient problem by introducing a Rectified Linear Unit (RELU) after passing each convolutional layer. VGG [209] replaced large kernel-sized filters in AlexNet with multiple 3×3 kernels and went deeper in network depth. Some other popular architectures include GoogLeNet [224] from Google and ResNet [78] from the Microsoft.

However, all frameworks mentioned above were designed for detection and classification tasks. The same characteristic they share is to extract high-level image descriptors from high dimensional raw images. The size of the extracted features is much lower than that of the input image. Therefore, this problem resembles dimensionality reduction. In our image hashing inversion case, however, the network is desired to generate a perceptually similar image given a low dimensional real-valued image hash vector. Therefore we are solving an inverse problem where the forward transform tends to be highly nonlinear.

There are a few networks available for dimensionality increment applications. For instance, Dong et al. proposed to generate super resolution images from low resolution ones [45]. Our challenge here is that our input is a much lower dimensional real-valued hash vector instead of an image with low resolution. Dosovitskiy et al. tried to generate 2D images from a class label with a high-level 3D descriptor vector [50]. However, it belongs to the generative model which is still unsuitable

to our problem. One work which shares some similar ideas to our work is the ReconNet [123], which was proposed to recover images from Compressed Sensing (CS) measurements. However CS recovery [178, 234, 235] is generally different from image de-hashing mainly for three reasons: 1) CS is a linear model, and the sampling process can be modeled by multiplying a random measurement matrix, i.e., $\mathbf{y} = \Phi\mathbf{x}$, while image hashing tends to be highly nonlinear. 2) The inverse transform of CS is an explicit matrix inverse while the inverse form is latent and hard to find for image hashing methods. 3) CS recovery can be guaranteed, e.g., with the Restricted Isometry Property (RIP) [18], while no theoretical reconstruction guarantees are yet formulated for image de-hashing. Indeed, we believe that some image hashes may not be invertible (e.g., [154]).

2.3 RevHashNet method: perceptually de-hashing real-valued image hashes

In the following section, as an illustrative study case of image hashing for similarity retrieval, we briefly describe a set of classic image hashing methods, named Binary Reconstruction Embedding (BRE) [120], Spectral Hashing (SH) [250], and Density Sensitive Hashing (DSH) [104]. The image hashing generation is treated as a blackbox, which means we know nothing about the image hashing algorithms. The illustration of hashing generation process is shown in Fig. 2.1. We then present the *RevHashNet* implementations layer by layer in detail.

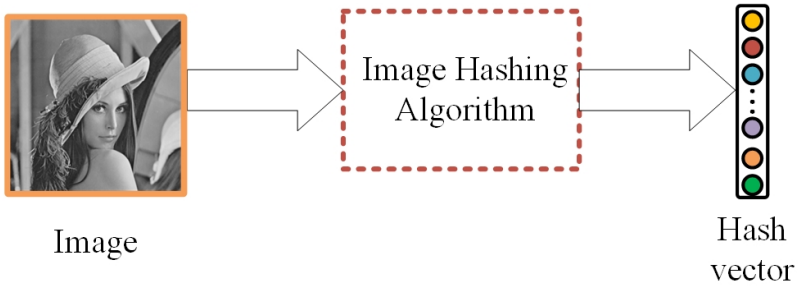


Figure 2.1: Illustration of the image hashing generation process.

2.3.1 Hashing generation

There are varieties of methods available for image hashing. One classic image hashing method is BRE [120], which was proposed to minimize the squared errors between the distances of original data points and those of corresponding hash vectors. In our experiments we selected BRE as one of the standard image hashing methods to generate the real-valued image hashes. Denote the vectorized image data set as $\mathbf{x}_i$ ($i = 1, 2, \cdots, n$). Then each data point can be projected to a much lower dimensional space using a set of hashing functions $h_1(\cdot), h_2(\cdot), \cdots, h_L(\cdot)$:

$$h_l(\mathbf{x}) = \sum_{q=1}^{s} \mathbf{w}_{lq} \kappa(\mathbf{x}_{lq}, \mathbf{x}) \tag{2.5}$$

$$\tilde{h}_l(\mathbf{x}) = \text{sign}(h_l(\mathbf{x})) \tag{2.6}$$

where L denotes the length of the hash vector, $\kappa(\cdot)$ is a kernel function which serves to introduce nonlinearity, $\mathbf{w}$ is a weight matrix of size $L \times s$, and can be learned by pairs of $(\mathbf{x}_i, \mathbf{x}_j), i, j = 1, 2, \cdots, n$.

It is worth emphasizing that, instead of using binary codes $\tilde{h}_l(\mathbf{x})$ from Eq.2.6, the inputs to our network are L real-valued image hashes from Eq.(2.5):

$$h(\mathbf{x}) = [h_1(\mathbf{x}), h_2(\mathbf{x}), \cdots, h_L(\mathbf{x})]^T \tag{2.7}$$

Our reasons for focusing on real-valued image hashes are as follows: 1) Reverting binary image hashes is more challenging than that of real-valued ones. We plan to work on binary image hashes in the future. 2) In practice, it is still possible for attackers to obtain real-valued image hashes. Therefore, we train the *RevHashNet* based on $h(\mathbf{x})$.

In addition to BRE, we apply another nonlinear image hashing method, spectral hashing [250], to verify the effectiveness of our proposed *RevHashNet*. The spectral hashing method [250] formulates the semantic image hashing task as a graph partioning problem. The image hashes are calculated from principal projections of the Laplacian of similarity graph. The third image hashing method we use is DSH [104]. DSH exploits geometric structure of the data. The DSH hashing functions are selected with the greatest entropy scores. Similar to BRE, we also use the real-valued

image hashes generated by these two image hashing methods.

2.3.2 RevHashNet model

To de-hash images from their hash values generated using Eq.(2.7), we propose *RevHashNet*. *RevHashNet* is a deep learning framework which resorts to find a mapping from image hashes generated by some image hashing algorithms to perceptually similar images.

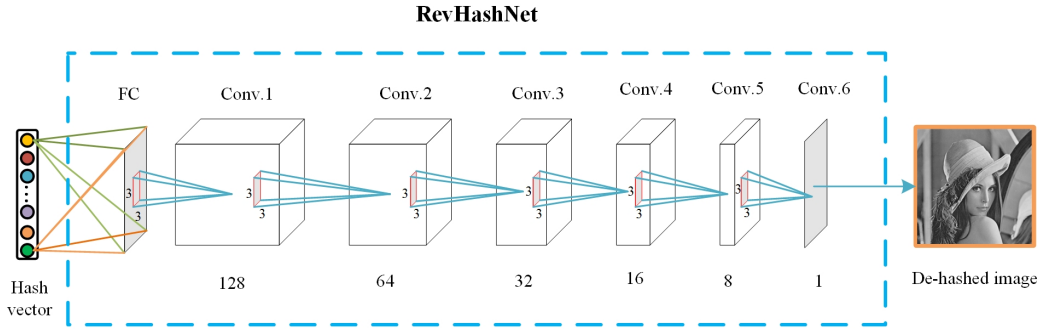


Figure 2.2: Illustration of the proposed *RevHashNet* architecture.

As illustrated in Fig.2.2, our proposed *RevHashNet* architecture consists of one Fully Connected (FC) layer, followed by six Convolutional (Conv.) layers. Firstly, we need to train the network to gain a satisfactory de-hashing performance. During training, the input to *RevHashNet* is one hash vector $h(\mathbf{x}) \in \mathbb{R}^L$, and the output is set as the groundtruth image $\mathbf{x} \approx g(h(\mathbf{x}), \mathbf{w}_{rec}) \in \mathbb{R}^{M \times N}$, where $M \times N$ is the spatial dimension of the original image. With different datasets, it is straightforward to modify the value of M, N without changing our network structure. After proper training, *RevHashNet* will automatically reconstruct a perceptually similar de-hashed image given an input hash vector.

Fully connected layer: In fully connected layers, every neuron in one layer is fully connected to all nodes in its previous layer. Without any information of the mapping function $g(\cdot)$, it is reasonable to assume that there exists one possible connection between each real hash value and each image pixel. This is the underlying justification of adopting a fully connected layer immediately after the input in Fig.2.2. More specifically, the input to the fully connected layer is a

hash vector $h(\mathbf{x}) \in \mathbb{R}^L$, and the output is a hidden vector in space $\mathbb{R}^{M\times N}$, which is then reshaped into an image (feature map) of dimension $M \times N$. In this way, the fully connected layer densely links the real hashed values with a feature map of size $M \times N$, where $M \times N \gg L$. However, the feature map is only a rough version of the original image. Therefore, we then feed the rough image through several convolutional layers with ReLU to learn a more complex nonlinear mapping function.

Convolutional layer: The essence of CNNs lies in the convolutional operation. In forward propagation of CNNs, a set of sliding filters are convolved with its previous layer to yield the feature maps. Denote the (i, j, k)-th pixel in the $(l-1)$-th layer as $x^{l-1}_{i,j,k} \in \mathbb{R}^{M\times N\times d^{l-1}}$, where d^{l-1} refers to the depth or number of channels in the $(l-1)$-th layer. Then we take d^l sliding fiters, each of which with the kernel size being $m \times m$ and depth d^{l-1}. After the convolutional operation, the output pixel $x^l_{i,j,k'}$ at the l-th layer can be formulated as:

$$x^l_{i,j,k'} = \sum_{a=0}^{m-1}\sum_{b=0}^{m-1}\sum_{k=1}^{d^{l-1}} x^{l-1}_{i+a,j+b,k}(w^l_{a,b,k,k'})_{rec} + b^l_{k'} \tag{2.8}$$

where $1 \le k' \le d^l$, and $(w^l_{i,j,k,k'})_{rec}$, $b^l_{k'}$ denote the trainable weights and bias of the k'-th filter in the l-th layer, respectively. To fix the spatial dimensions of the feature maps, we pad each feature map $x^{l-1}_{i,j,k}$ with $(m-1)/2$ lines of zeros around its borders. Specifically, in our proposed *RevHashNet*, we design a VGG style structure where we take uniform 3×3 kernel-sized filters for all of the six convolutional layers. Correspondingly, the number of channels are selected respectively as 128, 64, 32, 16, 8, and 1.

In detail, the input to Conv.1 is the rough image from the fully connected layer, and the output are 128 feature maps. We then apply ReLU to every pixel value of an individual feature map. After thresholding, all feature maps (with zero-paddings) are convolved with 64 spatial filters in the second convolutional layer to give 64 new feature maps. Similarly, the inputs for Conv.3 to Conv.6 are all zero-padded feature maps (passing through ReLU layer) from their preceding layer, and outputs are new feature maps with number 32, 16, 8 and 1, respectively. The output feature

map from Conv.6 is our perceptually de-hashed image.

ReLU layer: ReLU is a nonlinear activation function which introduces nonlinearity to the network, meanwhile making training more efficient by forcing negative values to be zero, i.e., $relu(x^l_{i,j,k'}) = max(x^l_{i,j,k'}, 0)$. Except the last convolutional layer, we apply a ReLU layer following the convolutional operations. Specifically, the input of the $l-$th ReLU layer are feature maps from the $l-$th convolutional layer, and ouput are thresholded feature maps which serve as input of the $(l+1)-$th convolutional layer. During backpropation training, it is straightforward to calculate the gradient,

$$\frac{\partial relu(x^l_{i,j,k'})}{\partial x^l_{i,j,k'}} = \begin{cases} 1, \, x^l_{i,j,k'} > 0 \\ 0, \, otherwise \end{cases} \tag{2.9}$$

It is worth noting that *RevHashNet* is a deep learning approach for image de-hashing. However, it does not necessarily adopt the same architecture as the one illustrated in Fig. 2.2, although as far as we are concerned our proposed architecture performs the best for perceptual image de-hashing amongst existing convolutional neural network architectures (e.g., yielding high visual quality). The GAN model [66] might also be valid for image de-hashing problems. Nevertheless, since currently GAN models are quite hard to train,there has not been much success in de-hashing images from image hashes using the deep generative architectures [42, 66, 129]. Here our primary purpose is to show the feasibility of image hashing inversion using deep convolutional neural networks. In practice, it is desired to design more sophisticated CNN architectures and tune the hyper parameters to yield even better performances. There are some empirical rules to design a new CNN network, and interested readers please refer to [211].

2.4 Experiments on RevHashNet

We conducted extensive experiments based on different datasets to verify the effectiveness of the proposed *RevHashNet*. Datasets and the source codes will be made publicly available online after publication.

2.4.1 Experimental datasets description

We conducted our de-hashing experiments on several publicly available datasets. The first dataset is Modified National Institute of Standards and Technology database (MNIST) [1], a standard dataset for handwritten digits 0-9 with uniform size 28×28. MNIST contains 60000 images for training, and 10000 images for test. Each digit has 600 training samples, and 100 test samples. MNIST dataset is frequently used to train/test deep learning architectures for recognition, classification, and similarity retrieval.

In addition to MNIST, we also compile and combine 3 small datasets to get our second dataset [45, 262], named MIX. The differences to MNIST dataset are: 1) Images in MIX are natural color or grayscale images and they vary in image sizes. 2) We provide much less number of training samples in this dataset than for MNIST. The reason is that in practical situations, the adversaries may not be able to obtain a large number of training samples. In detail, we split MIX into the training set, the validation set and the test set. The training set contains 91 color images, and the validation set includes 14 color images. Both of them can be downloaded from here [2]. We pre-process all color images by extracting their luminance components using YCbCr transform. For our test dataset, we compile two standard gray scale datasets, referred to as Dataset 1 [3] [4], and Dataset 2. In Dataset 1, there are 8 images of size 256×256. Dataset 2 consists of another 8 images of size 512×512, and can be downloaded from here [5]. All the 16 test images in MIX are high resolution and easily recognizable images.

2.4.2 Image hashing methods to be de-hashed

The first step is to generate image hashes. With BRE, SH, and DSH image hashing algorithms, we can readily prepare our training and validation pairs on both datasets. For image hashing generation, we use the codes provided by the authors' website and we use their default parameters. In contrast to the MIX dataset, MNIST is a

[1] http://yann.lecun.com/exdb/mnist/
[2] http://www.ifp.illinois.edu/ jyang29/
[3] https://github.com/ricedsp/D-AMP_Toolbox/tree/master/TestImages
[4] http://see.xidian.edu.cn/faculty/wsdong/NLR_Exps.htm
[5] http://decsai.ugr.es/cvg/dbimagenes/g512.php

simple grayscale dataset with uniform size, we make no data pre-processing for it. To fit the MNIST dataset in our proposed *RevHashNet*, we set the size of feature maps to be 28 in accordance with the image size.

The MIX dataset consists a number of large, high resolution images. Due to computational concerns, we divide each large image in this dataset into image blocks with a stride of 16 to form a set of sub-images. To make it consistent with the MNIST dataset, we choose the sub-image size to be 32×32. Then we apply three image hashing methods to all sub-images to obtain corresponding real-valued hash vectors, respectively.

In that way, we get our training pairs $(h(\mathbf{x}_i), \mathbf{x}_i)$. Here the output $\mathbf{x}_i$ is chosen to be a 32×32 image block, and the input to the network is an image hash vector $h(\mathbf{x}_i)$ generated by one of the three image hashing methods. Then we can readily train *RevHashNet* using the Stochastic Gradient Descent (SGD) method with mini-batch size $N = 128$. For all of our experiments, the learning rate is set to be 10^{-4}, and the momentum we use is selected to be 0.9. For fair comparison, we terminate our experiment when it iterates for 20000 steps. Each experiment takes about half an hour for training. After sufficient training, we can employ our pre-trained *RevHashNet* for test. In the following sections, we evaluate the de-hashing performance on both datasets.

Note that BRE, SH and DSH are all learning based image hashing methods and their model is trained once using the training data. We then apply the same trained hashing model to generate both the validation hash vectors and test ones. In addition to learning based image hashing methods, we also confirmed the de-hashing feasibility from Locality Sensitive Hashing (LSH) [63] and Kernelized Locality Sensitive Hashing (KLSH) [121], two popular image hashing benchmarks that are not learning based methods. For all of the experiments in the following section, we employ the same network architecture as shown in Fig. 2.2 . The experiments were carried out on one Nvidia Titan X (Pascal) GPU using Caffe [102] as our deep learning framework of choice.

2.4.3 Experiment 1: de-hashing tests on MNIST

After proper training of our proposed *RevHashNet*, we conduct extensive experiments to test its de-hashing performance on the MNIST and the MIX datasets. We also investigate its de-hashing performance in terms of different datasets, image hashing algorithms, and the number of training samples.

The de-hashing experiments for the MNIST dataset were carried out in 10000 images from the test dataset. After adequate training, *RevHashNet* can automatically reconstruct a perceptually similar image given a BRE hash vector. We found that the generated digits are human recognizable with image hashing length $L = 12, 16, 20$. Similarly, in our experiments, *RevHashNet* successfully de-hashed digit images from SH image hashes and DSH image hashes.

For demonstration, we randomly choose one ground-truth digit from each of the ten digit classes, and show in Fig. 2.3 the 10 digits and their reconstructed versions with different image hashing algorithms and image hashing length. In Fig. 2.3, we can recognize almost all de-hashed digits with high confidence. Also there are no significant distinctions of the reconstructed images generated by three different image hashing methods. The successful image de-hashing tests on MNIST verified the feasibility, and universality to some degree, of de-hashing images using our proposed deep learning architecture.

2.4.4 Experiment 2: de-hashing tests on MIX

Similar to the training stage, each image is individually cropped into a number of blocks. Each block is a sub-image, with equal size 32×32. We use proper zero padding around the image border to make the image size multiplier of the block size. Each subimage is then hashed into L real-valued hashes using the three studied image hashing methods.

During the test, we used the trained *RevHashNet* model to revert subimage hashes to subimages, and then concatenated corresponding reconstructed subimages together to form a large image. We further investigated the reconstruction quality of the images on two test datasets both qualitatively and quantitatively. Since we observed a similar reconstruction performance in de-hashed images from the three image hashing methods, for simplicity, we take the BRE image hashing method as

(a) DeHashing BRE hashes

(b) DeHashing SH hashes

(c) DeHashing DSH hashes

Figure 2.3: Reconstructed digit images using the proposed *RevHashNet* on MNIST: First three rows show de-hashed images when L=12, 16, 20 respectively, and the last row shows the original digits. (a), (b), and (c) are cases that image hashes are generated by BRE, SH and DSH algorithms respectively.

an illustrative example.

Firstly, we make a qualitative evaluation of the de-hashing performance. In the experiments, we observed similar de-hashing performances of *RevHashNet* for image hashes generated by three different image hashing methods. Due to space limit, here we just show comparison results of de-hashed images using BRE. Part of our de-hashing results are presented in Fig. 2.4 . The first three rows respectively represent the cases with the hashing length L=16, 32 and 64, and the images in the last row are the original ones. From the de-hashed images, we can clearly tell the content semantics, i.e., parrot, boats, cameraman, or Lena etc. Also, we generally have better visual quality as the length L increases since more information can be exploited for de-hashing. For Dataset 2, we noted similar de-hashing observations as illustrated in Fig. 2.5. Generally, the reconstructed images on Dataset 2 have better visual qualities than those on Dataset 1. By zooming into Fig. 2.5, we note better image quality with the increase of L. In practice, often the quality of the reconstructed image can be further improved with a denoiser, eg., BM3D [35], following the image de-hashing process.

Figure 2.4: Reconstructed images using the proposed *RevHashNet* on Dataset 1: First three rows show de-hashed images when L=16, 32, 64 respectively, and the last row shows the original images. The image hashes were generated using BRE. We recommend the digital version of this Chapter and zoom in to compare the de-hashing performance.

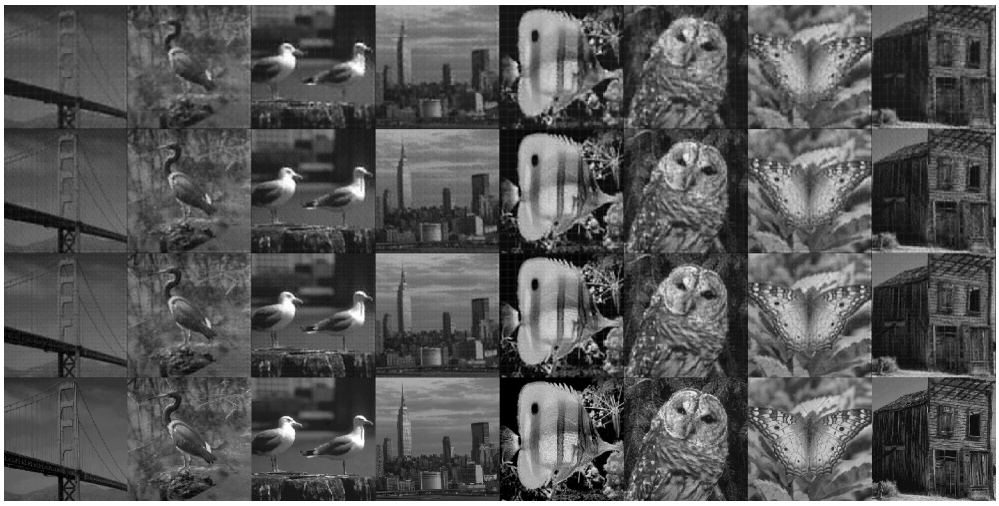


Figure 2.5: Reconstructed images using the proposed *RevHashNet* on Dataset 2: First three rows show de-hashed images when L=16, 32, 64 respectively, and the last row shows the original images. The image hashes were generated using BRE. We recommend the digital version of this Chapter and zoom in to compare the de-hashing performance.

Table 2.1: De-hashing performance on Dataset 1. Here L=16 equivalently yields a compression ratio as low as 1.56 %.

Dataset 1	L=16						L=32						L=64					
	PSNR (dB)			SSIM			PSNR(dB)			SSIM			PSNR(dB)			SSIM		
	BRE	SH	DSH	BRE	SH	DSH	BRE	SH	DSH	BRE	SH	DSH	BRE	SH	DSH	BRE	SH	DSH
Monarch	17.90	18.31	17.05	0.49	0.51	0.43	19.49	20.12	18.24	0.58	0.61	0.51	20.40	21.82	19.87	0.63	0.68	0.58
Parrot	21.50	22.01	21.15	0.66	0.70	0.63	22.23	22.91	22.03	0.70	0.73	0.68	22.90	24.75	22.74	0.72	0.77	0.70
Barbara	21.39	21.80	20.99	0.47	0.49	0.43	22.05	21.60	22.76	0.52	0.55	0.49	22.51	23.28	22.43	0.56	0.61	0.53
Boats	21.16	21.77	20.72	0.48	0.51	0.46	22.35	23.01	21.90	0.55	0.57	0.51	23.15	24.75	22.81	0.59	0.65	0.55
Cameraman	19.68	19.90	19.25	0.53	0.55	0.52	20.42	20.95	20.16	0.57	0.59	0.56	21.03	21.91	20.82	0.60	0.65	0.57
Foreman	25.31	25.89	24.29	0.68	0.71	0.64	26.28	27.64	25.93	0.72	0.76	0.70	27.54	29.24	27.25	0.74	0.79	0.71
House	22.96	23.35	22.36	0.60	0.64	0.58	23.92	24.97	23.38	0.65	0.68	0.62	24.87	26.77	24.50	0.68	0.73	0.63
Lena	21.55	22.25	20.84	0.55	0.59	0.52	22.68	23.65	22.28	0.61	0.65	0.59	23.41	24.90	23.18	0.65	0.70	0.62
Average value	21.43	21.91	20.83	0.56	0.59	0.53	22.43	23.23	21.96	0.61	0.64	0.58	23.23	24.68	22.95	0.65	0.70	0.61

Next, we qualitatively evaluated the de-hashing performance on two datasets with three different image hashing methods. We used the Peak signal-to-noise Ratio (PSNR) as one performance measure to evaluate the squared loss between the de-hashed image and the original image: $PSNR = 10log(\frac{MAX_I^2}{MSE})$, where MAX_I denotes the maximum possible pixel value of an image, MSE is defined as $MSE = \frac{1}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N}(\hat{\mathbf{x}}_{m,n} - \mathbf{x}_{m,n})^2$, where M and N are two dimensions of an image. We also used the structured similarity Structured Similarity (SSIM) index for visual

Table 2.2: De-hashing performance on Dataset 2. Here L=16 equivalently yields a compression ratio as low as 1.56 %.

Dataset 2	L=16						L=32						L=64					
	PSNR (dB)			SSIM			PSNR(dB)			SSIM			PSNR(dB)			SSIM		
	BRE	SH	DSH	BRE	SH	DSH	BRE	SH	DSH	BRE	SH	DSH	BRE	SH	DSH	BRE	SH	DSH
Goldbridge	27.27	27.66	27.63	0.68	0.72	0.70	27.37	28.58	28.26	0.70	0.74	0.71	28.08	29.35	28.19	0.72	0.77	0.69
Heron	25.78	26.50	25.47	0.59	0.63	0.58	26.38	27.70	26.47	0.64	0.67	0.62	27.28	28.84	26.84	0.67	0.73	0.62
Seagulls	24.78	25.38	24.64	0.69	0.72	0.71	25.82	26.56	25.66	0.73	0.75	0.73	26.67	28.37	26.04	0.75	0.80	0.73
Manhatan	25.57	26.22	25.45	0.59	0.62	0.59	25.92	26.96	26.20	0.62	0.65	0.61	26.47	27.89	26.29	0.64	0.70	0.60
Butterflyfish	19.42	18.90	19.02	0.28	0.28	0.26	20.05	19.97	19.72	0.32	0.35	0.29	20.63	20.72	19.80	0.36	0.40	0.29
Barowl	21.60	21.99	21.99	0.39	0.40	0.36	22.53	23.12	22.14	0.45	0.48	0.41	23.16	24.30	22.50	0.49	0.56	0.43
Butterfly	22.86	23.37	22.33	0.57	0.60	0.54	23.92	24.82	23.32	0.62	0.65	0.58	24.75	26.20	23.73	0.66	0.71	0.58
Bodie	20.88	21.21	20.79	0.35	0.37	0.34	21.47	21.84	21.37	0.40	0.42	0.38	21.85	22.84	21.50	0.43	0.51	0.38
Average value	23.52	23.90	23.31	0.52	0.54	0.51	24.18	24.94	24.14	0.56	0.59	0.54	24.86	26.06	24.36	0.59	0.65	0.54

evaluation [247]. SSIM is a built-in function in MATLAB as *ssim*. In addition, we assessed the impact of the hashing code length on reconstruction performance.

Specially, we carried out experiments with different number of real hash values: L=16, 32, and 64 respectively. The individual PSNR and SSIM measures are reported in Table 1. As observed in Table 1, PSNRs and SSIM indices generally increase about 1 dB and five percent as L gets doubled for all three image hashes, indicating a better reconstructed image quality. More importantly, we found that there exist only slight differences in reconstruction qualities for the same test image hashed with three different image hashing methods. For instance, PSNR indices of the de-hashed Foreman image are 19.68 dB from BRE hashes, 19.90 dB from SH hashes, and 19.25 dB from DSH hashes, respectively for the case of L = 16. We got the similar conclusion by checking Table 2 for the second test dataset of MIX. Successful de-hashing performances from image hashes generated by different image hashing methods show the effectiveness of the proposed *RevHashNet*.

Finally if we further visually check the images in the figures, we noted that, even with a low SSIM value, such as Monarch or Barbara for L=16, we can still have a good understanding of the image contents. A low SSIM probably originates from non-smooth transition between block borders. It is promising that the proposed *RevHashNet* is able to reconstruct perceptually similar images from some real-valued image hashes when L=16, which equivalently yields a compression ratio as low as 1.56 %.

De-hashing non-learning based image hashes

In addition to learning based image hashing methods (e.g., BRE, SH and DSH), similar experiments were conducted on non-learning based ones to verify the perceptually de-hashing performance of the *RevHashNet*. In non-learning based image hashing techniques, the hashing functions are not learned from the dataset but defined independently from the data at the cost of longer hashing codes [63, 240]. Among this category of image hashing methods, LSH [240] is known as an early exploration yet a most popular hashing algorithm which directly maps high dimensional original images to a low dimensional space. Indeed, many other benchmarks are variants of LSH [39, 121, 122]. Therefore, we set LSH as a standard non-learning based image hashing algorithm to generate image hashes for our following de-hashing experiments.

With image hashes generated from the LSH algorithm, we then train our proposed *RevHashNet* prior to testing its de-hashing performance on several datasets. During training, we used the same parameters as those in BRE, SH and DSH de-hashing experiments. From the experimental results, we observe that the visual quality of de-hashed images from LSH hashes is slightly worse than those from learning-based image hashing methods with $L \leq 32$. It is also observed that we have better understanding of the de-hashed images as L increases. Fig.2.6 shows 10 randomly selected digits that were de-hashed from the LSH image hashes with $L = 16$, 24, 32 and 48, respectively. Although vagueness exists in the reconstructed digits, we can distinguish most de-hashed digits without difficulty. Due to space limitation, we only present the averaged PSNR and SSIM indices for the MIX test datasets in Table 2.3. Similar to the MNIST dataset, we observed a satisfying de-hashing performance on this dataset.

Table 2.3: De-hashing LSH hashes on Dataset 1 and Dataset 2. Performance indices are averaged PSNR and SSIM values.

	L=16		L=32		L=64	
	PSNR(dB)	SSIM	PSNR(dB)	SSIM	PSNR(dB)	SSIM
Dataset 1	19.97	0.53	20.98	0.58	22.67	0.63
Dataset 2	22.10	0.49	22.89	0.53	24.38	0.58

Despite unsatisfying de-hashing performance was observed when $L \leq 16$, the

Figure 2.6: De-hashing LSH hashes using the proposed *RevHashNet* on MNIST dataset: First four rows show de-hashed images when L=16, 24, 32 and 48 respectively, and the last row shows the original digits.

visual quality of reconstructed images are still recognizable with relatively longer LSH hashes ($L \geq 32$). In practice, LSH algorithm tends to generate longer image hashes than those from learning-based image hashing methods to guarantee its retrieval performance.

It is worthy mentioning that, apart from LSH hashes, the proposed *RevHashNet* is also valid for some other non-learning based image hashing benchmarks. For instance, we observed comparable de-hashing performances both on the MNIST and the MIX datasets for KLSH [121], a more recent non-learning based hashing method. E.g., when *RevHashNet* works on Dataset 1 with hash length $L = 16$, the averaged PSNR and SSIM indices are 20.48 (dB) and 0.53, respectively. The two indices are 22.22 (dB) and 0.47 on Dataset 2 with the same length of image hashes. The metric values grow gradually as the hash length increases from $L = 16$ to $L = 64$. Compared with those shown in Table 3, we note slightly higher PSNRs and slightly lower SSIMs for KLSH hashes than for LSH hashes on both datasets, and the perceptual results are comparable.

2.4.5 Number of training samples

Image de-hashing problem falls into the security issue category since images in the dataset could contain confidential information. For a successful image de-hashing, the adversary needs a number of training samples to train *RevHashNet*. Since it

generally takes high cost to obtain training samples, a de-hashing mechanism which requires few training samples is preferable. Therefore, the number of training samples required for adequate training should be an essential quality measure of the *RevHashNet*.

We conducted experiments on the MIX dataset. In the experiments, we gradually decreased the number of training image samples from 91 to 20, and calculated the averaged PSNR and SSIM values of de-hashed images. Although we found a decreasing trend of performance indices during de-hashing DSH image hashes, PSNR and SSIM almost stay unchanged for BRE, SH, LSH and KLSH image hashes. In this matter, our model can still get reasonable performance with a few number of training samples.

2.4.6 Preliminary study of de-hashing secure image hashes

The proposed *RevHashNet* was primarily designed to perceptually de-hash image hashes for similarity retrieval, and its feasibility in such image hash inversion has been demonstrated in previous sections. In addition to image retrieval hashes, as a preliminary study, we also attempted to employ the proposed *RevHashNet* to revert images from secure image hashes, which are widely used in content authentication, identification, and other applications [77, 154, 168, 244]. In this type of secure image hashes, generally the *one-way* function property is explicitly assumed, as a necessary property, to guarantee that the image hash generation should be noninvertible.

In our preliminary study, we tried to de-hash images from a traditional secure image hashing based on Nonnegative Matrix Factorization (NMF) [168], and we obtained some promising results. For example, a well-trained *RevHashNet* is able to reconstruct an image from its NMF hash with relatively high visual quality, i.e., averaged PSNRs and SSIMs are 20.12 (dB), 0.53 on Dataset 1, and 22.34 (dB), 0.50 on Dataset 2, respectively.

Nevertheless, reverting secure image hashes are indeed much more challenging than that for image retrieval hashes. For instance, even for de-hashing NMF hashes on the MNIST dataset, the *RevHashNet* needs about 10^6 iterations to obtain a satisfactory de-hashing performance, which is about 50 times more than de-hashing similarity

(a) DeHashed MNIST digits from NMF hashes (b) Groundtruth MNIST digits

Figure 2.7: De-hashing NMF hashes using the proposed RevHashNet on the MNIST dataset when $L = 32$. (a) shows 100 randomly selected de-hashed MNIST digits, and the groundtruth digits are shown in (b).

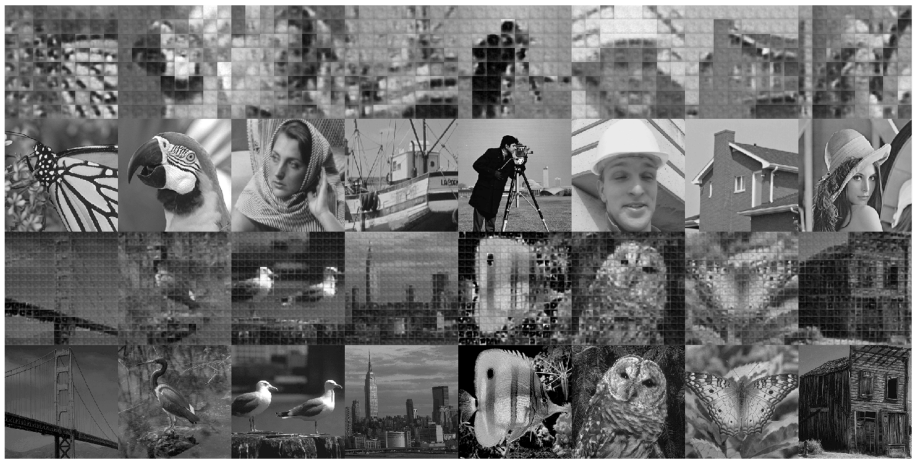


Figure 2.8: Reconstructed images using the proposed RevHashNet on Dataset 1 and Dataset 2: de-hashed NMF hashes from Dataset 1 and Dataset 2 with hash length $L = 32$. The first two rows show the de-hashed images on the Dataset 1, and groundtruth images, respectively. The third and fourth rows display de-hashed images on the Dataset 2.

retrieval hashes. In addition, the introduction of more sophisticated security-enhancing rules or complex feature extraction techniques (e.g., shape contexts) in secure image hashes would pose more challenges for the image de-hashing task. More thorough investigations on different secure image hashes are needed, and we plan to explore in depth towards de-hashing secure image hashes in our future work.

2.5 PyLRRNet method: coarse-to-fine image de-hashing using deep pyramidal residual learning

2.5.1 Image de-hashing with less number of bits

RevHashNet [245] is the first work to reconstruct perceptually recognizable images from image retrieval hashes based on deep learning approaches. In this work, RevHashNet learns the inverse mapping function from the hash space to the image space, using a fully connected layer followed by six convolutional layers with nonlinear activation. However, the perceptual quality of dehashed images degrades severely when the real-valued hashes have been quantized with less number of bits. Besides, RevHashNet was proposed to deal with gray scale and smaller-sized image dehashing problems, while color and larger image dehashing problems remains unexplored.

In this section, we propose a coarse-to-fine fully convolutional image dehashing framework using deep pyramidal residual learning. Instead of directly mapping image hashes to images with full connections, we learn to reconstruct images in a progressive way using fully convolutional operations. To better learn residuals of coarsely reconstructed images, we design a Long-Range Residue (LRR) module, which can be conveniently inserted at different image scales. Finally, we adopt the ℓ_1 loss and structural similarity measure as cost function to further improve the perceptual quality of reconstructed images. The proposed PyLRR-Net has shown superior performance over RevHashNet [245] in dehashing image hashes on the MIX and ImageNet datasets.

2.5.2 Progressive image de-hashing based on long-range deep residual learning

In the previous section, we show that the RevHashNet is feasible for the image dehashing task; however, the perceptual quality of dehashed images remains to be improved when real-valued hashes are quantized to less bits. In addition, the full connections in RevHashNet are prone to the well-known overfitting problem. The overfitting problem in turn limits the network's flexibility to extend to larger-sized images or color images.

To this end, we propose to progressively upscale image reconstruction sizes using fully convolutional operations. In addition, we design and use a plug-in Long-Range Residual Long-Range Residue (LRR) module at each image reconstruction scale to further boost the reconstruction quality of dehashed images.

Fig. 2.9 illustrates the overall working mechanism of the proposed PyLRR-Net. The top part shows the PyLRR-Net architecture, and the bottom part visualizes the outputs from the operational layers above them. We will explain the implementation details of each operation as follows.

The red block denotes deconvolutional operation, which is also known as the fractionally-strided convolution or transposed convolution. Deconvolution is a special form of convolution which reverses the forward and backward process of convolution to upscale the size of input images [57]. In the first deconvolutional layer, the kernel size k is set as 4×4, and in total there are 256 kernel-sized filters. With moving stride s as 1, sliding and convolving the hashes with the filters gives 256 feature maps, each with a size 4×4. The feature maps are activated using Rectified Linear Unit (RELU). In the second deconvolutional layer, we apply 128 kernel-sized sliding filters with stride 2, and kernel-size 4×4 to get 128 feature maps of size 8×8, followed by RELU for activation. From the third deconvolutional operation, we have a modularized sequential operations: firstly, a deconvolutional layer with kernel size 4×4, stride 2 and filter number equals image channel ch ($ch = 1, 3$ for gray scale and color images, respectively). Next follows the RELU activation and an LRR block to enhance residual learning. With one such sequential operation, we have a $2\times$ upscaling in image size. Thus, the output images form a pyramid with a scaling factor as 2. To get an image of size 2^K ($K \geq 4$), we can simply stack $(K-3)$ such modules after the second deconvolutional layer.

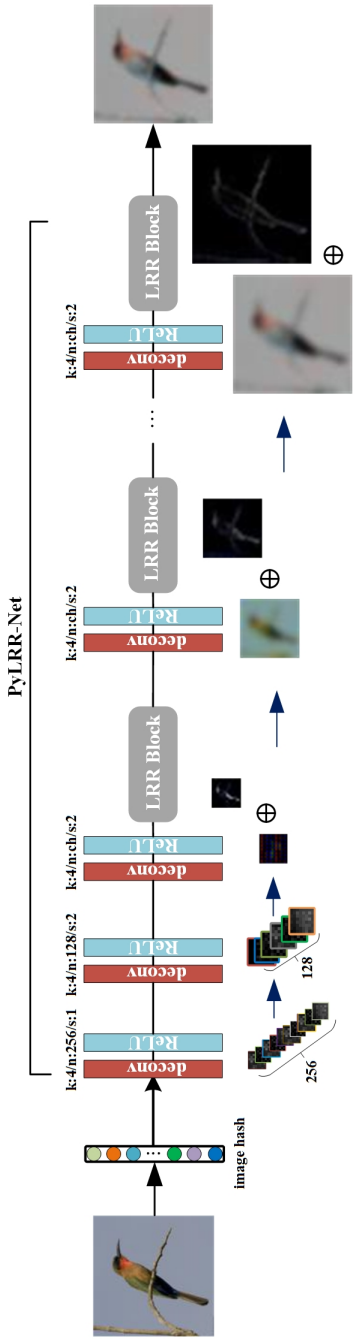


Figure 2.9: Illustration of the image de-hashing pipeline using the proposed PyLRR-Net.

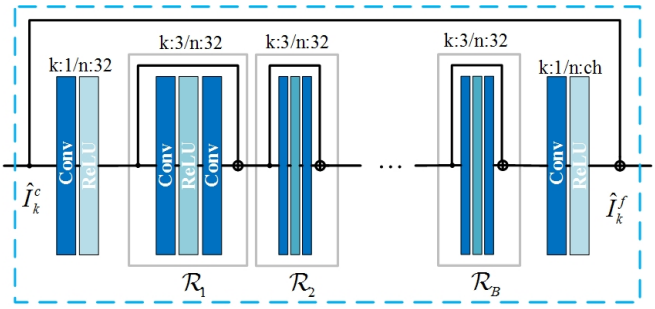


Figure 2.10: Illustration of the proposed LRR Block.

Fig. 2.10 shows an illustration of an LRR block. The LRR block is utilized to restore image residual components which necessitates deeper layers. We use a modified version of standard residual blocks [79] with the batch norm layer removed [136]. To increase the network capacity, we project the coarse image $\hat{I}_k^c$ from $n = ch$ channels to $n = 32$ using 1×1 convolutions. Next follows a sequence of residual blocks $\mathcal{R}_i, i = 1, 2, \cdots, B$ with uniform 3×3 kernels and number of filters as 32. Finally, to get the refined image, we use a second 1×1 convolution to convert the residual channel back to image channel ch. The sequential operations can be described as a function $\mathcal{R}$, where $\mathcal{R} = \prec_{1\times1} \circ \mathcal{R}_1 \circ \mathcal{R}_2 \circ \cdots \circ \mathcal{R}_B \circ \succ_{1\times1}$. However, we observe that trivially stacking residual blocks can impede the image dehashing performance, especially when the networks are getting deeper. Similar to residual modules, we shortcut a set of residual blocks to form the LRR block (we select $B = 3$ which is observed to work best). With a coarse image $\hat{I}_k^c$ at the k-th image scale, the refined image $\hat{I}_k^f$ can be obtained as,

$$\hat{I}_k^f = \hat{I}_k^c + \mathcal{R}(\hat{I}_k^c) \tag{2.10}$$

2.5.3 The objective function

In image restoration tasks, L_1 loss function generally outperforms the L_2 loss, which tends to generate blurry image reconstructions. Therefore we use L_1 loss function to measure the absolute distance between the dehashed images $\mathcal{G}(x_i)$ and their original counterparts $I_i, (i = 1, 2, \cdots, N)$,

$$\mathcal{L}_{\ell_1}(\mathcal{G}(x), I) = \frac{1}{N}\sum_{i=1}^{N} |\mathcal{G}(x_i) - I_i|_1 \tag{2.11}$$

To further improve the perceptual quality of image dehashing reconstruction, we introduce the Structured Similarity (SSIM) [247] as an additional loss term. SSIM is a perceptual metric that quantizes the structural distortion between two images, which relates to human vision system. For an image patch centered at pixel $p_{i,j}$, the SSIM index between a perceptually dehashed image patch $p_d \in \mathcal{G}(x)$ and the original image patch $p_o \in I$ can be calculated,

$$SSIM(p_{i,j}) = \frac{2\mu_{p_d}\mu_{p_o} + C_1}{\mu_{p_d}^2 + \mu_{p_o}^2 + C_1} \cdot \frac{2\sigma_{p_d p_o} + C_2}{\sigma_{p_d}^2 + \sigma_{p_o}^2 + C_2} \tag{2.12}$$

where μ_{p_d}, μ_{p_o} denotes the mean value of image patch p_d and p_o; $\sigma_{p_o}, \sigma_{p_o}$ represents the standard deviation of image patch p_d and p_o, respectively. $\sigma_{p_d p_o}$ is the covariance of the two image patches. As default parameters, the window size is set as 11×11, $C_1 = (0.01 \times l)^2$ and $C_2 = (0.03 \times l)^2$, where l refers to the dynamic range of an image [247].

For a training image pair $(\mathcal{G}(x), I)$, let W, H denote the width and height of the image I, then the SSIM loss is defined,

$$\mathcal{L}_{SSIM}(\mathcal{G}(x), I) = 1 - \frac{1}{WH}\sum_{i=1}^{W}\sum_{j=1}^{H} SSIM(p_{i,j}) \tag{2.13}$$

Then our overall loss function $\mathcal{L}$ is formulated as,

$$\mathcal{L} = \lambda_{\ell_1}\mathcal{L}_{\ell_1} + \lambda_{ssim}\mathcal{L}_{ssim} \tag{2.14}$$

where $\mathcal{L}_{\ell_1}$, $\mathcal{L}_{ssim}$ represent L_1 loss and SSIM loss, respectively. For the loss function, λ_{ℓ_1} and λ_{ssim} parameters are tuned to be 0.15 and 0.85 with grid-search strategy.

2.6 Experiments on PyLRRNet

2.6.1 Experimental dataset description

We conduct experiments on both gray scale and color image datasets to verify the dehashing performance of the proposed PyLRR-Net. For fair comparison [245], we tested on MIX [45, 245, 262], the same dataset used in RevHashNet for gray scale image dehashing [245]. MIX is a compilation of 3 datasets [45, 245, 262]. The images are converted to grayscale. Then each image is divided to 32×32 blocks with stride 16. And spectral hashing is applied to each block to obtain real-valued image hashes of length L. In summary, there are 17202 image blocks in the training dataset and 3426 blocks in the validation dataset. The test dataset consists of 16 images, with 8 images of size 256×256 (Dataset 1), and the rest 8 images of size 512×512 (Dataset 2). During the test stage, we concatenate the dehashed 32×32 image blocks to form the final images.

In contrast to RevHashNet [245], the proposed PyLRR-Net can be easily extended to dehash color images by simply changing $ch = 3$ at each image scale. In addition, larger image dehashing can be obtained conveniently by adding a deconvolutional layer followed by an LRR residual learning block. We show the feasibility of color image dehashing performance on the ImageNet subset, where we randomly select 100 image categories. To reduce the computational cost, we center crop each color image with window size 128×128 and then resize its image size to be 64×64. Finally, the image hashes can be obtained by applying image hashing method to each image. In total, there are 120k images in the training dataset, 4503 images for validation, and 4000 images for test.

2.6.2 Network training details

With the training pairs, $(x_i, I_i), i = 1, 2, \cdots, N$, we feed them to the deep neural networks with a mini-batch size of 128. For both RevHashNet [245] and PyLRR-Net, the learning rate is set as 0.0002, and both are optimized using the Adam optimizer [115] with momentum β_1=0.9, β_2=0.999, and weight decay as 0.02. The learning rate will be decayed by a factor of 0.9 for every 20 epochs. We train the neural networks with 500 epochs with early stopping. For the PyLRR-Net, in the

gray scale image dehashing experiments, image channel $ch = 1$, and image scale $K = 4$ for 32×32 block; while for color image dehashing, we can simply change $ch = 3$, and set $K = 5$ by adding an extra deconvolutional layer followed by one LRR block. All experiments were conducted with PyTorch on one GTX NVIDIA TITAN X GPU card.

2.6.3 De-hashing experiments on MIX

For grayscale image dehashing on the MIX dataset, we experimented with the hash length $L = 64$ and $L = 32$. Each hash value is quantized to 8 bits, 4 bits, 2 bits and 1 bit, respectively. To evaluate the perceptual quality of dehashed images, we adopt PSNR, SSIM [247] to measure the pixel-level distortion and structural distortion, respectively. Table 2.4 shows the experimental results of RevHashNet and PyLRR-Net, where $L = 32$. The proposed PyLRR-Net outperforms RevHashNet on both datasets in terms of PSNR and SSIM. For Dataset 1, in particular, the averaged PSNR/SSIM of dehashed images are improved by 0.39/3%, 0.49/3%, 0.78/6% and 0.15/5%, with quantization accuracy from 8 bits to 1 bit. Meanwhile, we employ Feature Similarity (FSIM) [269] / Visual Information Fidelity (VIF) [207] as additional visual metrics and observe an average improvement of 1.25%/2% in the four cases, quantitatively confirming a better perceptual quality. Even with a PSNR/SSIM improvement of 0.39 dB/3%, we observe a clearly better visual quality. As shown in Fig. 2.11, the first two columns show samples from Dataset 1 and the last two columns are samples from Dataset 2. The rows show dehashed images from RevHashNet [245], PyLRR-Net and the ground truth images, respectively. By zooming in the images, we can see a mitigation of blurry artifacts with PyLRR-Net (e.g., the elbow region in the 1st column, the helmet/face part in the 2nd column), and dehashed images appear to be less noisy (e.g., the sky in the 3rd column, the bird's eye and body regions in the last column). We note a larger performance gain for Dataset 1 than for Dataset 2, probably because the former dataset is harder to dehash with RevHashNet, while PyLRR-Net can still provide a good visual quality. The overall performance gain on both datasets could be explained by the leverage of the proposed coarse-to-fine deep residual learning schemes. Though not reported here due to space limit, we observe similar performance patterns with different

image hashing lengths (e.g., $L = 64$).

Table 2.4: Image dehashing performance on the MIX datasets ($L = 32$, PSNR measures in dB).

L=32		8bits				4bits				2bits				1bit			
Evaluation index		PSNR	SSIM	FSIM	VIF	PSNR	SSIM	FSIM	VIF	PSNR	SSIM	FSIM	VIF	PSNR	SSIM	FSIM	VIF
Dataset 1	RevHashNet	23.21	0.67	0.77	0.47	22.67	0.65	0.76	0.40	19.53	0.54	0.70	0.21	19.15	0.52	0.69	0.17
	PyLRR-Net	**23.60**	**0.70**	**0.78**	**0.50**	**23.16**	**0.68**	**0.77**	**0.43**	**20.31**	**0.60**	**0.72**	**0.22**	**19.30**	**0.57**	**0.70**	**0.18**
Dataset 2	RevHashNet	24.85	0.60	0.85	0.48	24.26	0.58	0.83	0.39	21.55	0.50	0.74	0.18	21.25	0.49	0.73	0.16
	PyLRR-Net	**25.16**	**0.60**	**0.86**	**0.52**	**24.60**	**0.59**	**0.83**	**0.40**	**22.18**	**0.52**	**0.75**	**0.19**	**21.42**	**0.51**	**0.74**	**0.17**

Figure 2.11: De-hashed MIX image samples, where $L = 32$ and each hash value is quantized into 4 bits. The first two rows show dehashed images using RevHashNet [245] and the proposed PyLRR-Net respectively, and the last row are original images. We recommend to zoom in the digital images for better comparison.

2.6.4 De-hashing experiments on ImageNet

For the color image dehashing experiments on the ImageNet subset, the hash length is set as $L = 256$. With different quantization accuracy of each real-valued hash

value, i.e., 8 bits, 4 bits and 2 bits, three experiments were carried out. The averaged PSNR(dB)/SSIM results are: 21.80/0.63, 20.33/0.60 and 19.38/0.55, and the FSIM/VIF metrics are: 0.76/0.52, 0.73/0.46, 0.67/0.33, respectively. Despite that certain image details can no longer be reconstructed, most of the objects are recognizable from the dehashed images. In Fig. 2.12, we show some representative images with different PSNR and SSIM values. With lower PSNR/SSIM indices, the dehashed images are either blurry (e.g., the 2nd row: the first and the second image from left) or display color distortion (e.g., the 2nd row: the third and fourth image from left). This could be explained by the highly ill-posedness nature of the image dehashing problem, where detailed information might not be well preserved in the image hashes.

2.7 Preliminary studies on different de-hashing schemes

In addition to RevHashNet and PyLRR-Net models, we also conducted preliminary studies on other image dehashing schemes. We will briefly describe and discuss such approaches in this section, and we will leave more comprehensive investigations as our future work.

2.7.1 Image de-hashing with adversarial losses

Generative Adversarial Network (GAN) is shown to be capable of generating photorealistic images. Therefore, we are motivated to formulate the image dehashing problem in an adversarial manner. Specifically, the proposed method consists of two components, the Generator (G) and the Discriminator (D). The role of G is to generate sharp and realistic images from image hashes, and D is to discriminate the authenticity of generated images by comparing with real images.

Let us denote a hash vector as $h(x) \in \mathbb{R}^L$ corresponding to the groundtruth image $x \in \mathbb{R}^{M \times N}$, then the adversarial loss function is expressed as,

$$\mathcal{L}_{adv} = \min_D \max_G \mathbb{E}_{x \sim p_{\text{data}}(x)} \log\Big[D(x)\Big] + \mathbb{E}_{z \sim \mathcal{H}(x)} \log\Big[1 - D(G(z))\Big] \qquad (2.15)$$

where we assume x and $h(x)$ follow data distributions $p_{\text{data}}(x)$ and $\mathcal{H}(x)$, respectively. To encourage the fidelity of reconstructed images over groundtruth images,

Figure 2.12: De-hashed ImageNet samples with different reconstruction quality. The 1st and 3rd row show the original images, and the 2nd and 4th row show the reconstructed images when $L = 256$ with 8-bit quantization. In the square brackets, the first number denotes the PSNR value (dB), and the second one is for SSIM index.

we further add a penalization loss,

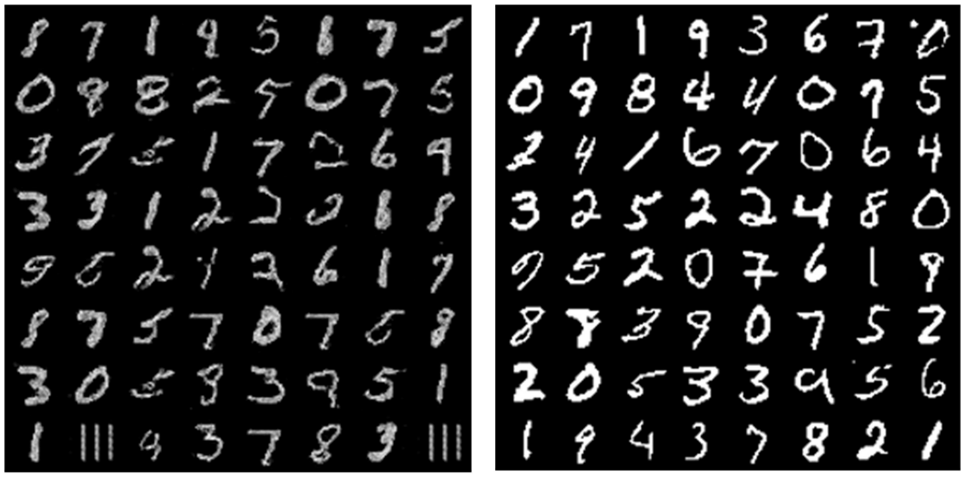
$$\mathcal{L}_{mse} = \frac{1}{N}\sum_{i=1}^{N}\left\|G(h(x_i)) - x_i\right\|_2^2 \tag{2.16}$$

The overall loss can be expressed as,

$$\mathcal{L} = \mathcal{L}_{adv} + \lambda\mathcal{L}_{mse} \tag{2.17}$$

where λ denotes the hyperparameter to balance the perceptual quality and image fidelity.

With the proposed loss, we conduct experiments on MNIST and CIFAR-10 (grayscale version) datasets, respectively. In both experiments, G adopts the RevHashNet architecture. D is a four-layer convolutional network. Except the last layer, we set kernel size as 4×4, stride as 2 and padding as 1; and the activation function is selected as LeakyReLU with parameter 0.2. For the last convlutional layer, we set kernel size as 4×4, stride as 1 and no padding; and the activation function is selected as the Sigmoid function. For both G and D, we employ the Adam optimizer with hyperparameters β_1, β_2 as 0.5 and 0.999, respectively. The learning rate is selected as 1e-3 with step decay by 0.999 for every 15 epochs. The overall training epoch is 500 and we adopt early stopping to choose our models on a validation dataset. For the hashing algorithm, we use SH [249]. On MNIST, we tune $\lambda = 1$; and on CIFAR-10, we use $\lambda = 2$. The hashing length for MNIST and CIFAR-10 are respectively: $L = 8$ binary codes and $L = 64$ and we quantize real-valued hashes to 8bits for each hash entry.



(a) DeHashed MNIST digits w/ GAN loss (b) Groundtruth MNIST digits

Figure 2.13: De-hashed MNIST digits using the proposed method with hash length $L = 8$. (a) shows 100 randomly selected de-hashed MNIST digits reconstructed with GAN loss, and groundtruth digits are shown in (b).

(a) Dehashed images w/ GAN **(b)** Dehashed images w/o GAN loss

(c) Groundtruth images

Figure 2.14: De-hashed images using the proposed method on CIFAR-10 with hash length $L = 32$: (a) and (b) show a comparison of de-hashed images with GAN loss and without GAN loss, respectively, and (c) shows the groundtruth images. We recommend to zoom in the digital images for better comparison.

In Fig. 2.13, we show some image dehashing examples on MNIST with hashing length $L = 8$ on binary codes. Despite limited information in binary codes, the proposed method is able to dehash digits from given binary codes with high visual quality. Nevertheless, we observe that some digits actually appear differently with the groundtruth digits. For instance, for the digit "1" in the 8^{th} row / 8^{th} column in (b), surprisingly, the generator produces a "111" which never exists in the training dataset. In Fig. 2.14, we compare dehashed images between using the adversarial loss and without the adversarial loss. The experimental comparison shows that, GAN loss can indeed improve the visual quality of some dehashed images by making them visually sharper (whereas G w/ the adversarial loss produces blurry images); however, such method can also produce completely different images by random guessing when the generator is not sure of its decision. This interesting observation raises new questions, e.g., how to avoid making wrong decisions when the generator does not have high confidence over its decisions? Besides, future effort can be put into investigating the underlying reason about the phenomenon that GAN dehashes very different images from its training dataset (e.g., in Fig. 2.13).

2.7.2 Image de-hashing with knowledge distillation

Knowledge distillation (KD) was proposed by Hinton et al. [83] to transfer knowledge from one model to the other. The two models are named as the teacher model and the student-model, respectively. Since KD sometimes can improve the performance of a student-model, we are therefore motivated to incorporate KD in the image dehashing scenario. We formulate image dehashing in a self-distillation manner. To be specific, we postulate the pretrained real-valued dehashing network as the teacher model while a new network as the student network which performs quantized-valued image dehashing. The intuition is that, a well-trained model may make it easier to train another one by transferring knowledge to it.

Fig. 2.15 illustrates the knowledge distillation-based image dehashing approach. On the left is a RevHashNet pretrained on real-valued image hashes, and on the right is a model targeting quantized-valued image hashes which has a same architecture as the pretrained RevHashNet. In our preliminary study, we quantize SH hashes into 8bits for each hash value, with hashing length $L = 32$ on the MIX dataset. The loss

function is selected as an addition of the vanilla loss computed using groundtruth images (as in Eq. 2.3) and a feature loss with identical weights. To compute feature losses, we move layers individually from the FC layer to 6 convolutional layers one by one. After sufficient training, we compute the averaged PSNR using distilled models separately. The PSNR (dB) results are respectively as: 20.75, 20.81, 20.35, 21.20, 21.23, 21.25. We observe that PSNRs tend to increase as we move the layers from the first layer to the last convolutional layer. This trend indicates that latter layers contribute more to image dehashing quality. However, these results are still inferior to image dehashing without feature losses, suggesting side effects of introducing additional supervision in intermediate layers in image dehashing.

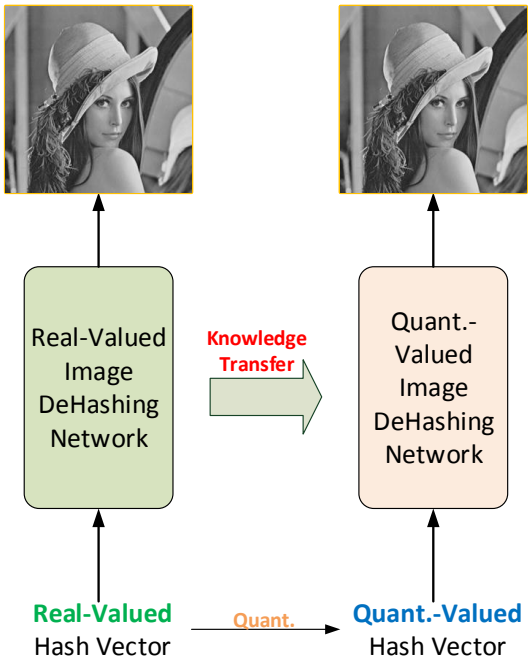


Figure 2.15: Illustration of image dehashing with self-knowledge distillation.

2.7.3 Image de-hashing in DCT domain

Natural images display the sparsity property in certain supports (e.g., DCT or wavelet domains). However, the sparsity is not explicitly encoded in image dehashing networks since the loss is defined in spatial domain. To possibly leverage the

sparsity property, we conduct a preliminary study of image dehashing in the DCT (Discrete Cosine Transform) domain.

Instead of directly learning to reconstruct image intensities, we are going to reconstruct DCT coefficients. The underlying reason is that, it may be easier to reconstruct larger DCT coefficients in this domain despite loss of accuracy in smaller ones. Therefore, compared with spatial domain reconstruction, we may have better dehashing quality since our larger DCT coefficients are more accurate. In the experiments, we keep the network and training strategy the same, while computing the dehashing loss in the DCT domain. The DCT transform is computed on 32×32 blocks (we found smaller blocks produce worse results than larger ones). Our experiments on the MIX dataset produce averaged PSNRs as 21.14 dB. The preliminary result indicates that image dehashing in the DCT domain does not display tendency to improve image dehashing performances.

2.8 Conclusion

In this Chapter, we consider a matching task type. The representative victim model is image hashing models for image similarity retrieval and authentication purposes. We study image de-hashing attacks on this representative model. Firstly, we propose the concept of image de-hashing and present the RevHashNet, a deep learning approach, to reconstruct perceptually similar images from the corresponding real-valued image hashes. Our extensive experimental results from several classic image hashing methods support that a trained RevHashNet is able to de-hash visually recognizable images similar to the original ones. Our preliminary experimental results also demonstrate the possibility of reverting images from some secure image hashes (e.g., for content authentication applications).

We then propose the PyLRR-Net, a novel image dehashing network based on deep residual learning. Instead of directly mapping image hashes to the image space, the proposed PyLRR-Net learns to reconstruct the original images in a progressive way. This modification makes it especially amendable for different image scales and image channels (grayscale or color images). To refine the reconstructed images, we design and insert an LRR module at each image scale to learn image residuals. Experimental results show that PyLRR-Net improves the image de-hashing quality

both qualitatively and quantitatively over RevHashNet. Through our exploration in such image de-hashing cases, this Chapter intends to raise the security awareness of model designers of image hashing. Adversaries may readily perform privacy attacks by leveraging the functional approximation capability of deep neural networks.

In addition to our proposed two methods, many open problems remain to be explored in the future. For example, it is desired to further improve the de-hashing performance on binary image hashes. Besides, we believe it is also interesting to extend the image de-hashing concept to other multimedia modalities, e.g., video de-hashing or multi-modal digital media de-hashing.

Chapter 3

A Case Study of Binary Classification Task: Exploring Imperceptible and transferable GAN-generated Fake Face Imagery AntiForensics

3.1 Introduction

In last Chapter, we have studied privacy attacks on the image hashing model, a representative case study of the matching task. In this Chapter, we will investigate the adversarial vulnerabilities of the classification task, and we focus on binary classification. As a case study, the representative example for binary classification we select is the forensic model (i.e., a binary classifier) for GAN-generated fake face imagery detection. Firstly, we will briefly describe some background knowledge on GAN-generated images and their forensics, and we then will summarize the challenges and our contributions in this introduction section.

Deep Neural Networks (DNNS) have been playing an overwhelming role in transforming our perspectives towards the digital world [78, 119, 151, 245]. Apart

from performing the human-aiding tasks, DNNs can also generate new digital objects/images. Recently, GAN models were used to generate photo-realistic fake face photos to easily fool human eyes [66, 108–110]. In Fig. 3.1, we show several human face images where some are captured from real person and some are generated from advanced GANs. Can you pick up GAN-generated fake face photos in Fig. 3.1? (*Answer*: Images in the first two rows are fake face images from styleGAN [109] and styleGAN2 [110], respectively; while those in the last row are real ones from the Flicker face dataset [109].)

Figure 3.1: Example images for fake face imagery detection. Question: Which images are from real persons and which ones are generated from GAN? Image samples are from [109, 110].

The widespread of such visually realistic fake face images may pose security concerns [132, 167, 228]. E.g., the Washington Post reported that some spies created social accounts with AI-generated fake face images to connect with politicians for malicious purposes [200]. Fake face photos may also be used to falsify identity information and create fake news. Therefore accurate and reliable detection of such fake face images is important.

In work [132], the authors fed features from a pretrained VGG network [209] to steganalysis classifiers [116] to identify fake face images from real ones. In work

[158, 264], the authors studied the existence of GAN fingerprints to distinguish fake images generated from different GAN models. In work [160], the authors analyzed the structure of GAN architectures and proposed to utilize saturation statistics as features, and the extracted features were classified with a support vector machine.

In work [167], a deep learning-based forensic detector was designed and gave high average accuracy, i.e., over 98% on fake face detection. The authors firstly cast color images to the residual domain with high-pass filters. Then a set of convolutional modules were applied for feature extraction and classification. More recently, in [243], the authors proposed a general fake face detector which was shown to generalize well to detect fake images from unseen GAN models. The authors in [133] investigated discernible color disparities between GAN-generated and real face photos. Then ensemble steganalysis classifiers were employed using features extracted from a third order co-occurrence matrix. Among non-deep learning based methods, the method in [133] achieved superior forensic accuracy on fake face imagery detection.

While existing forensics can successfully identify GAN-generated fake face images, there exists the concern that fake face imagery detectors might be easily bypassed by anti-forensic methods. Image anti-forensics is a countermeasure of image forensics by manipulating discernible traces to reduce the performance of forensic detectors [186, 215]. Existing anti-forensic methods often target at specific forensic detectors, e.g., JPEG compression detection [187], which are not directly applicable for our fake face detection task.

Though rarely investigated yet, studying fake face imagery anti-forensics is meaningful since it exposes possible vulnerability issues of forensic detectors. In turn, the anti-forensic study promotes researchers to propose more reliable and robust detectors, which is critical in safety-related forensic tasks. In this Chapter, our contributions are summarized as follows:

1. *We introduce adversarial attacks as an automatic anti-forensic approach for GAN-generated fake face detection. Our study shows both deep-learning and non-deep learning based methods can be vulnerable to such adversarial perturbations.*

2. *We investigated the perturbation residues of existing forensic models both*

in the RGB and YC_bC_r domains. Our analysis shows that existing gradient-based attacks display strong correlations for perturbations at RGB channels, while such correlations reduce in the YC_bC_r domain. The perturbation mainly concentrates on the Y component, leading to severe visual distortion effects.

3. *We propose a novel adversarial attack algorithm with perception constraints in the YC_bC_r domain. We allocate more perturbation for C_b and C_r channels while less for Y. More imperceptible and transferable, the proposed method significantly improves the visual quality and the attack success rate when compared with baseline attacks.*

4. *Finally, this study also reveals several interesting observations. For example, perturbations crafted for fake face images are significantly more transferable than those for real face images on all attacks we evaluated, which is worthy of further investigation.*

3.2 Related work

GAN-generated fake face imagery. GAN was formulated as a two-player game between a generator and a discriminator [66, 189]. In theory, the generator can generate visually realistic images by capturing the underlying distributions of real data when GAN reaches an equilibrium. In practice, vanilla GAN models often suffer from training instability issues. Subsequent studies then tried to stabilize training GANs (e.g. [52, 166, 268]). Specific to fake face imagery generation, Progressive GAN (PROGAN) [108] was the first GAN model to generate high resolution fake face images with relatively good visual quality. Then Karras et al. developped StyleGAN [109] which can generate human face photos with impressively realistic visual quality. Recently, StyleGAN2 [110] was proposed to achieve the state-of-the-art performance in fake face generation.

Adversarial attacks. Recent studies show DNNs are vulnerable to adversarial perturbations, termed as *adversarial examples* [46, 69, 147, 155, 222]. Adversarial examples crafted from one network can possibly fool an unknown model. This makes adversarial examples as potential threats to deployed safety-critical systems

built on DNNS. Despite active studies in the computer vision area, the existence of adversarial examples has raised relatively less attention in the forensic community [5, 157], which requires forensic detection to be both accurate and secure. For instance, a fake face image detection model is potentially meaningless if it is susceptible to certain carefully crafted perturbation.

Compared with general adversarial attacks, the anti-forensic method for GAN-generated fake face imagery detection has its unique characteristics. Generally, a higher perturbation budget indicates stronger attack ability, but degradation in visual quality. For natural-scene texture-rich images, relative higher perturbation does not seriously impair the perceptual quality. However, in fake face imagery anti-forensics, facial images are very sensitive to adversarial perturbation due to their large smooth regions. To avoid being spotted, the crafted perturbation should look ***imperceptible*** to human eyes. Otherwise, such perturbed images can be easily detected by visual sanity check.

For adversaries, another desirable property is that anti-forensic manipulations are ***transferable*** to unseen forensic models. *Transferability* means the anti-forensic perturbation designed for specific forensic models can also reduce the detectability of other unknown forensic models. This property also poses severe threats to fake fake forensic detectors.

In work [157], the authors employed existing attack methods [69, 155] to study the adversarial vulnerability of deep learning-based classifiers for camera model identification. In work [5], the authors examined adversarial attacks in the median filtering and image resizing forensic tasks, and concluded that adversarial examples are generally not transferable in image forensics. However, such conventional attack methods they used are less transferable and lead to perceptual issues in our specific anti-forensic task. Therefore, in this study we propose a novel perception-aware attack method which provides both imperceptible visual quality and higher transferability than those from the existing methods.

3.3 Method

3.3.1 The adversarial attack problem

Assume a forensic detector $f : \mathcal{D} \subseteq \mathbb{R}^d \mapsto \mathbb{R}^K$, where $\mathcal{D} = [0,255]^d$. Given a data sample $\boldsymbol{x} \in \mathbb{R}^d$, the detector correctly predicts its label as $y \in \mathcal{Y}$, i.e., $y = \underset{k=1,\cdots,K}{\arg\max}\, f_k(\boldsymbol{x})$.

The adversarial attack problem seeks an ϵ-ball bounded perturbation $||\boldsymbol{\delta}||_p \leq \epsilon$ within the vicinity of $\boldsymbol{x}$, which makes the forensic detector fail with a high probability. Here $||\cdot||_p$ denotes the ℓ_p norm constraint. Then the perturbed data $\boldsymbol{x}^{adv} := \boldsymbol{x} + \boldsymbol{\delta}$ is an adversarial example w.r.t the threat model if the following conditions are satisfied,

$$\underset{k=1,\cdots,K}{\arg\max}\ f_k(\boldsymbol{x}+\boldsymbol{\delta}) \neq y, \quad ||\boldsymbol{\delta}||_p \leq \epsilon \quad \text{and} \quad \boldsymbol{x}+\boldsymbol{\delta} \in \mathcal{D} \tag{3.1}$$

Denoting a surrogate function as $\mathcal{L}$, we define the constrained optimization problem as,

$$\underset{\boldsymbol{\delta}}{\arg\max}\ \mathcal{L}(f(\boldsymbol{x}+\boldsymbol{\delta}), y) \quad \text{s.t. } ||\boldsymbol{\delta}||_p \leq \epsilon,\ \boldsymbol{x}+\boldsymbol{\delta} \in \mathcal{D} \tag{3.2}$$

In this work, we use the ℓ_∞ norm constraint, a popular ℓ_p norm in the literature. The surrogate function $\mathcal{L}$ is selected as the binary cross entropy function in our setting.

To solve Eq.(3.2), [69] proposed the Fast Sign Gradient Method Fast Gradient Sign Method (FGSM), a one-step gradient-based perturbation, which utilizes the sign of the gradient w.r.t. the input data,

$$\boldsymbol{\delta}_{FGSM} = \epsilon \cdot \text{sign}(\nabla_{\boldsymbol{x}} \mathcal{L}(f(\boldsymbol{x}+\boldsymbol{\delta}, y)) \tag{3.3}$$

where the element-wise $\text{sign}(\cdot)$ function gives +1 for positive values, and -1 for negative values; otherwise, it gives 0.

The FGSM method was designed under the assumption that the decision boundary is linear around the input data. For neural networks with nonlinear activation function, this assumption does not hold, thus the FGSM attack generally "underfits"

the model, which compromises its attack ability. To increase the attack ability, adversaries can apply Eq.(3.3) iteratively for multiple times [155]. [46] further incorporates the momentum during the gradient update at each iteration and proposes the Momentum Iterative FGSM (MIM). We use FGSM (single-step) and MIM (multiple-step) as our baseline attacks.

3.3.2 Perturbation analysis in YCbCr domain

In this section, we investigate spatial correlations of adversarial perturbations in R, G, and B channels. We then show that for existing fake face forensic models (trained on RGB domain), with baseline attacks, the perturbation energy concentrates more in the Y component than in C_b and C_r components.

For simplicity, we analyze adversarial perturbations generated from FGSM, the single-step attack method with perturbation as the gradient (after sign). For a single pixel in an image, we denote the gradient (after sign) of R, G, B components as three random variables $\boldsymbol{S} = (s^r, s^g, s^b)^T$, where s^r, s^g, s^b follows the Bernoulli distribution. The statistical correlations of these three components are provided by the covariance matrix $\Sigma_{\boldsymbol{S}}$, which can be estimated via observations of the random variable $\boldsymbol{S}$,

$$\boldsymbol{\Sigma_S} \approx \frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{S}_i - \bar{\boldsymbol{S}}) \cdot (\boldsymbol{S}_i - \bar{\boldsymbol{S}})^T \tag{3.4}$$

where N denotes the number of observations of $\boldsymbol{S}$, and $\bar{\boldsymbol{S}}$ represents the sample mean of $\boldsymbol{S}$. The conversion from the RGB domain to the YC_bC_r domain is to perform an affine transformation,

$$\boldsymbol{S'} = \boldsymbol{A}\boldsymbol{S} + \boldsymbol{b} \tag{3.5}$$

where $\boldsymbol{S'} = (s^y, s^{Cb}, s^{Cr})^T$ denotes the transformed random variables in the YC_bC_r domain; $\boldsymbol{A}, \boldsymbol{b}$ denote respectively the transformation matrix and bias, with

$$\boldsymbol{A} = \begin{bmatrix} 0.2568 & 0.5041 & 0.0979 \\ -0.1482 & -0.2910 & 0.4392 \\ 0.4392 & -0.3678 & -0.0714 \end{bmatrix}$$

and

$$\boldsymbol{b} = (16, 128, 128)^T$$

Then we can obtain the covariance matrix of $\boldsymbol{S'}$ as,

$$\Sigma_{\boldsymbol{S'}} = \boldsymbol{A}\Sigma_{\boldsymbol{S}}\boldsymbol{A}^T \tag{3.6}$$

In Fig. 3.2, we illustrate the covariance matrices of $\boldsymbol{S}$ and $\boldsymbol{S'}$ estimated with the number of pixels as $N = 10, 10^2, 10^3$ and 10^4 on StyleGAN [109]. Clearly, s^r, s^g, s^b components are highly correlated; while the correlations reduce when we apply the YC_bC_r transform in Eq.(3.5). Also, we notice that the variances are almost identical for s^r, s^g, s^b, while the variance of s^y is significantly larger than that of s^{Cb} and s^{Cr} (i.e., around 3 times larger). It indicates that the perturbation energy concentrates more on the Y component than on C_b and C_r components.

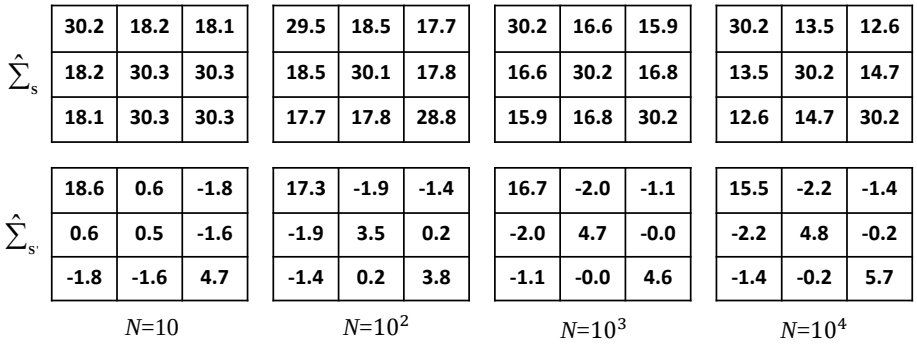


Figure 3.2: Illustration of estimated covariance matrices $\hat{\Sigma}_{\boldsymbol{S}}$ (1^{st} row) and $\hat{\Sigma}_{\boldsymbol{S'}}$ (2^{nd} row) with $N = 10, 10^2, 10^3$ and 10^4 respectively. Here $\epsilon = 5.5$.

To validate the analysis, we generate adversarial examples using FGSM and MIM, and show the histograms of perturbations in the YC_bC_r domain in Fig. 3.3. For both attacks, we observe that perturbation residues mainly cluster at ± 5.5 for Y while the perturbations peak around 0 for C_b and C_r components. We observed similar perturbation phenomena during attacking existing forensic models on StyleGAN2 [110] and ProGAN [108] datasets. Since the human visual system is more sensitive to perturbations in the Y component than in C_b and C_r components, this intuitively explains why the RGB domain attacks are prone to visual distortion.

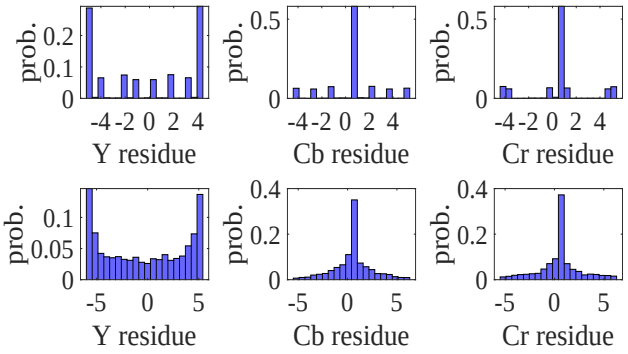


Figure 3.3: Example perturbation histograms of FGSM (1^{st} row) and MIM (2^{nd} row) attacks in the YC_bC_r domain. The histogram is generated by using 5000 adversarial samples of StyleGAN-generated fake face images. The perturbation bounds are $\epsilon = 5.5$ and $\epsilon = 6$ for FGSM and MIM, respectively.

3.3.3 Proposed adversarial attack

Based on the perturbation analysis discussed above, as an alternative to existing attacks on the RGB domain, we are motivated to perform adversarial attacks with explicit perturbation constraints in the YC_bC_r domain. By exploiting the perception characteristics, we propose to directly allocate more perturbations to C_b and C_r components than to the Y component to produce more visually pleasant adversarial examples.

Denote $\mathcal{T}$ as the transformation operator from the RGB domain to the YC_bC_r domain (see Eq.(3.5)), and $\mathcal{T}^{-1}$ as its inverse transformation back to the RGB domain. The proposed loss function is expressed as,

$$\begin{aligned}\mathcal{L}\left(f\left(\mathcal{T}^{-1}(\mathcal{T}\boldsymbol{x}+\boldsymbol{\zeta})\right), y\right) = &- y \cdot \log\left(f_y\left(\mathcal{T}^{-1}(\mathcal{T}\boldsymbol{x}+\boldsymbol{\zeta})\right)\right) \\ &- (1-y)\cdot\log\left(f_{1-y}\left(\mathcal{T}^{-1}(\mathcal{T}\boldsymbol{x}+\boldsymbol{\zeta})\right)\right)\end{aligned} \tag{3.7}$$

where $\boldsymbol{\zeta}$ denotes the perturbation that is directly optimized in the YC_bC_r domain.

Now our constrained optimization problem becomes,

$$\begin{aligned} \arg\max_{\boldsymbol{\zeta}} \; & \mathcal{L}\left(f\left(\mathcal{T}^{-1}(\mathcal{T}\boldsymbol{x}+\boldsymbol{\zeta})\right), y\right) \\ & \text{s.t. } ||\boldsymbol{\zeta}^{[c]}||_\infty \leq \epsilon^{[c]},\ c \in \{Y, C_b, C_r\}, \\ & \text{and} \quad \boldsymbol{x}+\mathcal{T}^{-1}\boldsymbol{\zeta} \in \mathcal{D} \end{aligned} \tag{3.8}$$

where $\boldsymbol{\zeta}^{[c]}$ and $\epsilon^{[c]}$ denote the constrained perturbation and its perturbation budget at channel $c, c \in \{Y, C_b, C_r\}$, respectively. To alleviate the visual distortion effects due to perturbations $\boldsymbol{\zeta}$, it is desirable to assign larger values for $\epsilon^{[C_b]}, \epsilon^{[C_r]}$ than $\epsilon^{[Y]}$. Assume that we have access to the forensic detector (or its substitute model), we can utilize the gradient-based approach to solve Eq.(3.8).

Denote any pixel in an image by $\boldsymbol{P}_{i,j} = (R(i,j), G(i,j), B(i,j))^T$, and its counterpart in the YC_bC_r domain as $\boldsymbol{P}'_{i,j} = (Y(i,j), C_b(i,j), C_r(i,j))^T$. We can propagate the gradient from the RGB to the YC_bC_r domain,

$$\begin{aligned} \nabla_{\boldsymbol{P}'_{i,j}} \mathcal{L}\left(f\left(\mathcal{T}^{-1}(\mathcal{T}\boldsymbol{x}+\boldsymbol{\zeta})\right), y\right) = (\mathbf{1} \oslash \boldsymbol{A}) \cdot \\ \nabla_{\boldsymbol{P}_{i,j}} \mathcal{L}\left(f\left(\mathcal{T}^{-1}(\mathcal{T}\boldsymbol{x}+\boldsymbol{\zeta})\right), y\right) \end{aligned} \tag{3.9}$$

where $\nabla_{\boldsymbol{P}'_{i,j}} \mathcal{L} = \left(\frac{\partial \mathcal{L}}{\partial Y(i,j)}, \frac{\partial \mathcal{L}}{\partial C_b(i,j)}, \frac{\partial \mathcal{L}}{\partial C_r(i,j)}\right)^T$ and $\nabla_{\boldsymbol{P}_{i,j}} \mathcal{L} = \left(\frac{\partial \mathcal{L}}{\partial R(i,j)}, \frac{\partial \mathcal{L}}{\partial G(i,j)}, \frac{\partial \mathcal{L}}{\partial B(i,j)}\right)^T$ denote the partial derivatives w.r.t. the loss function $\mathcal{L}(\cdot)$ in RGB and YC_bC_r domains, respectively; $\oslash$ denotes the elementwise division operation.

The flowchart of the proposed attack method is described in detail in Algorithm 1.

3.4 Experiments

3.4.1 Experimental setup

Datasets: We create face image datasets for the fake face imagery detection task: Dataset 1 and Dataset 2, respectively. Dataset 1 consists of 40,000 real face photos and 40,000 StyleGAN-generated photo-realistic facial images [109]. In Dataset 2, the fake face images are from StyleGAN2 [110]. For real or fake images in both datasets, image splits are: 30,000 images for model training, 5,000 images

Algorithm 1: The proposed algorithm of adversarial attacks in the YC_bC_r domain.

Data: A clean image $\boldsymbol{x}$ with label y, a fake-face forensic model f, channel-wise perturbation budget $\epsilon^{[c]}, c \in \{Y, C_b, C_r\}$, iteration number K and hyperparameter μ.

Result: Optimized perturbation $\boldsymbol{\zeta}$ that satisfies $\left\{\boldsymbol{\zeta} \mid \{||\boldsymbol{\zeta}^{[c]}||_\infty \leq \epsilon^{[c]}\} \cap \{\boldsymbol{x} + \mathcal{T}^{-1}\boldsymbol{\zeta} \in \mathcal{D}\}\right\}$, and the perturbed image $\boldsymbol{x}^{adv}$.

Initialize $\alpha^{[c]} = \epsilon^{[c]}/K, c \in \{Y, C_b, C_r\}$, $\boldsymbol{\zeta}_{(0)} = \mathbf{0}$, $\boldsymbol{g}'_{(0)} = \mathbf{0}$;

for $k = 0$ **to** $K - 1$ **do**

  Input $\boldsymbol{x}_{(k)}$ to the forensic model f, and compute gradients of $\boldsymbol{x}$: $\nabla_{\boldsymbol{x}_{(k)}}\mathcal{L}$;

  Compute gradients w.r.t. $\mathcal{T}\boldsymbol{x}_{(k)}$ using Eq.(3.9): $\nabla_{\mathcal{T}\boldsymbol{x}_{(k)}}\mathcal{L}$;

  Compute accumulated gradients w.r.t. $\mathcal{T}\boldsymbol{x}_{(k)}$: $\boldsymbol{g}'_{(k+1)} = \mu \cdot \boldsymbol{g}'_{(k)} + \nabla_{\mathcal{T}\boldsymbol{x}_{(k)}}\mathcal{L}/||\nabla_{\mathcal{T}\boldsymbol{x}_{(k)}}\mathcal{L}||_1$;

  Compute perturbation $\boldsymbol{\zeta}_{(k+1)}$: $\boldsymbol{\zeta}^{[c]}_{(k+1)} = \boldsymbol{\zeta}^{[c]}_{(k)} + \alpha^{[c]} \cdot \text{sign}\left(\boldsymbol{g}'_{(k+1)}\right), c \in \{Y, C_b, C_r\}$;

  Project $\boldsymbol{\zeta}_{(k+1)}$ within the ϵ-ball: $\boldsymbol{\zeta}_{(k+1)} = \max\left(\min\left(\boldsymbol{\zeta}_{(k+1)}, \epsilon\right), -\epsilon\right)$;

  Update adversarial example $\boldsymbol{x}_{(k+1)}$: $\boldsymbol{x}_{(k+1)} = \boldsymbol{x} + \mathcal{T}^{-1}\boldsymbol{\zeta}_{(k+1)}$;

  Project $\boldsymbol{x}_{(k+1)}$ within the feasible set $\mathcal{D}$: $\boldsymbol{x}_{(k+1)} = \text{Proj}_{\mathcal{D}}\left(\boldsymbol{x}_{(k+1)}\right)$;

end

Return: Optimized perturbation $\boldsymbol{\zeta} = \boldsymbol{\zeta}_{(K)}$ and the perturbed image $\boldsymbol{x}^{adv} = \boldsymbol{x}_{(K)}$.

for validation and the rest 5,000 images for test. To reduce the computational complexity, all images are resized to 128×128.

Models: We study seven effective fake face identification models [119, 133, 167, 189, 199, 209, 243], which are trained from scratch on the face datasets described above. For deep learning-based models (trained on RGB domain), the hyperparameters are as follows: the learning rate is set as 10^{-4} with weight decay 5×10^{-4}, the batchsize is selected as 64, and the number of epochs equals 20 with early stopping. For non-deep learning based fake-face detection models [133, 160], we consider the state-of-the-art method proposed in [133]. For convenient expression, we denote the deep-learning based forensic models as $m_i, i = 1, 2, \cdots, 6$ for six different architectures from work [119, 167, 189, 199, 209, 243] used in the literature respectively. We denote the selected non-deep learning forensic model as "*NDL*" [133].

To make sure that the forensic models work well (e.g., detection accuracy $\geq 90\%$), we adopt the true positive rate True Positive Rate (TPR) and the true negative rate True Negative Rate (TNR) as their performance measures, where TPR

and TNR are defined as:

$$TPR = \frac{TP}{TP+FN}, \quad TNR = \frac{TN}{TN+FP} \tag{3.10}$$

where TP, TN, FP and FN denote the numbers of correctly identified fake face images, correctly detected real face images, misclassified real face samples and misclassified fake face images, respectively. A good detector provides high TPR and TNR simultaneously. After proper training, all forensic models achieve high TPR and TNR values on both datasets, as shown in Table 3.1. We have released these pretrained models to the public.

Table 3.1: Pretrained forensic models we evaluated and their performances measured by TPR and TNR on Dataset 1 and Dataset 2, respectively.

Datasets	models	m_1	m_2	m_3	m_4	m_5	m_6	NDL
Dataset 1	TPR (%)	98.6	94.1	91.4	95.8	90.8	99.6	98.6
	TNR (%)	98.7	96.9	94.6	98.0	94.4	99.9	98.7
Dataset 2	TPR (%)	98.8	99.0	98.1	98.5	96.2	99.9	99.5
	TNR (%)	99.2	99.4	98.5	98.5	97.2	99.9	99.4

Parameters: In the following experiments, following the baseline MIM method [46], for iterative attacks, we set the iteration number K as 10, and the momentum decay factor μ as 1. The perturbation bound ϵ is often chosen as 16. However, this perturbation bound is generally too large in the fake face anti-forensic tasks since it can severely degrade visual quality. To have a good trade-off between visual quality and attack success rate, we set lower perturbation bound, e.g., on Dataset 1 we use ϵ as 5.5 and 6 for FGSM and MIM attacks, respectively. For the proposed method, we set larger values for $\epsilon^{[C_b]}$ and $\epsilon^{[C_r]}$ than $\epsilon^{[Y]}$ for better visual imperceptibility.

3.4.2 Attack success rate comparison

The attack success rate Attack Success Rate (ASR) is defined as the accuracy reduction of forensic models after applying adversarial attacks. Concretely, for the fake face detection problem, denote TPR' as the true positive rates after the attack on fake face images. Then $ASR^{[p]}$ on this given fake face image subset (5,000 images

in total) is calculated as,

$$ASR^{[p]} = TPR - TPR' \tag{3.11}$$

Similarly, we can define the attack success rate on real images as $ASR^{[n]} = TNR - TNR'$, where TNR' denotes the true negative rates after the attack on the real face image subset. Clearly, the stronger the adversary, the higher the attack success rates.

For the visual quality evaluation, we use three popular image quality assessment Image Quality Assessment (IQA) metrics: the Naturalness Image Quality Evaluator (NIQE) [165], a no-reference IQA to evaluate the naturalness of images (lower indices indicate more natural visual quality); the Learned Perceptual Image Patch Similarity (LPIPS) [271], a DL-based IQA for semantic similarity measurement (lower values suggest closer semantic similarity); and the Feature Similarity index ($FSIM_c$) [270], a full-reference IQA based on human visual system (normalized within $[0, 1]$, the higher the index, the better the visual quality).

As an adversary, we focus on attacking fake face images whose reliable detection is vital for forensic models. First, assume we have full access to m_1, then we can craft adversarial perturbations based on this model. On Dataset 1, ϵ are set as 5.5 and 6 for FGSM and MIM attacks, respectively. To have comparable average ASRs, the proposed method adopts $\epsilon^{[Y]} = 2.5$, $\epsilon^{[C_b]} = 6$, $\epsilon^{[C_r]} = 6$. Similarly, on Dataset 2, we use ϵ as 6 and 7.5 for FGSM and MIM; and $\epsilon^{[Y]} = 2$, $\epsilon^{[C_b]} = 6$, $\epsilon^{[C_r]} = 6$ for the proposed method for a fair comparison. On both datasets, the comparison results are reported in Table 3.2. The effects of adversarial perturbations crafted from other deep learning models are investigated in Section 3.4.4.

In Table 3.2, we can see that with comparable average $ASR^{[p]}$ on both datasets, the perceptual quality of the proposed method has been improved over FGSM and MIM attacks by a large margin quantitatively measured by three IQA metrics. Particularly on Dataset 2, the proposed method achieves considerably improved visual performance and 9.7% and 7.3% higher attack success rates on average on fake face imagery antiforensics.

Table 3.2: Performance comparisons of the attack success rate (%) and the visual quality when applying FGSM, MIM and the proposed method on fake face images from Dataset 1 and Dataset 2. The source model is m_1. On Dataset 1, ϵ is 5.5, 6 for FGSM and MIM attacks respectively; and on Dataset 2, ϵ is 6, 7.5 for FGSM and MIM, respectively. For the proposed method, $\epsilon^{[c]}$ are $2.5/6/6$ on Dataset 1 and $2/6/6$ on Dataset 2. The best performances are marked in bold.

Datasets	Attack	m_1	m_2	m_3	m_4	m_5	m_6	NDL	avg. $ASR^{[p]}$	NIQE	LPIPS	FSIM_c
Dataset 1	FGSM	98.6	90.9	73.4	58.9	20.6	72.5	97.6	73.2	1.188	0.026	0.955
	MIM	98.6	**91.6**	77.4	63.4	21.1	**81.2**	98.6	76.0	1.032	0.028	0.952
	Prop.	**98.6**	91.2	**83.3**	**78.3**	**40.9**	63.3	**98.6**	**79.2**	**0.798**	**0.020**	**0.984**
Dataset 2	FGSM	98.8	98.9	84.2	94.6	5.0	2.6	62.5	63.8	1.728	0.034	0.969
	MIM	98.8	99.0	97.5	97.5	6.4	**23.3**	41.0	66.2	1.660	0.036	0.965
	Prop.	**98.8**	**99.0**	**98.1**	**98.5**	**12.9**	14.4	**92.6**	**73.5**	**1.029**	**0.018**	**0.992**

3.4.3 Visual quality comparison

As shown in Table 3.2, when compared with baseline attacks, quantitatively, the proposed method has much improved IQA indices measured by NIQE, LPIPS and FSIM_c, i.e., we have higher fidelity with cleaner images either semantically or visually when using the proposed attack algorithm.

In Fig. 3.4, we show several perturbed fake face image examples from Dataset 1. The first row shows clean images, while the rest three rows display their perturbed versions using FGSM, MIM and the proposed method, respectively. By zooming in Fig.3.4, we can easily spot texture-like visual distortions on FGSM and MIM attacks, both in facial regions and background. By contrast, adversarial images from the proposed method still maintain smooth and appear more natural and more ***imperceptible***, compared with the clean images. More comparison examples can be found in the project website.

Moreover, we conduct the human subjective preference study to further validate the visual/quantitative comparison results. For each dataset, we randomly choose 50 comparison pairs: clean image and their adversarial version generated by FGSM, MIM and the proposed method, respectively. For each surveyed pair, we prepare two questions: (a) Is it hard to tell which perturbed one is the "cleanest"? If yes, we proceed to the next pair; otherwise we ask the interviewer to (b) Choose the "cleanest" one from three adversarial images. Overall, on each dataset, we

received 500 answers (10 volunteers for each dataset), and our subjective study shows that all interviewers perceive adversarial images from the proposed method as the best/cleanest one. With the human preference survey, we can safely conclude that the proposed method has indeed considerably improved the perceptual quality of adversarial images.

Figure 3.4: Examples of fake face images for visual quality comparisons on FGSM, MIM and the proposed method. For FGSM and MIM, ϵ are 5.5 and 6 respectively; for the proposed method, $\epsilon^{[c]}$ are 2.5/6/6 for Y, C_b, C_r channels. We recommend to zoom in the digital images for better visual comparison.

3.4.4 Adversarial transferability

In Fig.3.6, on Dataset 1 and Dataset 2, we visualize the transfer matrices of adversarial examples crafted from different forensic models using FGSM, MIM and the proposed method, respectively. In each matrix, each row denotes the same source model to craft adversarial examples and each column represents a target model on which to be evaluated.

For each source model, the proposed method achieves higher average attack

Figure 3.5: Examples of fake face images for visual quality comparisons on FGSM, MIM and the proposed method. For FGSM and MIM, ϵ are 6 and 7.5 respectively; for the proposed method, $\epsilon^{[c]}$ are $2/6/6$ for Y, C_b, C_r channels. We recommend to zoom in the digital images for better visual comparison.

success rates over FGSM and MIM on both datasets. Besides, we have several interesting observations. First, the NDL models are also likely to be fooled in the presence of antiforensic perturbations. Particularly on Dataset 1, NDL models almost completely fail. This indicates that even non-deep learning based forensic models can be vulnerable to adversarial perturbations crafted from deep forensic models, which *necessitates further security investigation into conventional forensic models* in the presence of adversarial attacks. Second, the adversarial transferability can be quite asymmetric between different forensic models. For instance, adversarial perturbations crafted from m_1 effectively transfer to m_4 for all three attacks. However, adversarial examples created from the source model m_4 hardly transfer to m_1. This intriguing phenomenon might be related with the sophisticated decision landscapes of DL models (which differ in network modules or depth). For instance, as shown in the transferability heatmap (i.e., Fig.3.6), with source model as m_1,

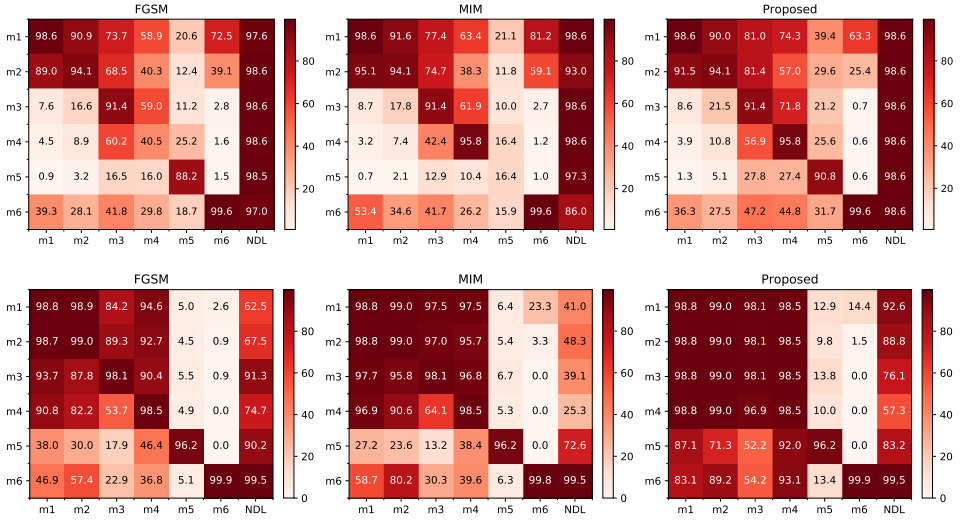


Figure 3.6: Comparisons of adversarial transferability of FGSM, MIM and the proposed method on fake face image forensic models on Dataset 1 (the 1^{st} row) and Dataset 2 (the 2^{nd} row), respectively.

we can see that the ASRs on m_5 or m_6 are lower than the rest forensic models. This phenomenon is mainly due to the differences in the network architecture (i.e. module components or network depth etc), which can influence the attack transferability. Generally, a source model transfers more easily to a target model when they adopt a similar architecture; or vice versa. To be specific, m_5 is a lightweight network architecture particularly designed for the mobile setting. It uses some specially designed modules (e.g. inverted residual blocks) while m1 does not adopt such sophisticated modules. As a result, the ASR on m_5 is lower than that of the rest NDL-based models. As for m_6, it adopts the Resnet50 as its backbone (50 layers), while the rest architectures are within 10 layers. The much differences in layers can also make the adversarial examples harder to transfer between different models. The asymmetry in network models can also account for the differences in the averaged ASRs when we choose different forensic model as the source model.

We also observe that by the careful selection of source forensic models, *adversaries can build more transferable attacks* with the same attack method. To demonstrate further in this direction, in our preliminary study, we ensemble different forensic models to compose new source models and evaluate their attack-

ing performances. As an example, by using the grid search, we combine three forensic models as $m_{ens(i,j,k)}$ with $i, j, k \in \{1, \cdots, 6\}$. Then we fuse their scores together with equal weights and generate adversarial perturbations with the proposed method. The average ASRs of some ensemble source models are reported in Table 3.3. Though it remains unclear on the optimal model ensemble selection (i.e., in terms of the model number and weights), we find some combinations indeed generate more transferable attacks, e.g., the ensemble model $m_{ens(1,4,6)}$ on Dataset 1 and $m_{ens(3,4,5)}$ on Dataset 2. We will investigate further on this phenomenon in the future (e.g., by varying different datasets and forensic models).

Table 3.3: Average $ASR^{[p]}$ results (%) from example combinations of the source models on Dataset 1 (#1) and Dataset 2 (#2).

Source model	$m_{ens(1,2,5)}$	$m_{ens(1,4,5)}$	$m_{ens(1,4,6)}$	$m_{ens(2,3,4)}$	$m_{ens(3,4,5)}$
avg. $ASR^{[p]}$ (#1)	78.5	78.5	**82.2**	72.4	57.1
avg. $ASR^{[p]}$ (#2)	83.3	83.2	72.5	72.5	**84.3**

3.4.5 Perturbation residues

In Fig. 3.7, we show the perturbations generated from the proposed method in the YC_bC_r domain. The 1^{st} and 2^{nd} rows illustrate the perturbation histograms on Dataset 1 and Dataset 2 with parameters the same as in Table 3.2. Compared with Fig.3.3, perturbations in the Y component approach ± 2.5 on Dataset 1 (± 2 on Dataset 2). By contrast, perturbations in C_b and C_r components spread away from 0 and concentrate around ± 6. This observation on perturbation residues aligns well with our expectation, which possibly explains the more imperceptible image quality of the proposed attack method. Besides the two datasets as reported, we also experimented on ProGAN [108] and StyleGAN (with image resolution as 512×512), we have similar conclusion: the proposed method also much improves the visual quality over baseline attacks.

3.4.6 Comparison on different parameters

In Fig.3.8, we show the averaged attack success rates and perceptual quality with different choices of ϵ for FGSM and MIM attacks on the fake face image subset where the source model is m_1. Generally, as the perturbation bound ϵ increases, ASR

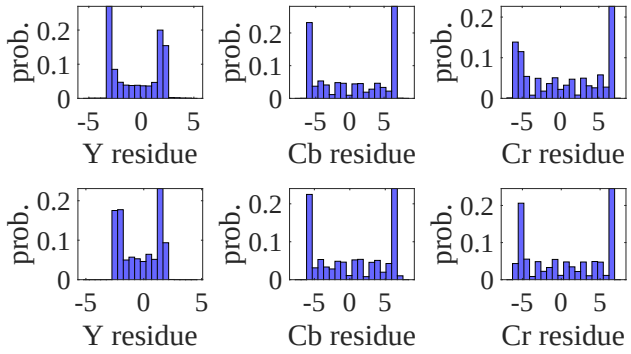


Figure 3.7: Example perturbation histograms of the proposed method in the YC_bC_r domain on two datasets: $\epsilon^{[Y]} = 2.5, \epsilon^{[C_b]} = 6, \epsilon^{[C_r]} = 6$ on Dataset 1; and $\epsilon^{[Y]} = 2, \epsilon^{[C_b]} = 6, \epsilon^{[C_r]} = 6$ on Dataset 2. The histogram is generated using 5000 adversarial samples of fake face images.

increases for both attacks at the cost of visual degradation. To keep comparably high visual quality, e.g., setting FSIM_c as 0.984, the averaged ASR of FGSM and MIM are only about 49.0%, 46.7%. However, the index of the proposed method is 79.2% ($\epsilon^{[c]} = 2.5/6/6$) , which is **30.2**% and **32.5**% higher than FGSM and MIM. Similarly, we can compute the approximate ASR improvement as **35.1**% and **28.5**% on FGSM and MIM on Dataset 2 when setting FSIM_c as 0.992. This result further convincingly shows the superiority of the proposed method over baseline attacks.

3.4.7 Experimental results with larger image resolution

We also experimented on StyleGAN with image size 512x512. To be specific, the dataset has the same train/validation/test split as that for StyleGAN (128x128). In the training dataset, we have 30,000 FFHQ (real face images) and 30,000 StyleGAN-generated fake face images. In the validation (or test) dataset, we have another 5000 FFHQ and 5000 StyleGAN-generated fake face images. All images have the same image resolution as 512x512. To obtain satisfactory forensic performances, we trained each of the six deep learning-based forensic models for 20 epochs with early stopping. On the clean test dataset, the performance of each model is reported in Table 3.4 as follows.

As shown in Table 3.4, all forensic models achieve good performances on the StyleGAN (512x512) dataset. Compared with StyleGAN (128x128) dataset, all

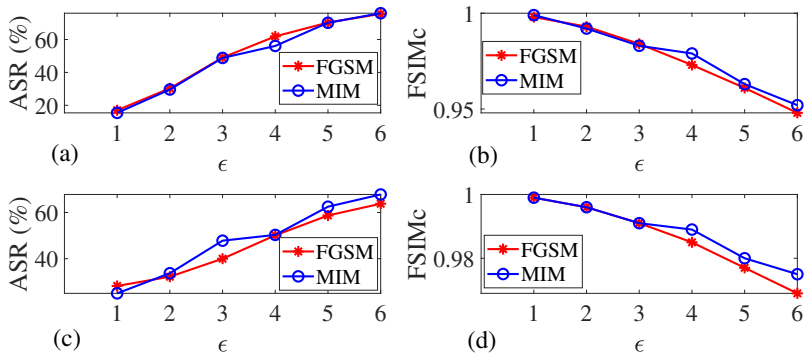


Figure 3.8: Illustration of the averaged attack success rate and visual quality with different ϵ values for FGSM and MIM attacks on Dataset 1 (1^{st} row) and Dataset 2 (2^{nd} row). (a) and (c): $ASR^{[p]}$ vs. ϵ; (b) and (d): FSIM_c vs. ϵ.

Table 3.4: Pretrained forensic models we evaluated and their performances measured by TPR and TNR on StyleGAN (512x512).

models	m_1	m_2	m_3	m_4	m_5	m_6	NDL
TPR (%)	99.9	99.9	100	99.9	99.7	100	99.9
TNR (%)	100	100	100	100	99.9	100	99.8

forensic models yield higher detection accuracy. This is because there are indeed more forensic traces to be made use of on larger images. In other words, the forensic task itself is more challenging to detect low-resolution fake images generated from the advanced GANs (e.g. StyleGAN/128x128).

As some examples, we evaluate the baselines and the proposed method with source model as m_1. First, we use a similar set of parameters (i.e. perturbation bound) as in StyleGAN (128x128), and the comparison results are shown in Table 3.5. We observe the adversarial images (512x512) from FGSM, MIM and the proposed method appear to have improved visual metrics than those from 128x128 images. Yet we observe their averaged ASRs also decreased, correspondingly. This is probably because, on larger images, the computed perturbations on some pixels equal zero (or almost zero but thresholded as zero). Despite that, the proposed method still achieves the best visual quality given comparable (or higher) averaged

attack success rates.

Table 3.5: Performance comparisons of the attack success rate (%) and the visual quality when applying FGSM ($\epsilon = 5.5$), MIM ($\epsilon = 8.0$) and the proposed method ($\epsilon^{[c]} = 2/6/6$) on fake face images from StyleGAN (512x512). The source model is m_1. The best performances are marked in bold.

Attack	m_1	m_2	m_3	m_4	m_5	m_6	NDL	avg. $ASR^{[p]}$	NIQE	LPIPS	$FSIM_c$
FGSM	99.9	0	5.6	99.9	10.1	0	**98.7**	44.9	2.124	0.043	0.995
MIM	99.9	99.9	16.5	99.9	23.3	0	97.3	62.4	1.973	0.050	0.995
Prop.	**99.9**	**99.7**	**23.7**	**99.9**	**24.1**	**0.1**	97.9	**63.6**	**1.279**	**0.033**	**0.999**

Table 3.6: Performance comparisons of the attack success rate (%) and the visual quality when applying FGSM ($\epsilon = 16.0$), MIM ($\epsilon = 16.0$) and the proposed method ($\epsilon^{[c]} = 5/12/12$) on fake face images from StyleGAN (512x512). The source model is m_1. The best performances are marked in bold.

Attack	m_1	m_2	m_3	m_4	m_5	m_6	NDL	avg. $ASR^{[p]}$	NIQE	LPIPS	$FSIM_c$
FGSM	99.9	99.9	11.7	99.6	**96.7**	**5.6**	**99.6**	73.3	3.179	0.200	0.966
MIM	99.9	99.9	19.8	99.9	86.3	1.2	98.7	72.2	3.179	0.152	0.980
Prop.	**99.9**	**99.9**	**34.6**	**99.9**	94.3	2.0	98.5	**75.6**	**2.557**	**0.134**	**0.993**

In addition, we experimented with larger perturbation bounds on StyleGAN (512x512), and we report the comparison results as in Table 3.6. As shown in Table 3.6, the averaged ASRs are largely improved for each of the three attacks at the expense of reduced visual quality. Nevertheless, we also conclude that, compared with baselines, the proposed method can still achieve much better visual quality given comparable (or higher) averaged attack success rates. We have shown some comparison examples in Fig. 3.9, which show superiority of our method.

3.5 Discussion

3.5.1 Attacking real face images

Although attacking fake face images poses more threats on forensic models, we also report the attack success rates $ASR^{[n]}$ on the real face image subset (5000 images in total) [109]. Consistent with the conclusion on fake face images, the proposed

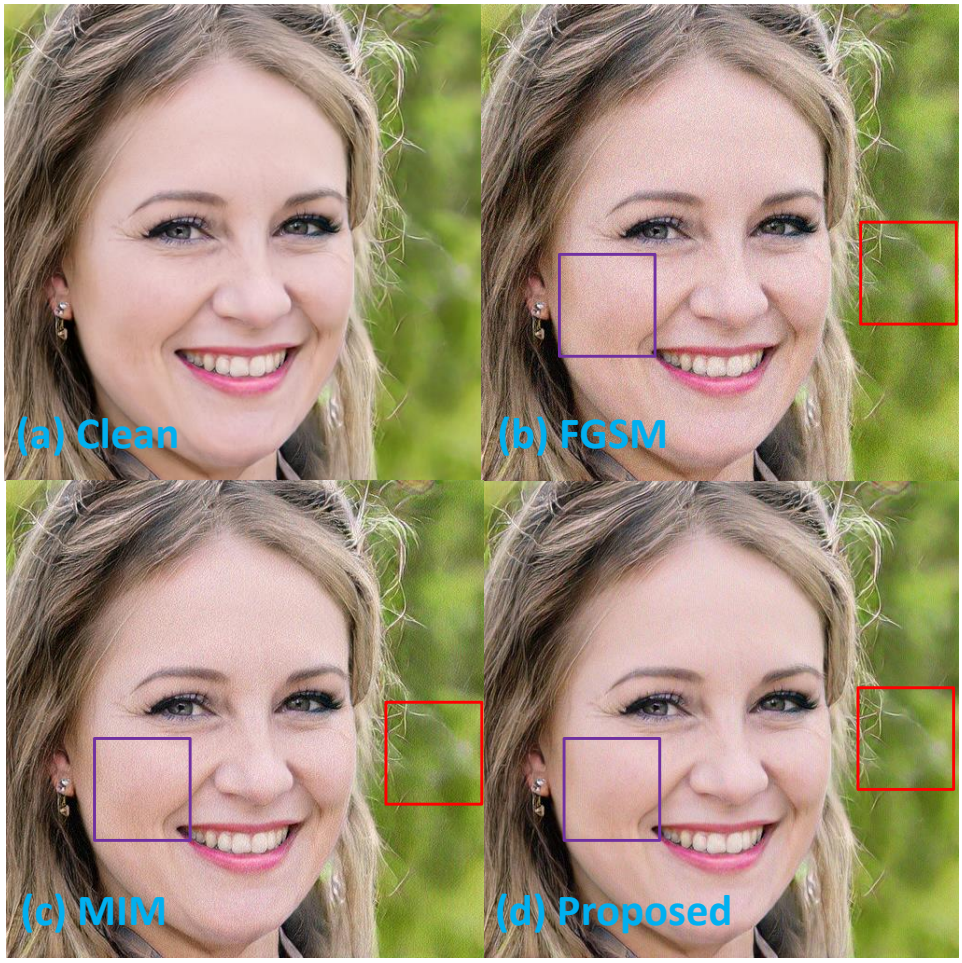


Figure 3.9: Examples of fake face images for visual quality comparisons on FGSM, MIM and the proposed method on StyleGAN (512x512). For FGSM and MIM, ϵ both equal to 16; for the proposed method, $\epsilon^{[c]}$ are $5/12/12$ for Y, C_b, C_r channels. We recommend to zoom in the digital images for better visual comparison.

method achieves the highest averaged $ASR^{[n]}$ and FSIM_c compared with FGSM and MIM attacks. Interestingly, we observe sharper degradation in attack success rates for each attack method on the real face images subset than those on fake face images. Particularly, the adversarial perturbations may fail completely for the non-deep learning based method [133]. We will further explore this phenomenon in the future work.

Table 3.7: The comparisons of the attack success rate (%) and visual quality between FGSM, MIM and the proposed method on real face images. The source model is m_1. ϵ is 5.5 for FGSM and 6 for MIM; $\epsilon^{[c]}$ are $4/7/7$ for the proposed method for Y, C_b, C_r channels. The best performances are marked in bold.

Attack	m_1	m_2	m_3	m_4	m_5	m_6	NDL	avg. $ASR^{[n]}$	FSIM_c
FGSM	98.7	75.4	20.1	10.0	4.6	48.8	0	41.8	0.955
MIM	98.7	90.8	43.5	**20.6**	7.6	81.9	0	52.2	0.955
Prop.	**98.7**	**94.4**	**47.3**	18.0	**11.3**	**88.9**	0	**53.9**	**0.965**

3.5.2 Attacks in HSV domain

In addition to adversarial attacks in the YC_bC_r domain, we also explored attacks in the HSV domain, since recent study shows relatively large discriminative statistics in the HSV domain for fake face forensics. Our preliminary study shows the inferior performance of HSV than that in the YC_bC_r domain. One possible explanation is the following: There does not exist a clear relationship between HSV channels and the human visual system, thus making it challenging to find adversarial examples both with high attack success rates and imperceptible visual quality.

3.6 Conclusion

In this Chapter, we study adversarial vulnerabilities of the classification task, and we focus on imperceptible anti-forensics on GAN-generated fake face imagery detection (i.e., a representative binary classification task) based on adversarial attacks. For existing attacks, our analysis on perturbation residues shows a significantly reduced perturbation correlation in the YC_bC_r channels when compared with RGB

channels, and these perturbations concentrate more on the Y channel than on C_b and C_r channels. Such perturbations can severely degrade the perceptual quality of facial images which have large smooth regions. Thus it makes existing attacks ineffective as a meaningful anti-forensic method. Considering the perception constraint, we propose a novel adversarial attack method that is better suitable for fake face imagery anti-forensics. Specifically, we allocate larger perturbation to C_b and C_r channels which are less sensitive to perception distortion. Simple yet effective, the proposed method achieves both higher adversarial transferability and significant improvement in visual quality when compared with baseline attacks. Moreover, we observe that the proposed method can also fool non-deep learning based forensic detectors with a high attack success rate. This study raises security concerns of existing fake face forensic methods.

In addition to fake face imagery anti-forensics, we believe all safety-critical forensic models need to be evaluated against such anti-forensics based on adversarial attacks. *More imperceptible* and *transferable*, we hope the proposed anti-forensic algorithm can be a good candidate to evaluate adversarial vulnerability of forensic models. In the future, we will further explore the anti-forensic feasibility in related forensic tasks, and develop improved algorithms to counter such anti-forensics.

Chapter 4

A Case Study of Multiclass Classification Task: Structure-Aware Imperceptible Black-box Adversarial Attacks on Image Classification

4.1 Introduction

In Chapter 3, we consider the binary classification task and study adversarial attacks on GAN-generated fake face imagery forensics (binary classifiers). The anti-forensically manipulated face images have high visual quality yet maintain high attack transferability. In this Chapter, we consider the multi-class classification task. As a representative case study, we study adversarial attacks on multi-class classifiers on natural images, a more general category of models than GAN-generated fake face imagery detectors. In this Introduction section, we will describe some background knowledge on our attacks, present challenges, and summarize our contributions.

Deep Neural Networks (DNNS) have achieved significant progress in a wide

range of machine learning tasks [78, 80, 119, 191, 241, 245]. However, their robustness has been greatly challenged by the existence of adversarial examples, where carefully perturbed images (as the inputs) can easily fool deep neural networks. Since Szegedy et al. [221] first reported adversarial examples, there have been intensive studies on the effectiveness of adversarial examples [23, 47, 55, 69, 156, 180, 188, 258].

In practice, a valid adversarial example satisfies two constraints: a) *high attack success rate*, i.e., adversarial examples can fool the target models with a high attack success rate; and b) *high perceptual quality*, i.e., adversarial examples are semantically meaningful, which indicates the image content is preserved and the image perceptual quality is as naturally-looking as possible.

White-box adversarial attack methods [55, 69, 156] can easily generate valid adversarial examples satisfying the above two constraints, because the adversary has full knowledge of the deployed model. However, to meet these constraints is much more challenging for black-box attacks [47, 258]. For example, [258], one of the latest attacks with high attack success rates, requires relatively large perturbations, which can generally degrade the perceptual quality of the generated adversarial examples. For example in Fig. 4.1, we depict an adversarial example with perturbation generated by [258], which displays unpleasant or unnatural visual artifacts. Despite those adversarial examples with poor visual qualities remain fooling the model, their threats to certain practical deployed systems (e.g., deepfake forensics [135]) can be largely compromised, because indeed they break the 'imperceptibility' property of adversarial attacks and can be easily spotted and filtered out by sanity checks. As a result, a key problem we need to solve for black-box adversarial attacks is *whether it is possible to keep a high attack success rate while preserving a naturally-looking visual quality?*

According to studies on the human cognition system, we realize that the essence of visual degradation issues is the identical and independent perturbation bound for each pixel. More specifically, such identical and independent perturbation bound is incompatible with the human visual system, which is highly sensitive to the structural information in scene perception [32, 263]. In detail, structural representations can be described by edge, texture and luminance contrast by extracting oriented gradients and relative intensity from neighboring pixels in the spatial domain [101].

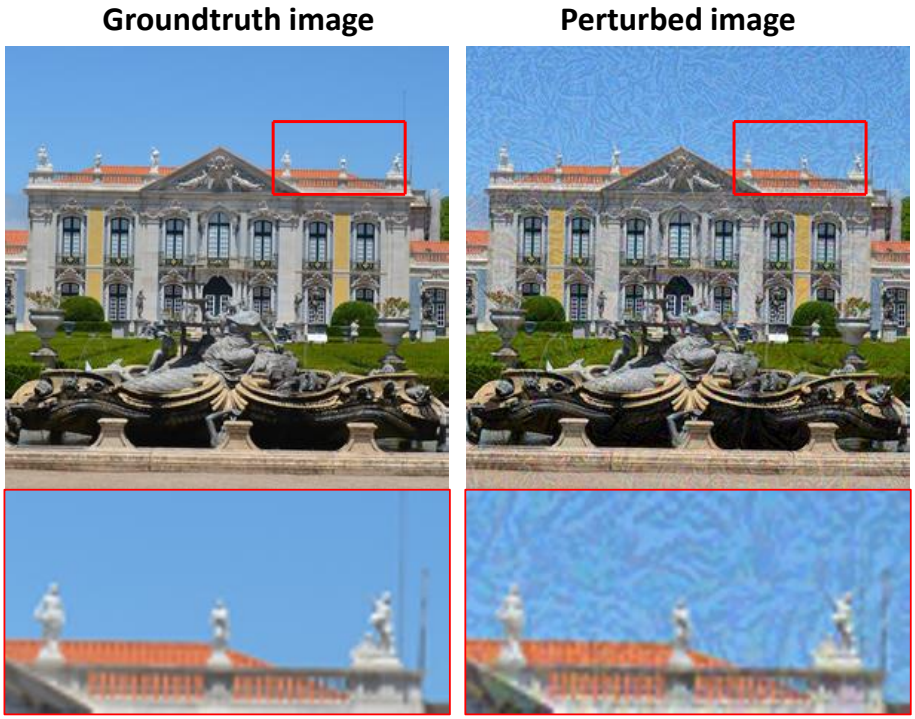


Figure 4.1: Illustration of a visually degraded adversarial example. Left: the groundtruth image; right: the perturbed image with adversarial perturbations generated by [258]. By zooming into the image patch (i.e, the red box), we can clearly notice the visual artifacts introduced by adversarial perturbations.

Moreover, visual frequency sensitivity can be integrated into constructing visual descriptors in the frequency domain [138, 273]. Therefore, uniform distortions in previous adversarial attack studies are not aligned well with the human visual system. The visual quality issue is not obvious for white-box attacks because the perturbation bound can be very small due to the fully known information. However, we have to solve the visual quality issue in black-box attacks.

To generate black-box adversarial examples by considering the human perception behavior is very challenging. Firstly, to replace the uniform perturbation, we need a new type of distortion metric to represent the structural properties of images. In this Chapter, we incorporate the results from psychological studies. The structure-aware image-dependent perceptual models [32] can identify which

regions the visual systems pay more attention and which regions is more likely to be ignored. These models have been applied in the fields of image compression [101] and video coding [263], where higher compression rates are applied on the unnoticeable regions, but lower compression rates or even not to compress on the noticeable regions. We propose to leverage these perceptual models on setting a structure-aware adversarial attack. More specifically, we allow higher perturbations on perceptually insignificant regions, while assigning lower or no perturbation on significant regions.

Secondly, only considering perturbations in the spatial domain is not enough, because perceptual systems are closely related to frequency selectivity [138]. Although there exist frequency perceptual models to quantitatively measure frequency sensitivity (e.g., [101, 273]), it is nontrivial to incorporate frequency visual models to the adversarial attack setting. To leverage the frequency perceptual models, we propose to directly add adversarial perturbations in the frequency domain, i.e., we formulate a novel adversarial attack objective function in the frequency domain with the frequency sensitivity constraint, then frequency perturbation is conducted with gradients derived for each frequency sub-band.

Thirdly, there is always a trade-off between the attack success rate and perceptual quality [24]. Simply achieving imperceptible perturbations alone is not enough, while we still need to keep a high success rate in the black-box attacks. In this Chapter, we carefully select the structure-aware perceptual incorporation strategy to make them independent of the existing gradient-based attack algorithms. As a result, we can leverage the state-of-art gradient estimation methods, while constrain the perturbation setting based on the perceptual models.

In this Chapter, we summarize our major contributions as follows:

1. *We design a framework to generate structure-aware distortions in adversarial attacks, and apply it on black-box adversarial attacks to preserve a naturally-looking visual quality while keeping a high attack success rate. Since the structure-aware strategy is independent of the gradient estimation, this framework can be generally extended to any gradient-based adversarial attack regardless of the white-box or black-box setting.*

2. *Besides the spatial structure-aware perturbations, we propose to incorporate*

the frequency perceptual models in the adversarial perturbation generation and we develop a novel structure-aware attack approach by adding adversarial perturbations in the frequency domain.

3. *Experiments demonstrate that, with the comparable attack success rate, the proposed methods have significant perceptual improvements when compared with the baseline attacks. Meanwhile, with the comparable perceptual quality, we also observe the improved attack success rate over the baseline attacks.*

4.2 Background

The existence of adversarial examples poses severe threats to deep learning models. A wide range of studies have been investigated to generate adversarial examples to fool neural networks with a high probability [23, 47, 55, 69, 125, 156, 180, 188, 258]. However, all of these studies neglected the perceptual quality evaluation on adversarial examples.

Meanwhile, limited attention has been paid to generating adversarial examples with high perceptual quality. Luo et al. introduced an overall noise sensitivity measure based on noise variance estimation for white-box attacks [152]. Croce et al. introduced a sparse ℓ_0 ball constraint to the query-limited attack and optimized the perturbation with local search [34]. The sparse perturbations are assigned to sparse regions with high variances to reduce visual distortions. However, neither of them is aligned with human visual systems, because some complicated structures (e.g., textures, edges, luminance contrast) or frequency response of an image are not explicitly modeled.

Furthermore, the idea of incorporating psychological studies [32, 263] is inspired by related works from image compression [101] and video coding [263]. For instance, in video coding [263], the perceptual-model incorporated codecs achieve both high perceptual fidelity and a high compression rate.

4.2.1 Adversarial attack models

Given a clean image $\boldsymbol{x}$, an image classifier f_θ predicts its label as y, i.e., $f_\theta(\boldsymbol{x}) = y$. Conventionally, a non-targeted adversarial example $\boldsymbol{x}^*$ can be formally defined as

$$f_\theta(\boldsymbol{x}^*) \neq y, \quad \text{s.t. } \|\boldsymbol{x}^* - \boldsymbol{x}\|_p \leq \epsilon \tag{4.1}$$

By definition, an adversarial example $\boldsymbol{x}^*$ is bounded within the ϵ ball of $\boldsymbol{x}$, with distance measured by the ℓ_p norm.

With a higher ϵ factor, the attack methods produce relatively high attack success rate at the expense of possibly severely degraded perceptual quality. The key issue lies in the fact that the distortion criterion treats each pixel independently and assigns a uniform bound with each pixel. However, human eyes mainly perceive images using local and regional statistics. Therefore, the tolerable distortion level should be different from pixel to pixel due to their different structure information defined by neighboring regions [101, 138, 263].

4.3 Perceptual models

The understanding of the human visual system is essential to generate high quality adversarial examples. We employ perceptual models to guide adversarial example generation. Perceptual models are developed over the years based on the property of human visual system over scene perception. Psychovisual study reveals that visual sensitivity relies on structural information rather than value changes at a single pixel [32, 101]. A common paradigm of perceptual models in image processing is the just-noticeable difference Just Noticeable Difference (JND) model, which was originally derived for image compression [101].

In JND models, the structure and local statistics are generally described by luminance sensitivity, contrast masking and frequency masking effects [138]. There are various types of JND models in the literature, e.g., the spatial domain JND [31] and the frequency domain JND [32, 85, 273]. Based on JND models, we can estimate the maximal perturbation bounds within the imperceptibility constraint. It also indicates the perceptual importance on each pixel, and so we leverage it to design non-uniform distortions. We are motivated to incorporate the perceptual-

model based constraint to generate adversarial examples with high visual quality. We briefly describe two JND models we adopt in the following sections.

4.3.1 Spatial JND model

In the spatial domain, we apply a basic JND model, which considers the image structure that consists of textures and local luminance distribution [263]. The JND profile is obtained by calculating the dominant values of its two structural components. Specifically, the spatial JND for a grayscale image, denoted by JND_s, is defined as follows:

$$\mathcal{JND}_s = \mathcal{TM} + \mathcal{LA} - C \cdot \min\{\mathcal{TM}, \mathcal{LA}\} \tag{4.2}$$

where $\mathcal{TM}$ represents the texture masking, $\mathcal{LA}$ represents the luminance adaptation, and $C \in (0,1)$ measures the overlapping effect between the texture masking and luminance adaptation effects. Empirically, we set C as 0.3 according to [263].

Texture masking refers to the ability of hiding or obscuring a superimposed stimulus with textures. [32, 263] show that the visual sensitivity to distortion is low in the texture-rich regions. The visual importance defined by texture masking is estimated as:

$$\mathcal{TM} = \max_{k=1,2,3,4} |\boldsymbol{x} * \boldsymbol{h}_k| \cdot (\boldsymbol{m}_{\boldsymbol{x}} * \boldsymbol{l}_g) \tag{4.3}$$

where $\boldsymbol{h}_k$ $(k = 1,2,3,4)$ are four directional high-pass filters for texture detection, $\boldsymbol{m}_{\boldsymbol{x}}$ denotes the edge map of image $\boldsymbol{x}$ given by the Canny edge detector [19], and $\boldsymbol{l}_g$ represents a Gaussian low-pass filter. The filter parameters are selected following work [263].

$$\boldsymbol{h}_1 = \frac{1}{16}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 3 & 8 & 3 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ -1 & -3 & -8 & -3 & -1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \boldsymbol{h}_2 = \frac{1}{16}\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 8 & 3 & 0 & 0 \\ 1 & 3 & 0 & -3 & -1 \\ 0 & 0 & -3 & -8 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{bmatrix}$$

$$\boldsymbol{h}_3 = \frac{1}{16}\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 3 & 8 & 0 \\ -1 & -3 & 0 & 3 & 1 \\ 0 & -8 & -3 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{bmatrix}, \boldsymbol{h}_4 = \frac{1}{16}\begin{bmatrix} 0 & 1 & 0 & -1 & 0 \\ 0 & 3 & 0 & -3 & 0 \\ 0 & 8 & 0 & -8 & 0 \\ 0 & 3 & 0 & -3 & 0 \\ 0 & 1 & 0 & -1 & 0 \end{bmatrix}$$

The parameters for the Gaussian low-pass filer $\boldsymbol{l_g}$ used in Eq.(4) are: 3×3 Gaussian kernel with mean as 0 and standard deviation as 0.5.

Compared with the absolute luminance of a single pixel, human perceptions are more sensitive to the relative luminance among its neighboring pixels. The luminance adaptation threshold is calculated based on Weber's law and deduced from psychological experiments under uniform background [174]. The luminance adaptation effect $\mathcal{LA}$ is modeled as,

$$\mathcal{LA}_{i,j} = \begin{cases} 17\times(1-\sqrt{\frac{\tilde{\boldsymbol{x}}_{i,j}}{127}}) & \text{if } \tilde{\boldsymbol{x}}_{i,j} \leq 127 \\ \frac{3\times(\tilde{\boldsymbol{x}}_{i,j}-127)}{128}+3 & \text{otherwise} \end{cases} \tag{4.4}$$

where (i,j) denotes pixel position of a grayscale image, $\mathcal{LA}_{i,j}$ denotes the $(i,j)-$th component of the luminance adaptation map, $\tilde{\boldsymbol{x}} = \boldsymbol{x} * \boldsymbol{l}$, and $\boldsymbol{l}$ is a low-pass filter.

4.3.2 Frequency JND model

In addition to spatial luminance adaptation and texture masking effects, the sensitivity of the human visual system is closely related to frequency sensitivity [101]. We adopt a frequency perceptual model proposed in [273]. To describe the frequency perceptual model in short, images are firstly decomposed into sub-band domains. Then, local contrast masking and spatial contrast sensitivity factors can be modeled based on frequency coefficients in each block. The final frequency JND is obtained as the multiplication of these two factors.

4.4 Method

4.4.1 Imperceptible spatial-domain attack

By departing from the identical ℓ_p bound for each pixel value, we consider the perceptual importance of pixels which directly depend on image local structures [69, 258]. Specifically, we allow larger perturbations to perceptually insignificant regions while smaller or no perturbations to perceptually significant regions. The perceptual importance is estimated from the neighborhood structures using the spatial JND model. To explicitly consider the imperceptibility property, we propose to incorporate and rectify existing adversarial example generation methods utilizing the spatial JND constraint.

In the spatial domain, the optimization function of our JND-constraint spatial adversarial attack model is formulated as,

$$\begin{aligned} & f_\theta(\boldsymbol{x}^*) \neq y \\ & \text{s.t.} \quad |\boldsymbol{x} - \boldsymbol{x}^*| \leq \mathcal{JND}_s \end{aligned} \tag{4.5}$$

where $|\cdot|$ is the absolute value operator, $\mathcal{JND}_s$ denotes the spatial importance matrix estimated from the JND model computed from $\boldsymbol{x}$. An intuitive explanation of our objective function is the following: We distinguish pixel-wise importance inherent in images extracted from local structures. Consequently the perturbation budgets vary from region to region. The gradient of the output with respect to the clean input $\boldsymbol{x}$ is a key value in the adversarial attack generation. In black-box attacks, as there is no internal knowledge on either the model architecture or the loss function, it is impossible to calculate gradient directly. However, different types of black-box attacks leverage different methods to estimate such gradient information. In the substitute model based attack which is our main focus, we can estimate the gradient with an substitute model [47, 69, 258], then generate adversarial examples regarding to the new constraint as in Eq.(4.5), and finally transfer examples to the black-box model. In this study, we denote the gradient estimation method as $\boldsymbol{g}^{est}(\boldsymbol{x}, y)$.

The perceptual-constraint model can be solved using the gradient-based method

iteratively,

$$\boldsymbol{x}_{t+1}^{*} = \boldsymbol{x}_{t}^{*} + \alpha \cdot \mathcal{JND}_{s} \odot \text{sign}(\boldsymbol{g}_{t}^{est}(\boldsymbol{x}_{t}^{*}, y)) \tag{4.6}$$

where $\odot$ denotes the elementwise product, $\boldsymbol{g}_{t}^{est}(\boldsymbol{x}_{t}^{*}, y)$ is the estimated gradient w.r.t. $\boldsymbol{x}_{t}$ at the $t-$th iteration. Informal studies show that exceeding JND thresholds occasionally does not yield severely degraded visibility. Therefore, we can exploit the additional tolerance by multiplying the importance map by a scalar factor $\alpha(\alpha \geq 1)$ in Eq.(4.6). Then, we can better balance the trade-off between the attack success rate and image quality.

Compared with the commonly used ϵ ball uniform bound, the proposed perturbation bound is image dependent and region dependent, which directly incorporates spatial perceptual models. In Eq.(4.6), our image-dependent and stepsize-variant expression is a more general solution. Moreover, our method reduces to existing methods (e.g.,[69]) when we choose a uniform perturbation bound ϵ as $\alpha \cdot \max\{\mathcal{JND}_{s}\}$, then we have the same constraint optimization problem as in Eq.(4.1) with an ℓ_{∞} norm.

The overall structure-aware adversarial spatial attack framework is illustrated in Algorithm 1. In this study, the JND threshold is calculated based on the grayscale version of a natural image. The final JND profile of a color image is formed by replicating the grayscale JND for each color channel. Although there exists color JND models, here we adopt a simple JND model in order to show the perceptual improvement by explicit utilization of structural information.

4.4.2 Imperceptible frequency-domain attack

Apart from the spatial domain perturbation, recently there were several pioneering works on perturbing images in the frequency domain. Tsuzuku el al. investigated the sensitivity of neural networks to certain Fourier basis functions based on the linearity hypothesis of neural networks [236]. The adversarial examples can be crafted by making queries to the target model to find suitable Fourier basis. Adversarial examples from single Fourier attack method display repeated patterns in the pixel domain. Guo et al. restricted the adversarial perturbation space to the low frequency domain, and proposed a query-efficient attack method [74]. Despite the effectiveness of low frequency perturbations, the visual quality of adversarial

Algorithm 2: The proposed spatial structure-aware (SSA) adversarial attack algorithm.

Data: A black-box model $f(\boldsymbol{x})$, clean image $\boldsymbol{x}$, correct label y, gradient estimation method $\boldsymbol{g}^{est}(\boldsymbol{x}, y)$, scalar factor α_0, and iteration number T.

Result: An adversarial example $\boldsymbol{x}^*$.

Calculate $\mathcal{JND}_s$ from the grayscale version of $\boldsymbol{x}$.
$\mathcal{JND} \leftarrow [\mathcal{JND}_s; \mathcal{JND}_s; \mathcal{JND}_s]$.
Initialize $\boldsymbol{x}_0^* \leftarrow \boldsymbol{x}$, $t \leftarrow 0$
while $t < T$ *and* $f(\boldsymbol{x}_t^*) == y$ **do**
    estimate the gradient $\boldsymbol{g}_t^{est}(\boldsymbol{x}_t^*, y)$.
    $\boldsymbol{x}_{t+1}^* \leftarrow \boldsymbol{x}_t^* + \frac{\alpha_0}{T} \cdot \mathcal{JND} \odot \text{sign}(\boldsymbol{g}_t^{est}(\boldsymbol{x}_t^*, y))$, $t \leftarrow t+1$
end
$\boldsymbol{x}^* \leftarrow \boldsymbol{x}_t^*$.

examples is significantly degraded [206].

In previous frequency attack methods, adversarial perturbations are added in the spatial domain with the uniform ℓ_p-norm bound with frequency correction. However, our proposed frequency domain attack is directly conducted in the frequency domain iteratively without any spatial domain constraint. Instead we explicitly consider the perceptual distortion bounds in the frequency domain. This makes the proposed perceptual-constraint frequency attack different from existing adversarial attack methods. We observe that the proposed perceptual frequency-constraint adversarial attack generally yields higher perceptual quality than the spatial domain attack.

In this study, we use the discrete cosine transform (DCT) domain for the frequency domain attack. For expression convenience, we consider the single channel case, since it is straightforward to extend operations to color images by performing transformations for each channel. Assume the clean image $\boldsymbol{x} \in \mathbb{R}^{N \times N}$, then we can obtain $\boldsymbol{X}$ by dividing the spatial image into square blocks of size $\mathcal{B} \times \mathcal{B}$. The DCT transform is conducted for each block $\boldsymbol{x}^b (b = 0, 1, \cdots, \lceil \frac{N}{\mathcal{B}} \rceil - 1)$ as,

$$\boldsymbol{X}^b = \boldsymbol{D}\boldsymbol{x}^b\boldsymbol{D}^T \tag{4.7}$$

where $\boldsymbol{D}$ is an orthogonal matrix, $\boldsymbol{D}\boldsymbol{D}^T = I_{\mathcal{B}\times\mathcal{B}}$, with entries $\boldsymbol{D}_{m,n} (m, n = 0, 1, \cdots, \mathcal{B} -$

1) as,

$$\boldsymbol{D}_{m,n} = \begin{cases} \sqrt{\frac{1}{\mathcal{B}}} & \text{if } m = 0 \\ \sqrt{\frac{2}{\mathcal{B}}} cos(\frac{(2m+1)n\pi}{2\mathcal{B}}) & \text{otherwise} \end{cases} \tag{4.8}$$

Similarly to the perceptual model-based spatial attack, we formulate the objective function for the frequency attack as,

$$\begin{aligned} & f_\theta(\boldsymbol{x}^*) \neq y \\ & \text{s.t.} \quad |\boldsymbol{X} - \boldsymbol{X}^*| \leq \mathcal{JND}_f \end{aligned} \tag{4.9}$$

where $\boldsymbol{X}, \boldsymbol{X}^*$ denote the clean and adversarial example in the frequency domain, respectively; $\mathcal{JND}_f$ refers to the JND matrix estimated by the frequency JND model.

To solve Eq.(4.9), we need to calculate the gradient of the loss function w.r.t. $\boldsymbol{X}$. At each block, the gradient w.r.t. each frequency coefficient $\boldsymbol{X}^b_{u,v}(u, v = 0, 1, \cdots, \mathcal{B}-1)$ can be calculated by propagating spatial gradient to the DCT domain,

$$\boldsymbol{G}^{est}(\boldsymbol{X}^b_{u,v}, y) = \sum_{i=1}^{\mathcal{B}} \sum_{j=1}^{\mathcal{B}} \boldsymbol{g}^{est}(\boldsymbol{x}^b_{i,j}, y) \cdot \frac{\partial \boldsymbol{x}^b_{i,j}}{\partial \boldsymbol{X}^b_{u,v}} \tag{4.10}$$

And we derive the gradient propagation in the matrix form,

$$\boldsymbol{G}^{est}(\text{Vec } \boldsymbol{X}^b) = \boldsymbol{g}^{est}(\text{Vec } \boldsymbol{x}^b)^T \cdot \left(\boldsymbol{D}^T \otimes \boldsymbol{D}^T \right) \tag{4.11}$$

where Vec denotes the matrix vectorization operation, $\cdot$ and $\otimes$ denote inner product and matrix Kronecker product, respectively. Finally, we obtain the frequency gradient estimation $\boldsymbol{G}^{est}$.

With the frequency coefficient gradient computed from Eq.(4.11) as $\boldsymbol{G}^{est}_t$ at the $t-$th iteration, we can readily perform frequency attack with frequency JND in the DCT domain,

$$\boldsymbol{X}^*_{t+1} = \boldsymbol{X}^*_t + \beta \cdot \mathcal{JND}_f \odot \boldsymbol{G}^{est}_t \tag{4.12}$$

where $\boldsymbol{X}^*_t$ denotes adversarial example in the DCT domain at iteration t ($t = 1, 2, \cdots, T$), $\beta = \beta_0 / T$, β_0 is a scalar factor of frequency JND to balance the compromise between perceptual quality and attack success rates.

Algorithm 3: The proposed frequency structure-aware (FSA) adversarial attack algorithm.

Data: A black-box model $f(\boldsymbol{x})$, clean image $\boldsymbol{x}$, correct label y, gradient estimation method $\boldsymbol{g}^{est}(\boldsymbol{x}, y)$, scalar factor β_0, and iteration number T.

Result: An adversarial example $\boldsymbol{x}^*$.

Calculate $\mathcal{JND}_f$ from DCT coefficients of grayscale version of $\boldsymbol{x}$.
$\mathcal{JND} \leftarrow [\mathcal{JND}_f; \mathcal{JND}_f; \mathcal{JND}_f]$.
Initialize: $\boldsymbol{x}_0^* \leftarrow \boldsymbol{x}$, $\boldsymbol{X}_0^* \leftarrow DCT(\boldsymbol{x})$, $t \leftarrow 0$.
while $t < T$ *and* $f(\boldsymbol{x}_t^*) == y$ **do**
    Estimate the spatial gradient as $\boldsymbol{g}_t^{est}(f_\theta, \boldsymbol{x}_t^*, y)$.
    Calculate gradient w.r.t. DCT coefficient using Eq.(4.11) as $\boldsymbol{G}_t^{est}$
    $\boldsymbol{X}_{t+1}^* \leftarrow \boldsymbol{X}_t^* + \frac{\beta_0}{T} \cdot \mathcal{JND} \odot \boldsymbol{G}_t^{est}$
    $\boldsymbol{x}_{t+1}^* \leftarrow iDCT(\boldsymbol{X}_{t+1}^*)$, $t \leftarrow t+1$
end
$\boldsymbol{x}^* \leftarrow \boldsymbol{x}_t^*$

Finally, the structure-aware frequency perturbation method is described in detail in Algorithm 2.

4.5 Experiments

In this section, we evaluate the proposed structure-aware algorithms on three baseline attacks: Fast Gradient Sign Method (FGSM) [69], Momentum Iterative FGSM (MIM) [47], and Diverse Inputs MIM (DIM) [258]. Firstly, we describe the experimental setup and introduce the quantitative visual metrics that we adopted in the comparison. We then experimentally demonstrate the superiority of the proposed methods over baselines on the perceptual quality and the attack success rate. The perturbation residues are illustrated to show the structure-aware property. Finally, we discuss the sensitivity of the parameters in the methods.

4.5.1 Experimental setup

For substitute model based attacks, the substitute model is a cleanly trained Inc-v3 model [225] provided by the PyTorch pretrained model zoo [184]. We evaluate the effectiveness of the adversarial examples on six models, three of which are cleanly

trained, i.e,. Inc-v4 [226], ResNet-101, ResNet-152 [78], and the rest three models are adversarially trained, i.e., Inc-v3$_{adv}$, Inc-v3$_{ens3}$, Inc-v3$_{ens4}$ [233]. These models are from the NeurIPS 2017 competition track on adversarial attacks [53].

For the dataset, we randomly choose 1000 images from the ImageNet validation dataset [198], which can be correctly classified by the six evaluation models in the substitute model attack setting. The uniform perturbation bound ϵ is set as 14 to have a good attack success rate. The maximum iteration number T is set as 10, which is a default parameter in existing studies [47, 258]. For DCT transformation, we set block size as 8×8, as commonly used in JPEG compression and video coding [273].

4.5.2 Evaluation metrics

To reliably evaluate the perceptual improvement of the proposed structure-aware attacks, we adopt four image quality assessment Image Quality Assessment (IQA) metrics: multiSim3 [246], Feature Similarity for color images (FSIMc) [270], Naturalness Image Quality Evaluator (NIQE) [165] and Mean Opinion Score (MOS) [216]. MultiSim3 and FSIMc are full-reference IQA metrics, with scores within [0,1] where a higher score indicates better visual quality. NIQE is a no-reference IQA metric to measure the naturalness of tested images. NIQE produces a non-negative value, where lower values suggest better naturalness. MOS is a popular human subjective test where we adopt the absolute category rating principle, with image quality score ranging from 1 to 5. The higher the MOS, the better images appears visually similar to clean images. The detailed setting of our MOS test is as follows.

To test the perceptual improvement of our proposed framework, we design a subjective test for perceptual image quality evaluation. We invite 10 volunteers to score the visual quality of the adversarial images. In each series of comparisons from Section 4.5.3, i.e. FGSM series, MIM series and DIM series, we randomly choose 50 adversarial examples from each adversarial attack method. For example, in the FGSM series, we randomly select 50 adversarial images generated by FGSM, 50 images generated by SSA-FGSM, and 50 images generated by FSA-FGSM. During the subjective test, we show a volunteer one pair of images and give her/him two seconds to review. The pair of images include an adversarial image and its corresponding

clean image as reference. Finally the volunteer rates the adversarial image with a score. We repeat this process until all selected images are reviewed by this volunteer. The order of images for different volunteers are different. In this experiment, we employ the commonly used absolute category rating principle [216], with image quality score ranging from 1 to 5. MOS is computed by averaging subjective scores from all volunteers for each adversarial attack method. Specifically, the scores indicate:

- score = 1: visually bad and very disturbing;
- score = 2: poor visual quality with disturbing visual artifacts;
- score = 3: fair visual quality with acceptable perceptual distortion;
- score = 4: good visual quality with slight perceptual distortion;
- score = 5: excellent visual quality with almost imperceptible distortion.

To evaluate the attack effectiveness, we employ the averaged attack success rates (ASR) on six victim models [53]. In the following sections, for simplicity, we term structure-aware approaches as SSA (Spatial-Structure-Aware) and FSA (Frequency-Structure-Aware). Since our proposed methods are independent of gradient estimation methods, we individually incorporate the structure-aware strategies to different gradient-based baseline attacks in the following sections.

4.5.3 Perception improvement assessment

In this section, we compare the perceptual quality between three baseline attacks [47, 69, 258] and our proposed ones, respectively. In each comparison, we firstly keep the average ASRs comparable between the baseline and the proposed ones, i.e., the proposed methods produce equal or slightly higher ASR than the baselines. Then, we provide both quantitative and qualitative comparison results on generated adversarial examples.

Comparison with FGSM Attack [69]: The Fast Sign Gradient Method (FGSM) is a one-step gradient-based attack method, which is a fundamental and widely adopted attack method. The perturbation is generated by maximizing the

loss (e.g. cross-entropy) function $J(f_\theta, \boldsymbol{x}, y)$ w.r.t. the input image. The FGSM method meets $\|\boldsymbol{x} - \boldsymbol{x}^*\| \leq \epsilon$, and it has an expression as,

$$\boldsymbol{x}^* = \boldsymbol{x} + \epsilon \cdot \text{sign}(\boldsymbol{g}) \tag{4.13}$$

where $\boldsymbol{g} = \nabla_{\boldsymbol{x}} J(f_\theta, \boldsymbol{x}, y)$ denotes the gradient of the loss function w.r.t the clean sample.

Table 4.1 shows the attack success rates of FGSM and our proposed variants, i.e., SSA-FGSM and FSA-FGSM. To have a comparable attack success rate, we choose $\alpha_0 = 2.2$, $\beta_0 = 50$. This table shows that SSA-FGSM has similar attack success rate with FGSM for both cleanly trained models and adversarially trained models, while FSA-FGSM approach gives superior attack success rate for adversarially trained models than cleanly trained ones. For a fair comparison, we keep their averaged ASR comparable as: 27.8% (FGSM), 27.8% (SSA-FGSM) and 29.8% (FSA-FGSM), respectively.

Table 4.1: Attack success rate comparisons between FGSM and the proposed SSA-FGSM and FSA-FGSM methods. The attack success rate is in percent (%).

Attack	ResNet-101	ResNet-152	Inc-v4	Inc-v3$_{adv}$	Inc-v3$_{ens3}$	Inc-v3$_{ens4}$	Avg ASR
FGSM	33.2	32.0	**35.3**	22.7	25.3	18.1	27.8
SSA-FGSM	**34.9**	**34.4**	34.2	21.8	25.4	16.1	27.8
FSA-FGSM	28.4	27.2	29.7	**25.5**	**31.7**	**36.4**	**29.8**

Then we quantitatively assess the *visual superiority* of the proposed methods in Table 4.2. For IQA metrics, i.e., multiSim3, NIQE, FSIMc and MOS, the proposed SSA-FGSM achieves improvement by 3.4%, 5.2%, 0.45 and 1.09; and the proposed FSA-FGSM improves four IQA metrics by: 7.5%, 15.3%, 1.44, and 2.0, respectively. The quantitative comparison results confirm the significant perceptual improvement of the proposed methods over the vanilla FGSM attack. It is worthy to note that such visual improvement is obtained for free since we directly incorporate our strategies into vanilla FGSM. More importantly, compared with vanilla FGSM, the proposed methods require no sacrifice of the attack performance (i.e., average attack success rates).

Comparison with MIM Attack [47]: To improve the adversarial transferabil-

Table 4.2: Visual quality comparisons between FGSM, SSA-FGSM and FSA-FGSM methods. The symbol "↑" ("↓") indicates that a higher (lower) value is better in perceptual quality.

Attack	multiSim3 (↑)	FSIMc (↑)	NIQE (↓)	MOS (↑)
FGSM	0.862	0.762	3.002	1.79
SSA-FGSM	0.896	0.814	2.664	2.88
FSA-FGSM	**0.937**	**0.915**	**1.560**	**3.79**

ity, momentum is introduced to obtain the iterative version of FGSM as Momentum Iterative Method (MIM):

$$\boldsymbol{x}_{t+1}^{*} = \boldsymbol{x}_{t}^{*} + \frac{\epsilon}{T} \cdot \text{sign}(\boldsymbol{g}_{t+1}) \tag{4.14}$$

where T denotes the number of iterations, and the accumulated gradient is updated as, $\boldsymbol{g}_{t+1} = \mu \cdot \boldsymbol{g}_{t} + \frac{\nabla_{\boldsymbol{x}} J(f_{\theta,\boldsymbol{x}_t^*,y})}{\left\|\nabla_{\boldsymbol{x}} J(f_{\theta,\boldsymbol{x}_t^*,y})\right\|_1}$. After getting the updated gradient, we incorporate structural-aware strategies into MIM, and obtain the proposed SSA-MIM and FSA-MIM methods. We use $\mu = 1.0$ as suggested in [47].

Table 4.3: Attack success rate comparisons between MIM and the proposed SSA-MIM and FSA-MIM methods. The attack success rate is in percent (%).

Attack	ResNet-101	ResNet-152	Inc-v4	Inc-v3$_{adv}$	Inc-v3$_{ens3}$	Inc-v3$_{ens4}$	Avg ASR
MIM	**46.8**	44.8	56.0	24.8	29.3	30.0	38.6
SSA-MIM	44.2	**46.0**	**56.1**	26.1	30.3	29.8	38.8
FSA-MIM	40.7	39.2	47.0	**34.3**	**34.5**	**41.2**	**39.5**

In Table 4.3, we compare the attack success rates of the MIM method and the proposed variants, e.g., SSA-MIM and FSA-MIM methods. The parameters for the two methods are $\alpha_0 = 2.3$, $\beta_0 = 6.0$ for a comparable attack success rate with respect to the vanilla MIM method, i.e., the averaged attack success rates are 38.6% for MIM, 38.8% for SSA-MIM, and 39.5% for FSA-MIM, respectively.

The quantitative IQA results are computed and reported in Table 4.4. Overall, SSA-MIM improves four metrics individually and FSA-MIM achieves even more improved perceptual qualities. Specifically, the quantitative IQA improvements are: 4.3% on multiSim3, 8.3% on FSIMc, 0.83 on NIQE and 1.68 on MOS, respectively.

Therefore, the perceptual qualities of vanilla MIM can be largely improved by the utilization of the proposed structural-aware approaches.

Table 4.4: Visual quality comparisons between MIM, SSA-MIM and FSA-MIM methods. The symbol "↑" ("↓") indicates that a higher (lower) value is better in perceptual quality.

Attack	multiSim3 (↑)	FSIMc (↑)	NIQE (↓)	MOS (↑)
MIM	0.905	0.815	2.398	2.12
SSA-MIM	0.927	0.855	2.058	3.17
FSA-MIM	**0.948**	**0.928**	**1.569**	**3.80**

Comparison with DIM Attack [258]: In the DIM method, the inputs to the model are stochastically transformed copies of the original image to increase the adversarial transferability. At each iteration, correspondingly the gradient is updated with the transformation with a probability p. In the experiments, we selected p as 0.7 which was reported to achieve the highest averaged attack success rates [258]. Based on the DIM method, we have derivations of our proposed structure-aware variants, i.e., SSA-DIM and FSA-DIM methods.

Table 4.5: Attack success rate comparisons between DIM and the proposed SSA-DIM and FSA-DIM methods. The attack success rate is in percent (%).

Attack	ResNet-101	ResNet-152	Inc-v4	Inc-v3$_{adv}$	Inc-v3$_{ens3}$	Inc-v3$_{ens4}$	Avg ASR
DIM	**64.2**	62.8	73.6	31.6	32.6	32.1	49.5
SSA-DIM	62.8	**63.1**	**73.7**	33.6	35.3	33.4	50.3
FSA-DIM	53.0	52.2	60.5	**45.3**	**43.0**	**49.2**	**50.5**

To make the proposed attacks comparable with DIM [258] in ASR, we adopt $\alpha_0 = 2.35$, $\beta_0 = 6.5$ and show the attack success rate comparison between DIM, SSA-DIM and FSA-DIM methods in Table 4.5. The averaged attack success rates are 49.5%, 50.3% and 50.5%, respectively.

Compared with vanilla MIM (Table 4.3), vanilla DIM improves the averaged ASR by about 10% (Table 4.5). Correspondingly, we observe that the proposed attacks (i.e. SSA-DIM and FSA-DIM) also improve their ASRs over MIM-based methods (i.e. SSA-MIM and FSA-MIM) by a similar margin. This observation confirms that our proposed structure-aware strategies are indeed independent of

Table 4.6: Visual quality comparisons between DIM, SSA-DIM and FSA-DIM methods. The symbol "↑" ("↓") indicates that a higher (lower) value is better in perceptual quality.

Attack	multiSim3 (↑)	FSIMc (↑)	NIQE (↓)	MOS (↑)
DIM	0.906	0.816	2.431	2.15
SSA-DIM	0.926	0.851	2.112	3.15
FSA-DIM	**0.941**	**0.921**	**1.684**	**3.58**

gradient-based methods, i.e., the incorporation of perceptual models into existing attacks can still maintain their attack ability.

Meanwhile, we notice that vanilla DIM also suffers from the visual quality problem as reported in Table 4.6, e.g., the FSIMc is only 0.816. By contrast, SSA-DIM improves the metric by 3.9% and FSA-DIM further boosts its FSIMc metric to be 0.921.

Finally, we show several typical adversarial examples for qualitative visual comparison in Fig. 4.2, Fig. 4.3 and Fig. 4.4 . Let us use Fig. 4.4 as a visual comparison example. The first row depicts the clean images, and the last three rows display adversarial examples generated from DIM, SSA-DIM and FSA-DIM, respectively. For DIM, we observe the perceptual degradation phenomenon in adversarial images, especially in the smooth regions. In detail, the texture-like distortions make adversarial examples visually unpleasant and easily to be spotted (please zoom in Fig. 4.4 for better comparison). Compared with DIM, SSA-DIM clearly improves the perceptual quality by re-allocating larger perturbation budgets to those visual insensitive regions based on spatial perceptual models. In the last row, we observe that the proposed FSA-DIM produces adversarial examples with almost imperceptible visual quality. Therefore, with the proposed structure-aware strategies (i.e., SSA and FSA), we can achieve comparable attack success rates yet with significantly higher visual quality over baseline methods, both quantitatively and qualitatively.

This section shows experimental results and compare the perceptual improvement of the proposed structure-aware attacks with baseline attacks respectively. Overall, both the proposed spatial perceptual and frequency perceptual approaches can clearly improve the visual quality of adversarial examples with comparable aver-

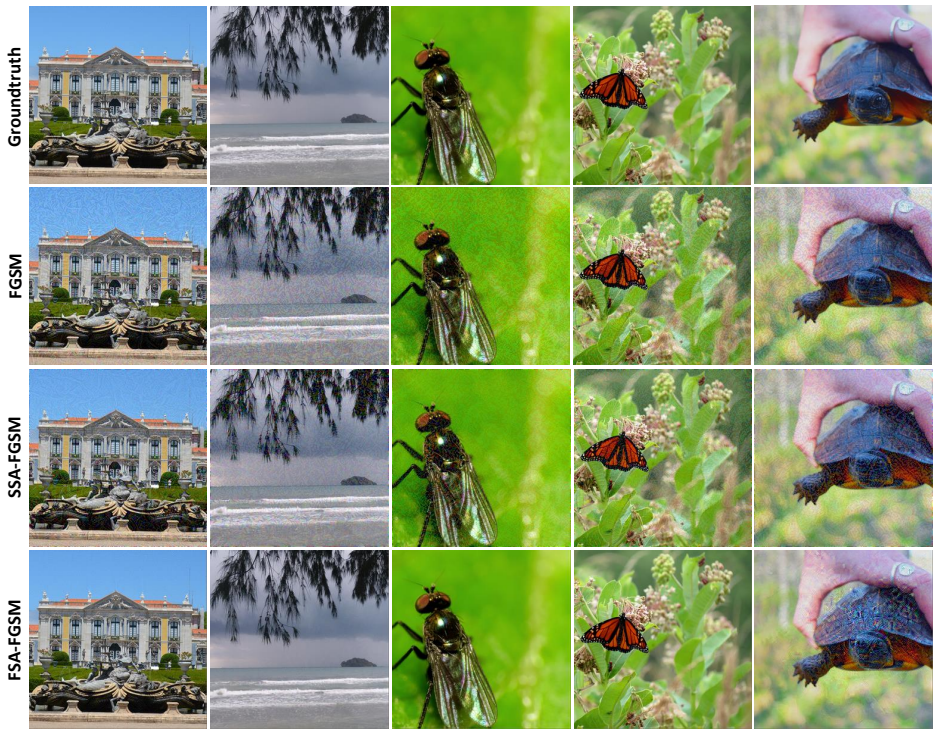


Figure 4.2: Examples of perceptual image quality comparison between FGSM, SSA-FGSM and FSA-FGSM methods. We recommend to zoom into the digital version for better visual comparison.

age ASRs. Particularly for the frequency perceptual attacks, our proposed methods can generate almost imperceptible adversarial examples for each compared baseline attack.

4.5.4 ASR improvement assessment

In general, for the same attack, we can always maintain better visual quality (with less adversarial perturbations) at the expense of lower attack success rates [47]. In this section, we compare ASRs of three baseline methods and our proposed methods. To be specific, we decrease the perturbation budget ϵ of each baseline attack to make their IQA metrics comparable with the proposed methods individually. The IQA values of SSA and FSA have been reported in Table 4.2 - Table 4.6 as the comparison reference.

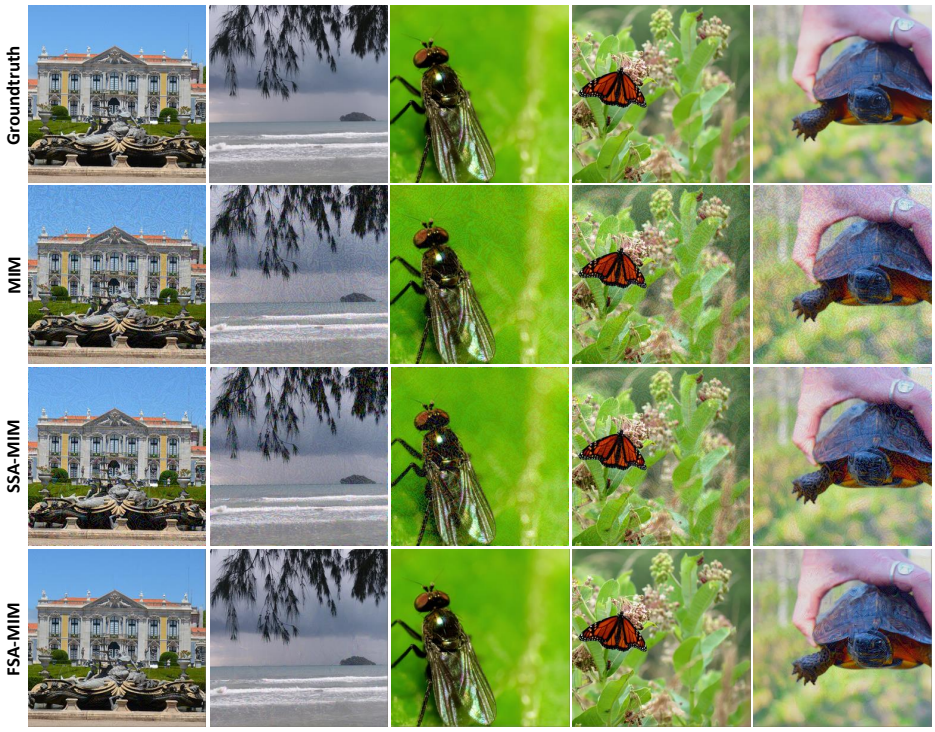


Figure 4.3: Examples of perceptual image quality comparison between MIM, SSA-MIM and FSA-MIM methods. We recommend to zoom into the digital version for better visual comparison.

Table 4.7: ASR improvement comparisons between the baseline attacks and their SSA/FSA versions, with the comparable visual quality.

Attacks	SSA equivalent		FSA equivalent	
	FSIMc	ΔASR (%)	FSIMc	ΔASR (%)
FGSM	0.801	1.9	0.913	10.1
MIM	0.851	3.0	0.924	11.5
DIM	0.851	4.5	0.923	13.1

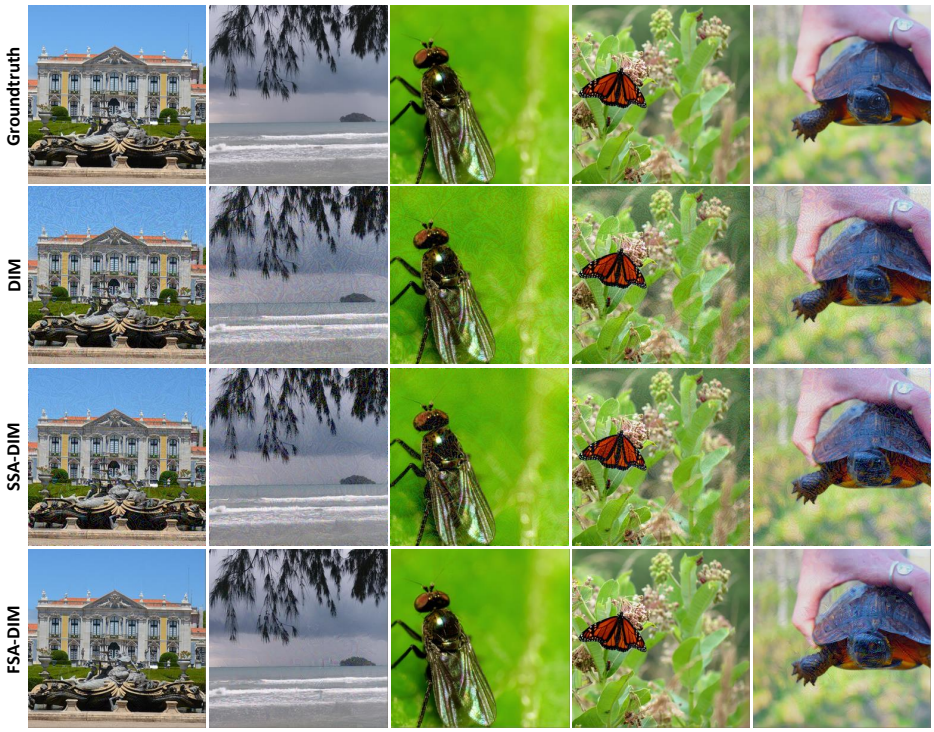


Figure 4.4: Examples of perceptual image quality comparison between DIM, SSA-DIM and FSA-DIM methods. We recommend to zoom into the digital version for better visual comparison.

With comparable visual quality (e.g. FSIMc), we show the ASR improvement ΔASR in Table 4.7. For the spatial perception-based methods, ASR improvement ranges from 1.9% to 4.5%. For the frequency perception-incorporated methods, ASR improves over baselines by 10.1% to 13.1%. This comparison result reveals another superiority of the proposed methods: by incorporating the proposed structure-aware strategies, we can achieve higher ASRs than baselines with comparably good visual quality. Therefore we can conclude that, compared with baseline attacks, the proposed methods manage to obtain a better trade-off between the attack success rates and perceptual quality.

4.5.5 Perturbation residues

To better understand the perceptual-based attacks, as an example we visualize perturbation residues of the DIM-based methods in Fig. 4.5. The parameters of attacks are the same as in Table 4.6. In general, we observe that DIM uniformly perturbs all pixels of the image which accounts for the visual degradation issue. By contrast, SSA-DIM mainly perturbs the visual insignificant regions which can be computed from spatial perceptual structures. Meanwhile, FSA-DIM approach perturbs the clean images with frequency insensitive adversarial perturbations in frequency perceptual bands, which generally appears invisible in the spatial domain.

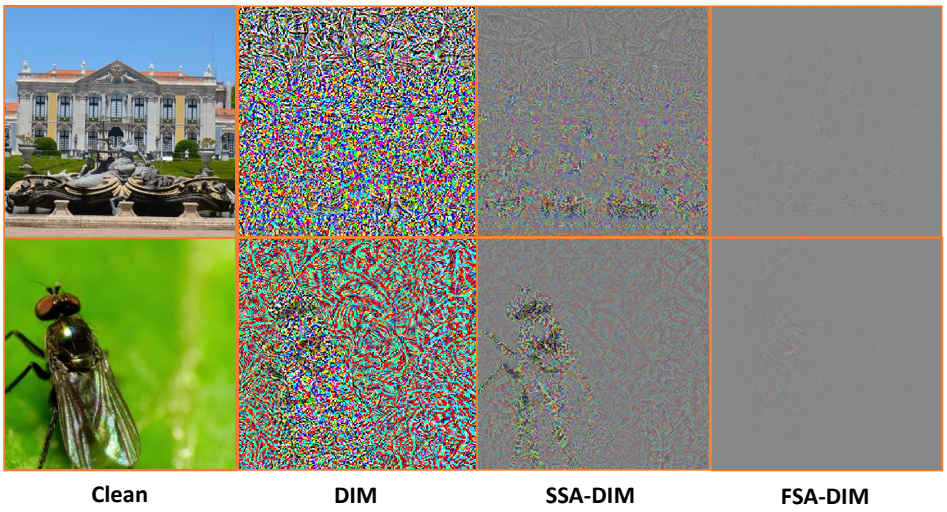


Figure 4.5: Comparison of perturbation residues between DIM ($\epsilon = 14$), SSA-DIM ($\alpha_0 = 2.35$) and FSA-DIM ($\beta_0 = 6.5$) attacks on example images.

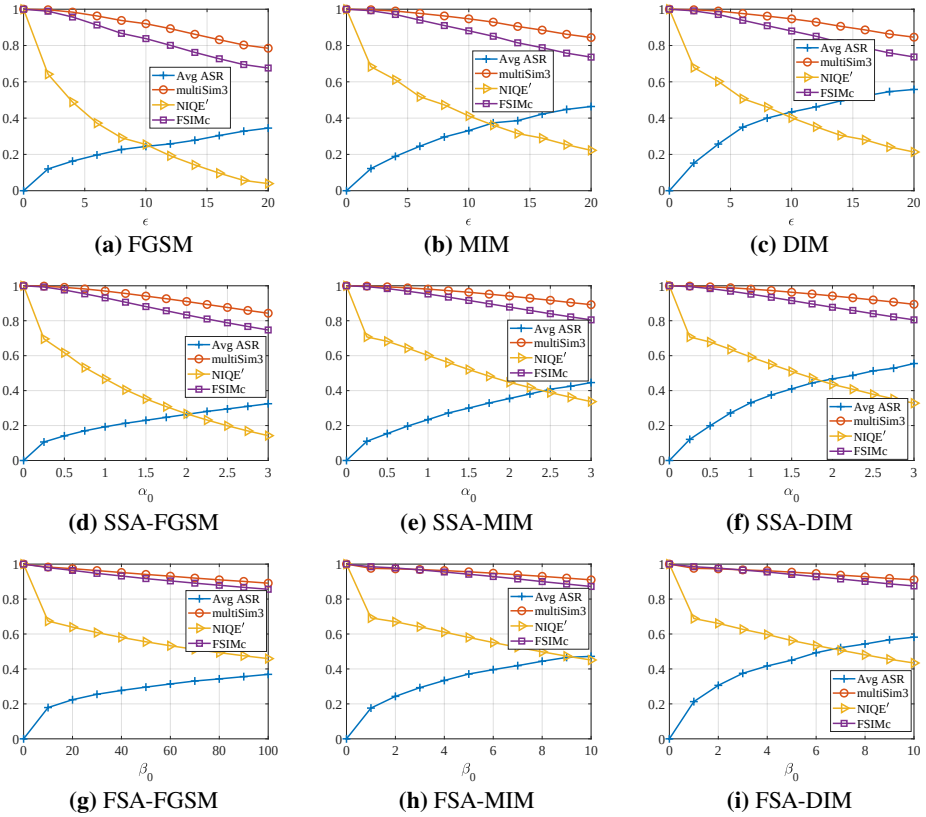
4.5.6 Parameter sensitivity

In this section, we study the effect of hyperparameters ϵ, α_0 and β_0 in the proposed attacks. To better illustrate the comparison trend of visual quality with respect to different hyperparameters, we normalize the NIQE values to be NIQE′: $NIQE' = 1 - NIQE/NIQE_{ub}$, where $NIQE_{ub}$ is an upper bound of $NIQE$ values for all experiments we conducted. A higher NIQE′ value indicates the better visual quality or vice versa.

In Fig. 4.6, we depict the parameter sensitivity curves for three baseline attacks (FGSM, MIM and DIM) and their perception-incorporated SSA/FSA based methods. The normalization factor $NIQE_{ub}$ equals 3.5 in the figures. In general, for each attack method, as hyperparameters (perturbation budgets) increase, the averaged ASRs increase at the expense of degraded visual quality (i.e. lower multiSim3, NIQE′ and FSIMc indices). We also observe that with comparable ASRs, the proposed methods consistently outperform their baselines. For instance, DIM achieves averaged ASR as 43.5% at $\epsilon = 10$ and its FSIMc equals 0.880. As a comparison, SSA-DIM produces averaged ASR to be 44.4% with FSIMc as 0.896 at $\alpha_0 = 1.75$. Meanwhile, FSA-DIM attains its ASR as **45.0**% with FSIMc equals **0.941** at $\beta_0 = 5.0$. The comparison results answer the question that *it is indeed possible to achieve a high ASR with improved visual quality*.

4.6 Conclusion

In this Chapter, we investigate evasion attacks on the multi-class classification task on natural images. This Chapter proposes two novel approaches to improve the perceptual quality of adversarial examples for deep networks in the transfer-based black-box setting. Since the existing uniform perturbation constraint does not align well with human visual systems, we explicitly consider the regional and structural information of images and incorporate the perceptual models into adversarial attacks. Specifically, we firstly introduce a spatial perceptual model and propose a structure-aware adversarial attack framework in the spatial domain. This framework is general and is compatible with all gradient-based attack methods. Further, we propose an adversarial attack framework by perturbing images in the frequency perceptual domain. Due to the structural constraints we explicitly consider, compared with baseline attacks, we demonstrate that adversarial examples produced by the proposed methods can generally have imperceptible or higher natural visual quality than the original attack methods with comparable attack success rates. Moreover, with comparable perceptual quality, the proposed methods produce higher attack success rates than baseline methods. In the future work, we plan to investigate and extend the proposed structure-aware frameworks to related tasks, e.g., imperceptible physical adversarial attacks. Through this work, we hope to raise the security



(a) FGSM **(b)** MIM **(c)** DIM

(d) SSA-FGSM **(e)** SSA-MIM **(f)** SSA-DIM

(g) FSA-FGSM **(h)** FSA-MIM **(i)** FSA-DIM

Figure 4.6: Parameter sensitivity comparisons between the baseline methods (i.e. subfigure (a)-(c)) and the proposed SSA (i.e. (d)-(f)) and FSA (i.e. (g)-(i)) based approaches.

awareness of researchers to develop more secure and more robust DL-based image classifiers.

Chapter 5

A Case Study of Composite Task: Towards Universal Physical Attacks on Single Object Tracking

5.1 Introduction

In previous Chapters, we explore attacks on matching and classification tasks alone. Also, the adversarial attacks take place in the digital image domain, e.g., by changing the pixel values in the image to fool image classifiers. In this Chapter, we consider the more challenging task: We study adversarial attacks on a combination of the three fundamental tasks: the matching, classification and regression tasks; Also, the attack takes place in the physical world, a more challenging scenario than digital attacks. As a representative case study, we investigate the single object tracking model. Specially, we investigate the feasibility of adversarial attacks on visual trackers in real-world scenes, e.g., by changing the physical part of a target object so that the target object cannot be correctly tracked by the tracker. In the following, we will introduce some background knowledge on the single object tracking model, discuss the challenges of this topic, and summarize our contributions in this Chapter.

Single object tracking has attracted increasing attention in security related applications, such as autonomous driving, intelligent surveillance and human-machine interaction [58, 159, 254]. The visual tracking task resorts to creating the dynamic correspondence (e.g., position) between a moving object in a given template frame and that in subsequent search frames without prior knowledge of object categories. Recently, there have been significant improvements in tracking performance with the adoption of Deep Neural Networks (DNNS). Providing a good tradeoff between real-time tracking and accuracy, the Siamese-based trackers, e.g., SiamRPN [130], SiamRPN++ [131], SiamMask [242], have become the mainstream approaches in visual tracking.

DNNS are shown vulnerable to adversarial perturbations, termed as adversarial attacks [223]. Such attacks exist for different vision tasks implemented with DNNS, e.g., image classification [124], object detection [257] and visual tracking [259]. Generally, adversarial attacks can be categorized into digital attacks and physical attacks, depending on which domain to inject the perturbations [92]. Specifically for single object tracking, recent studies primarily target at digital attacks [76, 259], leaving physical visual attacks rarely explored. Indeed, physical attacks are much more challenging than digital attacks due to practical constraints and feasibility.

In digital attacks, adversarial perturbations can be injected into any pixel of an image, and they can be different from image to image. In physical attacks, however, it requires the perturbation region to be small enough to be physically feasible, universal to diverse instances and robust to physical conditions (e.g., preprocessing, luminance factor). Also, physical attacks are more challenging to be detected and defended against, making them more threatening to trackers than digital attacks.

Despite certain pioneering explorations in physical attacks, existing works mainly focus on attacking image classifiers [3, 54] or object detectors [27, 92]. Probably against our intuition, though the task of visual tracking appears related with object detection (i.e., providing object bounding-box), their working mechanisms differ considerably. Object detection has one input and it estimates all locations of interested objects (instance-agnostic and category-dependent) while the single object tracking has two inputs and only localizes the user-specified target dynamically yet with no prior information of its category. In essence, visual tracking extracts and matches features between the template and search frames.

As illustrated in Fig. 5.1, it is challenging to attack the feature matching module. First, the dimensionality of feature maps are different between template and search frames. Therefore it is infeasible to employ the feature space adversarial loss proposed for classifiers [98]. Second, the coordinates of the tracking object are not spatially aligned in the feature space, which makes pixel-wise feature comparison proposed for physical attacks on object detectors [276] fail to generate effective perturbations in visual tracking. It is worth emphasizing that existing attack methods on classification and object detection cannot be applied on visual tracking. It necessitates a novel approach to *de-match* the Siamese features from the two branches, because of their differences in tasks, loss functions and architectures.

Meanwhile, it is desirable that the adversaries can control the shape of the target's bounding-box predictions, i.e., misleading bounding boxes to dilate or shrink promptly yet consistently over time. Further, physical attacks demand practical considerations, e.g., a patch small enough to be physically feasible, universally valid to different instances within the same category, and robust to physical conditions and tracker re-initialization. In this Chapter, we made the first attempt towards physically feasible universal attacks on SOTA Siamese-based visual trackers. The proposed method generates effective patches to significantly reduce the tracking performance of victim trackers in physically feasible scenarios.

In this Chapter, our major contributions are three-fold:

1. *We present the first physically feasible attack approach to evaluate the adversarial vulnerability of SOTA Siamese-based visual trackers. Our attack is universal to different instances from the same category and robust in physical conditions. The proposed framework can be a baseline to evaluate the robustness of Siamese-based visual trackers in the wild.*

2. *We propose the maximum textural discrepancy (MTD) loss function to misguide visual trackers by de-matching the template and search frames at hierarchical feature scales. Further, we consider the entire tracking pipeline, evaluating different shape attacks and optimization strategies to generate stronger and more controllable attacks.*

3. *Experimental results show that the proposed physically feasible attacks can*

efficiently fool SiamMask and SiamRPN++ both in standard visual tracking datasets and in physical conditions. (Digital scenes are to imitate universal physical attacks in the digital domain.)

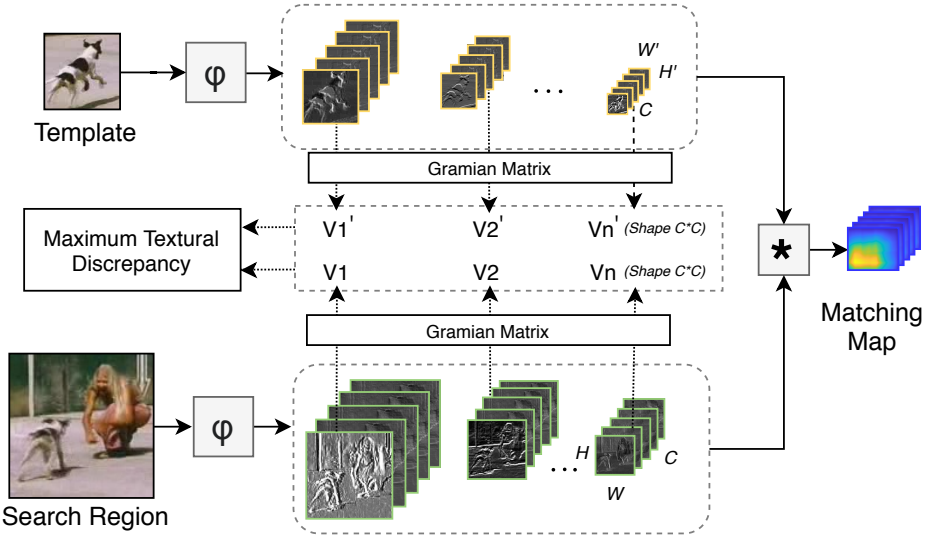


Figure 5.1: The mechanism of the Siamese-based matching network. Given a target template and the search region, their features are extracted by a Siamese network φ. Since the features could be in different shapes due to different image sizes of the template and search input, the features are matched using a cross-correlation layer to generate the matching map (here $*$ denotes the cross correlation).

5.2 Related work

5.2.1 Siamese-based visual tracking

Single Object Tracking Single Object Tracking (SOT) aims to track an arbitrary object in an online video stream without knowing the object category in advance. Different from object detection, SOT requires the tracker capable of tracking any object with a one-shot glance. Generally, SOT can be formulated as a *similarity learning* problem. Since the seminar work in [8] based on a fully-connected Siamese

architecture (SiamFC), there has been increasing interest in SOT by leveraging the fast running speed and expressiveness power of deep neural networks.

A Siamese network consists of two identical φ branches, which transform an exemplar image $\boldsymbol{z}$ and a candidate image $\boldsymbol{x}$ to the feature space prior to fusing them to return a score. The SiamFC tracker [8] first introduced a correlation layer which highly improved the tracking accuracy. SiamRPN [130] formulated SOT as a local one-shot detection task. Then it explored the region proposal sub-network Region Proposal Network (RPN) [192] to yield faster speed and competitive tracking performance. To address the translation invariance issue, SiamRPN++ [131] introduced a spatial-aware sampling strategy to significantly boost its performance gain by utilizing more sophisticated networks. SiamRPN++ also introduced the layerwise and depthwise aggregation module to further increase the tracker's performance. More recently, researchers studied the computational speed of visual trackers due to the pixel-level position estimate. SiamMask [242] alleviated this problem by formulating SOT as a multi-task learning problem. The SiamMask tracker involved training three tasks jointly, i.e., similarity matching module for dense response maps, RPN subnetwork for bounding box regression and binary segmentation for position refinement. SiamMask achieves the state-of-the-art performance on real-time visual tracking.

5.2.2 Digital attacks on visual trackers

It's demonstrated that DNNS are vulnerable to adversarial attacks on various computer vision tasks in the digital image domain, e.g., classification [70, 223, 257], object detection and segmentation [257], or some recent explorations on visual trackers [259, 261]. The work [259] employed the generative adversarial networks [67] with the proposed cooling-shrinking loss to generate imperceptible noise to attack the SiamRPN++ tracker. Then the perturbation was added to the template or search images on the network input (after pre-processing), which makes the attack unfeasible even in digital attacks. Guo et al. [76] proposed an online incremental attack. This attack exploits the spatial and temporal consistency in video frames so that the adversary fools object trackers with slight perturbations at each temporal frame. In work [261], the authors evaluated the vulnerability of Siamese-trackers

by hijacking the bounding box of the tracking object to be a different shape or to targeted position. Chen et al. [29] generated the perturbation on the template frame with dual attention. Wu et al. [253] proposed 3D adversarial examples for visual trackers. Unfortunately, these attacks only work in the digital domain since adversarial perturbations were specifically designed differently for exemplar and search frames. This makes it infeasible to extend to physically realizable attacks.

One more attack on visual trackers: In the interesting work [252], the authors designed an adversarial poster displayed on a big screen to fool the GOTURN tracker [81] when a person approaches the screen. However, there are major differences between this work and our proposed attack. First, [252] requires the poster to be big enough to fully cover tracking objects – the poster size is 2.6m×2m while halving the size would fail to attack. Essentially, this is an extension of digital attacks since the perturbations can lie in arbitrary regions in the background. Second, the work only largely perturbs search images without changing the template image. This is unrealistic in physical conditions and it cannot handle model re-initialization in trackers. Third, the victim GOTURN tracker is obsolete which works differently from SOTA trackers. In contrast, we examine the tracking pipeline and propose novel loss functions for adversarial attacks. The proposed method generates portable adversarial patches, small yet effective to attack the state-of-the-art trackers. Since our patches appear both in template and search images, they are capable of fooling trackers even with model re-initialization. We focus on more practical physical attacks on SOTA trackers. Such physically feasible attacks are more dangerous yet less explored. In Table 5.1, we compare the proposed methods with existing attacks from different aspects.

Table 5.1: Comparison of existing and the proposed adversarial attacks on visual trackers. "PF" denotes "Physically Feasible"; "SOTA" indicates the attacks are effective for SOTA visual trackers.

Attack	PF	SOTA	Universal	Re-initialization
[76]	×	✓	×	×
[259]	×	✓	×	×
[29]	×	✓	×	×
[261]	×	✓	×	×
[253]	×	✓	×	×
[252]	×	×	✓	×
Proposed Attacks	✓	✓	✓	✓

5.3 Physically feasible attacks

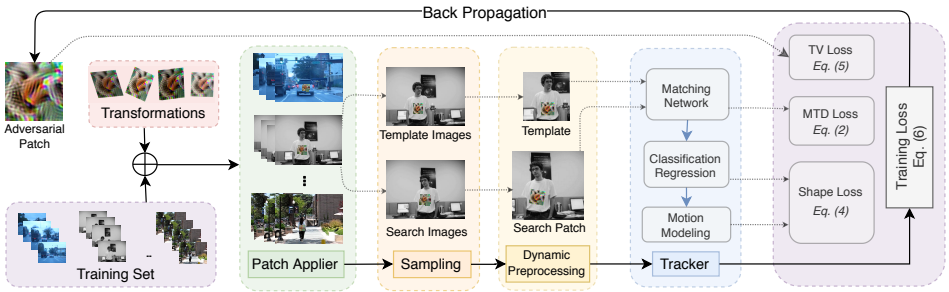


Figure 5.2: Overview of the proposed attack pipeline. Given a randomly-initialized patch and the training streams, firstly the patch is transformed randomly (e.g., random change in brightness, contrast, color, translations, rotation, sheering etc). Then the *patch applier* overlays the patch onto the target. At each iteration, we sample a random batch of frames, dynamically processing and passing them into the victim trackers. Finally the patch is updated by minimizing the proposed overall loss function.

In this section, we present the pipeline of the proposed method, as shown in Fig. 5.2. We first formulate the problem of physically feasible attacks on visual tracker, and we then elaborate our method in detail.

For a Siamese tracker with the matching network φ (see Fig. 5.1), we denote the template image as $\boldsymbol{z}^{(t)} \in \mathbb{R}^{w_z \times h_z \times c}$ at the t-th re-initialization, and the search image as $\boldsymbol{x}^{(t,s)} \in \mathbb{R}^{w_x \times h_x \times c}$ at the s-th frame corresponding to the t-th trial. The search image passes through a sub-window ω which involves cropping, padding and resizing operations before it is fed into the matching network. We denote the extracted features from the template and search images as $\varphi(\boldsymbol{z}^{(t)}) \in \mathbb{R}^{w'_z \times h'_z \times c'}$, $\varphi(\omega(\boldsymbol{x}^{(t,s)})) \in \mathbb{R}^{w'_x \times h'_x \times c'}$, respectively.

In physically feasible tracking attacks, adversaries attempt to find a universal patch $\boldsymbol{\delta}$ to significantly degrade the performance of the visual trackers *over time*. Suppose a target has been camouflaged by the adversarial patch, let us denote the exemplar image as $\boldsymbol{z}_{\delta}^{(t)}$ and the search image as $\boldsymbol{x}_{\delta}^{(t,s)}$, respectively. $\boldsymbol{z}_{\delta}^{(t)}$ and $\boldsymbol{x}_{\delta}^{(t,s)}$ can be expressed as,

$$\begin{aligned} \boldsymbol{z}_{\delta}^{(t)} &= \boldsymbol{z}^{(t)} \odot \boldsymbol{M}^{(t)} + (\boldsymbol{I}^{(t)} - \boldsymbol{M}^{(t)}) \odot \boldsymbol{\delta} \\ \boldsymbol{x}_{\delta}^{(t,s)} &= \boldsymbol{x}^{(t,s)} \odot \boldsymbol{M}^{(t,s)} + (\boldsymbol{I}^{(t,s)} - \boldsymbol{M}^{(t,s)}) \odot \boldsymbol{\delta} \end{aligned} \tag{5.1}$$

where $\odot$ represents the element-wise Hadmard product; $\boldsymbol{M}^{(t)}, \boldsymbol{M}^{(t,s)}$ denote binary masks for $\boldsymbol{z}_{\delta}^{(t)}$ and $\boldsymbol{x}_{\delta}^{(t,s)}$, respectively; $\boldsymbol{I}^{(t)}, \boldsymbol{I}^{(t,s)}$ represent all-one matrices with the same dimension as $\boldsymbol{M}^{(t)}, \boldsymbol{M}^{(t,s)}$, respectively.

Similar to existing digital attacks [76, 259], we will blind the Siamese-based visual trackers *over time*. Concretely, assume that a victim tracker is re-initialized with an exemplar image $\boldsymbol{z}_{\delta}^{(t)} \in \mathcal{Z}_a$ where an adversarial patch has been attached on the tracking object. Correspondingly, the search frames are $\boldsymbol{x}_{\delta}^{(t,s)} \in \boldsymbol{\mathcal{X}}_a^{(t)} \triangleq \{\boldsymbol{x}_{\delta}^{(t,1)}, \boldsymbol{x}_{\delta}^{(t,2)}, \cdots, \boldsymbol{x}_{\delta}^{(t,S)}\}$. Then with $\boldsymbol{\delta}$, the tracker will fail to correctly match $\boldsymbol{z}_{\delta}^{(t)}$ with $\boldsymbol{x}_{\delta}^{(t,s)}$, $s = 1, \cdots, S$.

5.3.1 Maximum textural discrepancy

As illustrated in Fig. 5.1, the essence of Siamese-trackers is to locate the target in search regions via a matched filter represented by φ. The activated matched features will then be delivered to downstream functional sub-networks. Therefore, a sufficient condition to blind a Siamese-tracker is to de-match its upstream representations.

The work [98] proposed a targeted-classification attack by minimizing the representation distance between the source and target images in the feature space. This attack was shown to achieve high transferability for classification tasks. However, unlike the targeted classification attack, there is no "target image" (i.e., an instance of a targeted class) in visual tracking. Also, the dimensionality difference of $\varphi(\boldsymbol{z}^{(t)})$ and $\varphi(\boldsymbol{x}^{(t,s)})$ hinders the direct calculation of the feature distance.

Recent studies reveal an intriguing phenomenon that neural networks are biased towards *textures* in image classification [62, 272]. Textures refer to certain spatially stationary statistics in natural images, which can be calculated from the Gramian matrix in the feature space [61, 105]. The textural feature is independent of feature dimensionality and it also explicitly exploits the vulnerability of neural networks.

Motivated by this, we propose the Maximum Textural Discrepancy (MTD) as

a novel loss function to fool the matched filter. Specifically, the objective is to maximize the discrepancy of textural representations between $\boldsymbol{z}_\delta^{(t)}$ and $\boldsymbol{x}_\delta^{(t,s)}$ so that the feature representations are de-matched in the upstream of visual trackers. The hierarchical MTD loss on D layers, $\mathcal{L}_{MTD}$, is defined as:

$$\mathcal{L}_{MTD}(\boldsymbol{z}_\delta^{(t)}, \boldsymbol{x}_\delta^{(t,s)}) = -\frac{1}{D} \sum_{d \in D} \left\| \boldsymbol{\mathcal{G}}\left(\varphi_d(\boldsymbol{z}_\delta^{(t)})\right) - \boldsymbol{\mathcal{G}}\left(\varphi_d(\omega(\boldsymbol{x}_\delta^{(t,s)}))\right) \right\|_{\mathcal{F}} \tag{5.2}$$

where $\mathcal{F}$ denotes the Frobenius norm; $\boldsymbol{\mathcal{G}}$ represents the Gramian matrix operator $\boldsymbol{\mathcal{G}} : \boldsymbol{f}^{(d)} \in \mathbb{R}^{w_d \times h_d \times c_d} \mapsto \mathbb{R}^{c_d \times c_d}$, where $\boldsymbol{f}^{(d)} \triangleq \{\boldsymbol{f}_{(1)}^{(d)}, \cdots, \boldsymbol{f}_{(c^d)}^{(d)}\}$ denotes the feature map profile composed of feature map $\boldsymbol{f}_{(i)}^{(d)} \in \mathbb{R}^{w_d \times h_d}$ of the i-th channel $(i = 1, \cdots, c_d)$, at the d-th layer $(d = 1, 2, \cdots, D)$. Concretely, given two feature maps $\boldsymbol{f}_{(i)}^{(d)}, \boldsymbol{f}_{(j)}^{(d)} \in \mathbb{R}^{w_d \times h_d}$ from the feature map profile $\boldsymbol{f}^{(d)}$, the Gramian output $\boldsymbol{\mathcal{G}}_{i,j}(\boldsymbol{f}^{(d)})$ at the i, j-th component $(i, j = 1, \cdots, c_d)$ can be computed:

$$\boldsymbol{\mathcal{G}}_{i,j}(\boldsymbol{f}^{(d)}) = \left\langle Vec(\boldsymbol{f}_{(i)}^{(d)}),\, Vec(\boldsymbol{f}_{(j)}^{(d)}) \right\rangle \tag{5.3}$$

where $<,>$ and $Vec(\cdot)$ denote the inner product and matrix vectorization operation, respectively.

Proposition 1. *The Gramian matrix in Eq. (5.3) turns out to be the correlation matrix of feature maps from different channels. By maximizing the textural discrepancy measured by the Gramian matrix, we can minimize the correlation between* $\boldsymbol{z}_\delta^{(t)}$ *and* $\boldsymbol{x}_\delta^{(t,s)}$ $(\forall t, s)$ *in the feature space*. Please find our proof in Section 5.5.

Remark. From the analysis above, we conclude that the MTD loss (in Eq. (5.2)) explicitly de-matches the feature representations produced from the matched filter φ.

5.3.2 Shape attacks

In visual attacks, it is desirable that attackers can misguide bounding-box predictions promptly and consistently *over time*. Here we consider shape attacks (i.e., shape dilation or shrinking) by fooling the downstream regression sub-network to make visual attacks in a controllable manner.

The SOTA Siamese trackers, e.g. SiamRPN++ and SiamMask, use RPN to locate the object's position, which consists of two branches: the regression network for proposal regression and the classification network for target or background prediction. The regression network predicts the shape of bounding boxes $\{(\widetilde{x}_i^{(s,t)}, \widetilde{y}_i^{(s,t)}, \widetilde{h}_i^{(s,t)}, \widetilde{w}_i^{(s,t)})\}_{i=1}^{N}$. The classification network discriminates the target from its background with the classification feature map and generates the similarity map. Further, motion modeling is adopted to re-rank the proposals' score $\{\widetilde{p}_i^{(s,t)}\}_{i=1}^{N}$. Finally, the bounding box with the highest score is selected as the target position.

In SiamMask and SiamRPN++, the motion model penalizes the position prediction and encourages the output to be spatially stable. Therefore it is challenging to interfere the final classification which could misguide trackers. As a result, alternatively we propose shape attacks by distracting the shape of the bounding boxes. In shape attacks, firstly we select a set of bounding boxes which provide top-K penalized scores: $\{\widetilde{p}_k^{(s,t)}\}_{k=1}^{K}$. Based on these penalized scores, we can explicitly consider the motion model in visual tracking. Concretely, the selected bounding boxes form a set $\Omega^{(s,t)} = \{(\widetilde{h}_1^{(s,t)}, \widetilde{w}_1^{(s,t)}), (\widetilde{h}_2^{(s,t)}, \widetilde{w}_2^{(s,t)}), \cdots, (\widetilde{h}_K^{(s,t)}, \widetilde{w}_K^{(s,t)})\}$. Denote the targeted bounding box shape as $(\overset{\vee}{h}, \overset{\vee}{w})$ and the regression margin as m_τ. The loss for regression shape attacks $\mathcal{L}_{Sha}$ can be written as:

$$\mathcal{L}_{Sha}(\boldsymbol{z}_\delta^{(t)}, \boldsymbol{x}_\delta^{(t,s)}) = \frac{1}{K}\sum_{k=1}^{K} max\left(\left|\widetilde{h}_k^{(s,t)} - \overset{\vee}{h}\right|_1 + \left|\widetilde{w}_k^{(s,t)} - \overset{\vee}{w}\right|_1, m_\tau\right) \tag{5.4}$$

In Eq.(5.4), with specified parameters ($\overset{\vee}{h}, \overset{\vee}{w}$ and m_τ), adversaries can control the desired shape of predicted bounding boxes after attack, i.e., shape dilation ($\overset{\vee}{h} = \overset{\vee}{w} = 1$) or shape shrinking ($\overset{\vee}{h} = \overset{\vee}{w} = -1$) attacks. It is worthy to mention that the proposed Shape loss is distinct from that in [259], because their loss necessitates a clean video as the input; however, we do not have such information in the physically feasible scenes. Moreover, we incorporate into our loss formulation the motion model in tracking attacks.

5.3.3 Universal physical attacks

Practical physical attacks present more challenges than digital attacks on the trackers. As elaborated below, we address three challenges: (1) physically realizable; (2) universal to diverse instances; and (3) robust to physical conditions and tracker re-initialization.

Physically feasible. The proposed patch-based loss functions (i.e. Eqs.(5.2) and (5.4)) can be directly applied to the physical conditions. Since natural images (or patches) generally look smooth, we consider the smoothness constraint to avoid sharp texture transitions and increase its "stealthiness". We use the total variation Total Variation (TV) [205] to penalize the smoothness term,

$$\mathcal{L}_{TV}(\boldsymbol{z}_{\delta}^{(t)}, \boldsymbol{x}_{\delta}^{(t,s)}) = \frac{1}{w_{\delta}\, h_{\delta}} \sum_{i=1}^{w_{\delta}} \sum_{j=1}^{h_{\delta}} \left\{ \left| \boldsymbol{\delta}_{i+1,j} - \boldsymbol{\delta}_{i,j} \right|^2 + \left| \boldsymbol{\delta}_{i,j+1} - \boldsymbol{\delta}_{i,j} \right|^2 \right\}^{1/2} \tag{5.5}$$

where w_{δ}, h_{δ} represent the width and height of the adversarial patch $\boldsymbol{\delta}$, respectively.

Universality. *Universality* could mean two aspects in physical attacks. First, the patch is effective for different instances within the same category (e.g. human, cars). Second, the patch remains adversarial for instances from different categories. Here we focus on the first case, and we leave the latter scenario as future work. Given the randomly sampled exemplar image $\boldsymbol{z}_{\delta}^{(t)} \in \mathcal{Z}_a$ and search frame $\boldsymbol{x}_{\delta}^{(t,s)} \in \boldsymbol{\mathcal{X}}_a^{(t)} \triangleq \{\boldsymbol{x}_{\delta}^{(t,1)}, \boldsymbol{x}_{\delta}^{(t,2)}, \cdots, \boldsymbol{x}_{\delta}^{(t,S)}\}$, the overall objective function $\mathcal{L}$ for universal physical attacks becomes,

$$\mathcal{L}(\boldsymbol{z}_{\delta}, \boldsymbol{x}_{\delta}) = \sum_{\boldsymbol{z}_{\delta}^{(t)} \in \mathcal{Z}_a} \sum_{\boldsymbol{x}_{\delta}^{(t,s)} \in \boldsymbol{\mathcal{X}}_a^{(t)}} \alpha \mathcal{L}_{MTD} + \beta \mathcal{L}_{Sha} + \gamma \mathcal{L}_{TV} \tag{5.6}$$

where α, β, γ denote the weights for loss functions $\mathcal{L}_{MTD}$, $\mathcal{L}_{Sha}$ and $\mathcal{L}_{TV}$, respectively.

Robustness. Robustness is important to ensure that the attacks work properly in the physical world, where the patch may suffer from different visual distortions when captured by a visual tracker (e.g. camera from a moving car). To mimic the real world conditions, we include diverse transformations and apply the expectation over transformation Expectation over Transformation (EOT) [3] on adversarial patches. Apart from some affine transforms (e.g. rotation, translation) in [3], we also

consider changes in perspectives, brightness, contrast and color jittering. Detailed setting can be found in Section 5.5. Denote the transformation as $\mathcal{T}$. Our robust adversarial patch on visual tracking $\boldsymbol{\delta}^{UPS}$ can be obtained by,

$$\boldsymbol{\delta}^{UPS} = \arg\min_{\boldsymbol{\delta}} \mathbb{E}_{\boldsymbol{\delta} \sim \mathcal{T}\boldsymbol{\delta}} \left[\mathcal{L}(\boldsymbol{z}_\delta, \boldsymbol{x}_\delta) \right] \tag{5.7}$$

where $\mathcal{L}(\boldsymbol{z}_\delta, \boldsymbol{x}_\delta)$ is given in Eq. (5.6).

The overall pipeline of the proposed attack, the universal physically feasible attack, is described in Algorithm 4.

Algorithm 4: The proposed algorithm of universal and physically feasible attacks on visual tracking.

Data: Training video streams set $\boldsymbol{X}$, number of template images T, number of search images S at one template, regression margin m_τ, loss weights α, β, γ.
Result: Optimized adversarial patch $\boldsymbol{\delta}^{UPS}$.

Initialize adversarial patch with Gaussian noise;
for $t = 0$ **to** $T - 1$ **do**
  Sample one template image $\boldsymbol{z}^{(t)}$ from $\boldsymbol{X}$;
  for $s = 0$ **to** $S - 1$ **do**
    Sample one search image $\boldsymbol{x}^{(t,s)}$ from $\boldsymbol{X}$ from nearby frames of $\boldsymbol{z}^{(t)}$;
    Sample a transform $\mathcal{T}$ on patch $\boldsymbol{\delta}$, warp transformed patch $\boldsymbol{\delta}$ to template $\boldsymbol{z}^{(t)}$;
    Sample a transform $\mathcal{T}$ on patch $\boldsymbol{\delta}$, warp transformed patch $\boldsymbol{\delta}$ to search image $\boldsymbol{x}^{(t,s)}$;
    Pre-process the image pair $\{\boldsymbol{z}^{(t)}, \boldsymbol{x}^{(t,s)}\}$ and input them to victim tracker;
    Compute MTD loss $\mathcal{L}_{MTD}$ from Eq.(5.2);
    Select bounding boxes set $\Omega^{(s,t)} = \{(\widetilde{h}_1^{(s,t)}, \widetilde{w}_1^{(s,t)}), (\widetilde{h}_2^{(s,t)}, \widetilde{w}_2^{(s,t)}), \cdots, (\widetilde{h}_K^{(s,t)}, \widetilde{w}_K^{(s,t)})\}$ based on top-K penalized scores;
    Compute the shape loss $\mathcal{L}_{Sha}$ from Eq.(5.4);
    Compute the total variation loss $\mathcal{L}_{TV}$ from Eq.(5.5);
    Compute the overall loss $\mathcal{L}(\boldsymbol{z}_\delta, \boldsymbol{x}_\delta)$ from Eq.(5.6);
    Optimize $\boldsymbol{\delta}$ using the Adam optimizer from Eq.(5.7);
  end
end

5.4 Experiments

In this section, we empirically evaluate the effectiveness of the proposed attacks on visual tracking both in digital and physically feasible scenes. The attacks in digital

scenes are to imitate the physically feasible attacks in the real world. Therefore we can quantitatively assess our attacks on the standard datasets and tune parameters more efficiently. The experiments were conducted on one NVIDIA RTX-2080 Ti GPU card using PyTorch [183].

5.4.1 Experimental setup

In all experiments, we keep the patch and object size ratio within 20% to be physically feasible. The victim models are SOTA Siamese-based trackers: SiamMask and SiamRPN++ [131, 242]. Adversarial patches were trained and tested on different instances and background to better evaluate their generalization and robustness. Please refer to Section 5.5 for detailed settings.

To quantitatively evaluate the attack performance, we employ three popular metrics in visual tracking: *success*, *precision* and *normalized precision* [56, 171]. The *success* is computed as the Intersection-over-Union (IOU) between the predicted bounding box and the groundtruth. The *precision* is measured by the distance between the tracking result and groundtruth bounding box in pixels. The *normalized precision* is computed with the Area Under Curve (AUC) between 0 and 0.5 [171].

5.4.2 Physically feasible attacks in digital scenes

For the physically feasible attacks in digital scenes, we experimented on three object categories: *person*, *car* and *cup* from the Large-scale Single Object Tracking (LASOT) dataset [56]. Each category consists of 20 videos, among which we randomly select one video for adversarial patch generation. We then attack the rest 19 videos within the same category by warping the patch on the target.

In Table 5.2, we report the performance drop on both trackers [242] where we consider the white-box attacks individually. As a comparison experiment, we also evaluate the influence of random patches (i.e. without training) with the same patch/object ratio. Interestingly, we observe that random patches can even boost the tracking performance. The reason might be that the random patch essentially provides more useful information for target localization. By contrast, there is a sharp performance decrease with adversarial patches in each category. We also quantitatively compare three metrics with different thresholds in Fig. 5.3

on “person”. Clearly, on SiamMask and SiamRPN++, adversaries can significantly reduce the tracking performance with our generated patches while the non-trained random patches improve the tracking performance. We have the same observations on “car” and “bottle” categories.

Table 5.2: Quantitative performance evaluation of the proposed attacks on SiamMask (#1) and SiamRPN++ (#2) with *person*, *car* and *bottle* categories. The table reports the percentage of performance drop in tracking with patches from: *Random, Dilation* and *Shrinking* attacks, respectively. “↓” denotes performance drop and larger values are preferred.

Category	Metric	Random (↓%)		Dilation Attack (↓%)		Shrinking Attack (↓%)	
		#1	#2	#1	#2	#1	#2
Person	Success	-24.7	8.4	42.5	37.0	38.8	65.1
	Precision	-37.9	-1.3	38.9	35.8	20.4	74.2
	Norm Precision	-24.4	9.0	36.0	24.3	17.8	76.9
Car	Success	-11.9	-2.7	46.9	57.5	32.2	44.5
	Precision	-11.1	-5.3	39.5	41.2	21.2	42.8
	Norm Precision	-12.6	-3.0	45.2	41.7	19.8	43.0
Bottle	Success	-9.9	-16.5	49.5	38.2	45.7	25.4
	Precision	-14.9	-30.3	69.5	67.9	18.5	20.5
	Norm Precision	-12.9	-22.9	44.1	27.6	5.6	34.0

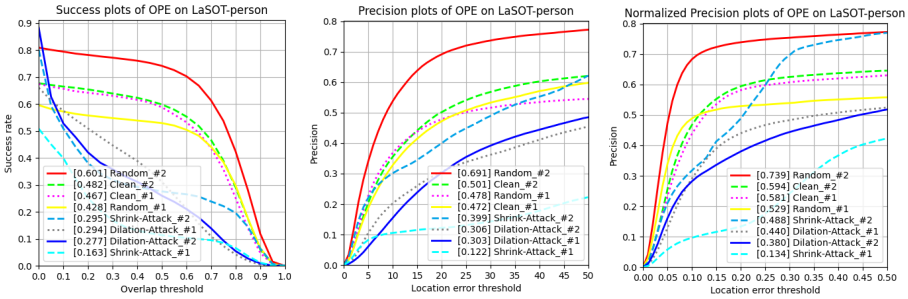


Figure 5.3: Quantitative comparison of three metrics on *person* with different thresholds.

In Fig.5.4, we show visual examples of the “dilation” and “shrinking” attacks on the “person” object on SiamRPN++. There are two observations: (1) the IoU (2nd row) of both attacks quickly drop to a low value with an adversarial patch; otherwise, the IoU keeps a high value without attacks. Correspondingly, SiamRPN++ produces dilated (1st row) or shrinked prediction boxes (3rd row). (2) The tracker keeps losing

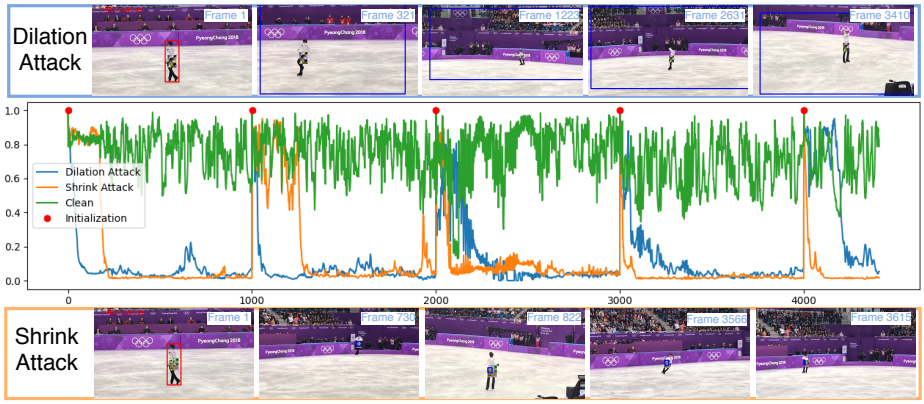


Figure 5.4: Illustration of the effectiveness of the generated patch. The 1st and 3rd rows show visual examples of the proposed dilation and shrinking attacks on "person". Bounding boxes in red depict the initialization positions while the blue ones display predicted positions after our attacks. The 2nd row shows the comparison in IoU prediction between clean and attacks over time. The red dot indicates model initialization at that time.

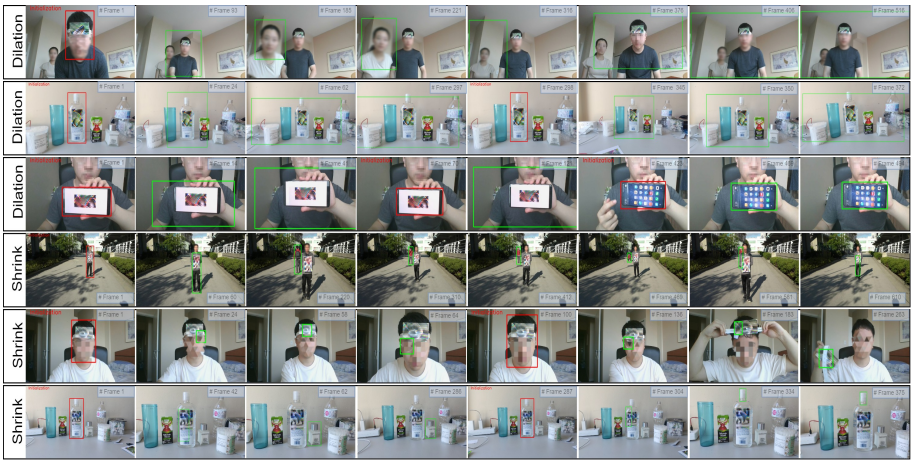


Figure 5.5: Example frames of tracking results of the proposed physically feasible attacks in real scenes.

the target even if we re-initialize it with a new template image (2^{nd} row). These observations further confirm the effectiveness of the proposed physically feasible attacks.

5.4.3 Physically feasible attacks in real scenes

After having verified our attacks in virtual scenes, we conduct experiments to demonstrate their efficacy in real world environments. In physical attacks, we mainly experiment on the "person" and "bottle" categories with diverse instances.

In Fig. 5.5, we show example frames of tracking results after physical attacks with "dilation" (rows 1 - 3) and "shrinking" attacks (rows 4 - 6), respectively. In "dilation" attacks, the predicted bounding box dilates to the full frame size which fails the victim tracker gradually yet promptly. Specifically, in the second row, the template is selected as the white bottle (i.e. Frame #1 in red as the target, textural region in the middle as our patch). In less than one second (*Frame #24*, real time=0.8 second), however, the tracker has been confused by erroneously tracking two other bottles as the target. The bounding box continues dilating until it "fills in" the whole image frame (*Frames #24 – 297*). At Frame #298, we re-initialize the tracker with the target object, however again the tracker has been easily fooled by our patch on it. In the third row, we display a small patch on a mobile phone screen (screen size 14.9 cm $\times$ 7.1 cm), the tracker quickly gets misguided even with model re-initialization (*Frames #70 – 121*). By contrast, when we remove the patch, the tracker can track well (*Frames #423 – 494*). Conversely, in "shrinking" attacks (rows 4 – 6), the predicted bounding box quickly shrinks to a small region and produces unstable predictions which eventually fails in tracking the target. For example, the tracker may confuse itself with objects near the target (*Frames #220 – 460* in row 4; *Frames #62 – 286* in row 6). The tracking predictions may also fall onto the patch and the predicted bounding boxes could be "threw away" intentionally (*Frames #136 – 263* in row 5).

We also quantitatively measure the performance of physical attacks. We manually annotate target objects on "bottle"(row 2) and "person" (row 5). To approximately measure the performance on clean objects (without patch), we manually annotate the patch region and replace patch values with uniform intensity as 127.

The performance drop of metrics are reported in Table 5.3.

Table 5.3: Quantitative performance evaluation in physical attacks. The symbol "↓" denotes performance drop and larger values indicate stronger attacks.

Category	Success (↓%)	Precision (↓%)	Norm Precision (↓%)
Person	65.6	36.0	54.1
Bottle	83.5	77.5	91.7

5.4.4 Ablation studies

We evaluate the influence of the MTD loss and patch size ratios. Ablation studies were conducted on the LASOT dataset.

MTD loss function. We compare the performance drop (i.e. w/ and w/o the MTD loss) on the "person" object on SiamMask in Table 5.4. Clearly, the MTD loss can boost the attack performance on three metrics. Similar observations for SiamRPN++ are shown in Section 5.5. This observation implies that MTD loss indeed enhances the attack ability.

Table 5.4: Ablation study of the MTD loss on SiamMask. "↓" denotes performance drop and larger values are preferred.

Metric	Dilation Attack (↓%)		Shrink Attack (↓%)	
	w/o MTD	w/ MTD	w/o MTD	w/ MTD
Success	28.5	**37.0**	53.7	**65.1**
Precision	26.5	**35.8**	55.5	**74.2**
Norm Precision	18.9	**24.3**	62.7	**76.9**

Patch size ratio. The patch size ratio is an important parameter in physically feasible attacks. Therefore, we evaluate the attack performance wrt different patch size ratios on SiamMask (#1) and SiamRPN++ (#2) in Fig. 5.6. In general, as the patch ratio increases from 15% to 35%, all three metrics decrease gradually, indicating stronger attack abilities. Therefore, the reported attack performances (with patch ratio as 20%) can be further improved if we utilize a larger patch size ratio in the experiments.

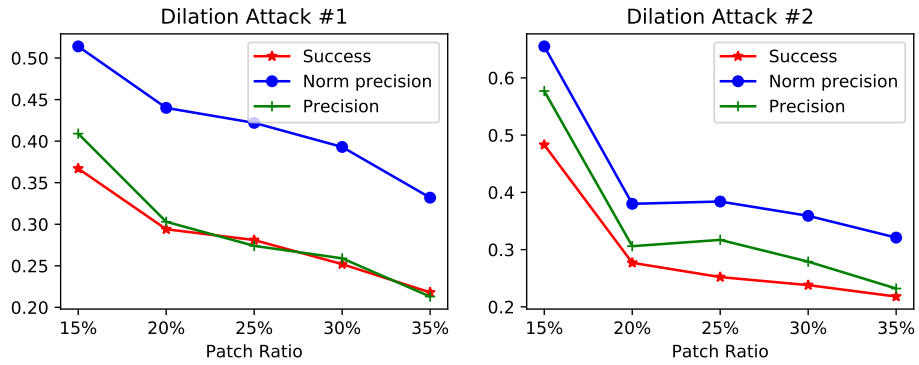


Figure 5.6: Attack performances as a function of the patch ratio.

5.5 Detailed setup and more results

Proof of proposition 1:

Proof. For the template frame, denote its feature map profile $\boldsymbol{f} \triangleq \{\boldsymbol{f}_{(1)}, \boldsymbol{f}_{(2)}, \cdots, \boldsymbol{f}_{(c)}\}$ where c represents the number of channels produced by a certain layer of interest. Assume the feature maps can be modeled by multivariate random variables $\boldsymbol{r} \triangleq \left(r_{(1)}, r_{(2)}, \cdots, r_{(c)}\right)^T$. Denote the observations of a random variable $r_{(k)}$ by $\boldsymbol{v}_{(k)} \in \mathbb{R}^{wh}$ $(k = 1, \cdots, c)$, where $\boldsymbol{v}_{(k)}$ is the vectorized representation of the k-th feature map $\boldsymbol{f}_{(k)}$. For the search frame, we follow the same assumption in the feature layer and denote the random variables by $\boldsymbol{r}' \triangleq \left(r'_{(1)}, r'_{(2)}, \cdots, r'_{(c)}\right)^T$. We also denote the observations of a random variable $r'_{(k)}$ by $\boldsymbol{v}'_{(k)} \in \mathbb{R}^{w'h'}$, where $\boldsymbol{v}'_{(k)}$ is the vectorized representation of the k-th feature map $\boldsymbol{f}'_{(k)}$ $(k = 1, \cdots, c)$.

The i, j-th component $(i, j = 1, \cdots, c)$ of the Gramian operator $\boldsymbol{\mathcal{G}}_{i,j}(\boldsymbol{v})$ and $\boldsymbol{\mathcal{G}}_{i,j}(\boldsymbol{v}')$ can be computed:

$$\boldsymbol{\mathcal{G}}(\boldsymbol{v})_{i,j} = \boldsymbol{v}_{(i)}^T \cdot \boldsymbol{v}_{(i)}$$
$$\boldsymbol{\mathcal{G}}(\boldsymbol{v}')_{i,j} = \boldsymbol{v}'^T_{(i)} \cdot \boldsymbol{v}'_{(i)}$$

The textural discrepancy term described by $\boldsymbol{\mathcal{G}}_{i,j}(\boldsymbol{v})$ and $\boldsymbol{\mathcal{G}}_{i,j}(\boldsymbol{v}')$ can be ex-

pressed:

$$
\begin{aligned}
\mathcal{L}_{td} &= \left\| \mathcal{G}(\boldsymbol{v}) - \mathcal{G}(\boldsymbol{v}) \right\|_{\mathcal{F}}^2 \\
&= \sum_i \sum_j \left(\mathcal{G}(\boldsymbol{v})_{i,j} - \mathcal{G}(\boldsymbol{v}')_{i,j} \right)^2 \\
&= \sum_i \sum_j \left[(\mathcal{G}(\boldsymbol{v})_{i,j})^2 + (\mathcal{G}(\boldsymbol{v})_{i,j})^2 \right] - 2 \sum_{i \neq j} \mathcal{G}(\boldsymbol{v})_{i,j} \mathcal{G}(\boldsymbol{v}')_{i,j} \\
&- 2 \sum_i \mathcal{G}(\boldsymbol{v})_{i,i} \mathcal{G}(\boldsymbol{v}')_{i,i} \\
&= \sum_i \sum_j \left[(\mathcal{G}(\boldsymbol{v})_{i,j})^2 + (\mathcal{G}(\boldsymbol{v}')_{i,j})^2 \right] - 2 \sum_{i \neq j} \mathcal{G}(\boldsymbol{v})_{i,j} \mathcal{G}(\boldsymbol{v}')_{i,j} \\
&- 2 \sum_i \left| \mathcal{G}(\boldsymbol{v})_{i,i} \mathcal{G}(\boldsymbol{v}')_{i,i} \right|
\end{aligned}
$$

Clearly, maximizing $\mathcal{L}_{td}$ involves minimizing the inner product of diagonal components from two Gramian operator outputs.

Let us zero-pad $\boldsymbol{v}_{(k)} \in \mathbb{R}^{wh}$ and it yields $\widetilde{\boldsymbol{v}}_{(k)} \in \mathbb{R}^{w'h'}$, where we have $\widetilde{\boldsymbol{v}}_{(i)}^T \widetilde{\boldsymbol{v}}_{(j)} = \boldsymbol{v}_{(i)}^T \boldsymbol{v}_{(j)}$. Then,

$$
\begin{aligned}
\sum_i \left| \mathcal{G}(\boldsymbol{v})_{i,i} \mathcal{G}(\boldsymbol{v}')_{i,i} \right| &= \sum_i \left| \widetilde{\boldsymbol{v}}_{(i)}^T \cdot \widetilde{\boldsymbol{v}}_{(i)} \cdot \boldsymbol{v}'^T_{(i)} \cdot \boldsymbol{v}'_{(i)} \right| \\
&= \sum_i ||\widetilde{\boldsymbol{v}}_{(i)}||^2 \cdot ||\boldsymbol{v}'_{(i)}||^2 \\
&\geq \sum_i ||\widetilde{\boldsymbol{v}}_{(i)}^T \cdot \widetilde{\boldsymbol{v}}'_{(i)}||^2
\end{aligned}
$$

Denote the cross-correlation between r.v. $r_{(i)}, r'_{(i)}$ as $\boldsymbol{K}_{r_{(i)}, r'_{(i)}}$, then we have:

$$
\lim_{w'h' \to \infty} \widetilde{\boldsymbol{v}}_{(i)}^T \cdot \widetilde{\boldsymbol{v}}'_{(i)} \;=\; (w'h')^2 \cdot \boldsymbol{K}_{r_{(i)}, r'_{(i)}}
$$

Therefore, given sufficient observations ($w'h' \to \infty$), we have:

$$\mathcal{L}_{td} \leq \sum_i \sum_j \left[(\boldsymbol{\mathcal{G}}(\boldsymbol{v})_{i,j})^2 + (\boldsymbol{\mathcal{G}}(\boldsymbol{v})_{i,j})^2 \right] - 2 \sum_{i \neq j} \boldsymbol{\mathcal{G}}(\boldsymbol{v})_{i,j} \boldsymbol{\mathcal{G}}(\boldsymbol{v}')_{i,j} - 2(w'h')^4 \cdot \sum_i |\boldsymbol{K}_{r_{(i)}, r'_{(i)}}|^2$$

Therefore, the textural discrepancy function $\mathcal{L}_{td}$ is a lower bound of the expression to the right represented by $|\boldsymbol{K}_{r_{(i)}, r'_{(i)}}|$. By maximizing $\mathcal{L}_{td}$, we can minimize the absolute value of the cross-correlation between feature representations of template and search frames modeled by multivariate r.v. $r_{(i)}$ and $r'_{(i)}$ for $i = 1, 2, \cdots, c$.

Parameters and setup:

We provide more details of the parameter setting in our experiments. We employ the Adam optimizer from the PyTorch platform with hyperparameters: exponential decays $\beta_1 = 0.9$, $\beta_2 = 0.999$, learning rate $lr = 10$ (for intensity between [0,255]), weight decay set as 0, the batchsize set as 20, and the training epochs as 300.

In the MTD loss in Eq.(5.2), we choose $D = 3$. Specifically, for SiamMask and SiamRPN++ trackers, which utilize the ResNet-50 as the backbone network, D layers are the last three residual blocks for multi-scale feature extraction.

In the Shape loss in Eq.(5.4), we set $K = 20$. More concretely, for the shrinking attack, we set $\overset{\vee}{h} = -1, \overset{\vee}{w} = -1, m_\tau = 0.7$; and for the dilation attack, we use $\overset{\vee}{h} = 1, \overset{\vee}{w} = 1, m_\tau = 0.7$.

In the overall loss in Eq.(5.6), the loss weights are set respectively as: $\alpha = 1000, \beta = 1, \gamma = 0.1$.

In the final loss in Eq.(5.7), the transforms that we employed and their parameters have been listed in Table 5.5.

To generate adversarial patches, firstly we randomly select one video from a category (e.g. person) as the training data and create the training pairs. The rest videos (with different instances/background) serve as the test set. Then we warp the trained patch on each frame of the test videos with the fixed patch size ratio to evaluate the attack performance.

Table 5.5: Patch transforms and parameters in the experiments.

Transforms	Parameters	Remark
brightness	0.1	brightness factor chosen uniformly from [0.9, 1.1]
contrast	0.1	contrast factor chosen uniformly from [0.9, 1.1]
hue	0.1	hue factor chosen uniformly from [-0.1, 0.1]
saturation	0.01	saturation factor chosen uniformly from [0.99, 1.01]
rotation	5	range of degrees chosen uniformly from [-5, 5]
translation	0.02	maximum absolute fraction (wrt image size) for translations
scaling	0.02	scale factor (wrt image size) chosen uniformly from [0.98, 1.02]
shearing	5	range of degrees chosen from [-5, 5]

Quantitative metrics comparison wrt different thresholds:

In Fig. 5.7, we visualize the quantitative comparison of three metrics (*success, precision, normalized precision*) on the *car* and *bottle* categories with respect to different thresholds. Consistent with the observations as reported in Section 5.4.2 on *person*, we observe that on the SiamMask and SiamRPN++ trackers, adversaries can also reduce the tracking performance considerably with our generated patches on *car* and *bottle* categories, while the non-trained random patches improve the tracking performance. These observations further confirm that our patches are indeed effective in misguiding the advanced visual trackers.

Ablation study on MTD for SiamRPN++:

We conduct and report in Table 5.6 the ablation study of the MTD loss on the SiamRPN++ tracker on the *person* category. For both of the dilation and shrinking attacks, we observe that MTD improves the attacking performance in three metrics. For instance, in the dilation attacks, the *success* metric improves by 11.3% by the incorporation of MTD; and this metric improves by 24.3% in the shrinking attack. Therefore, we can enhance the attack ability in the visual tracking attacks by additionally utilizing the MTD loss.

Ablation study on patch size ratio:

Supplementary to Fig. 5.6, the ablation study of the patch size ratio on "Shrinking" attacks is depicted in Fig. 5.8. Generally, we observe that the metrics decrease gradually as the patch size ratio increases from 15% to 35%. This observation

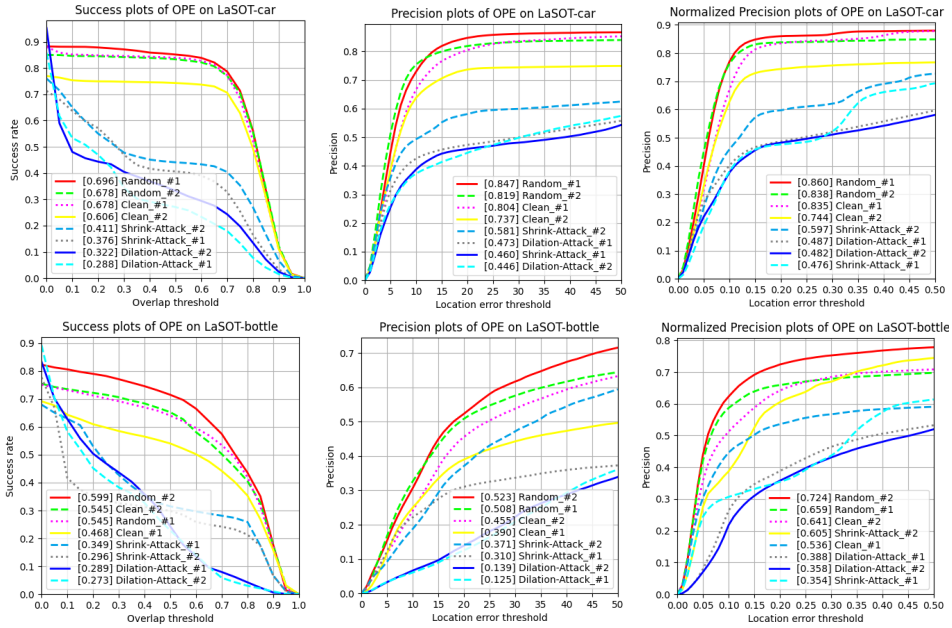


Figure 5.7: Quantitative comparison of three metrics on *car* and *bottle* categories with different thresholds.

Table 5.6: Ablation study on the MTD loss on SiamRPN++ on the "person" category. The symbol "↓" denotes performance drop and larger values indicate stronger attacks.

Metric	Dilation Attack (↓%)		Shrink Attack (↓%)	
	w/o MTD	w/ MTD	w/o MTD	w/ MTD
Success	31.2	**42.5**	14.5	**38.8**
Precision	24.7	**38.9**	14.0	**20.4**
Norm Precision	25.4	**36.0**	16.0	**17.8**

indicates stronger attack ability when we have a larger patch/object area ratio.

Visualization of patch examples:

In this section, we give several patch examples which were used in the experiments.

As examples for the physically feasible attacks in the digital scenes, we show in Fig. 5.9 of the adversarial patches that we used in Fig. 5.4. Specifically, Fig. 5.9a and Fig. 5.9b denote the adversarial patches on the dilation attack (1st row in Fig. 5.4)

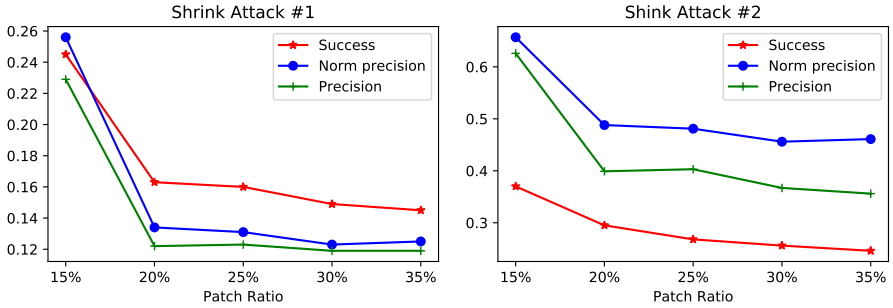
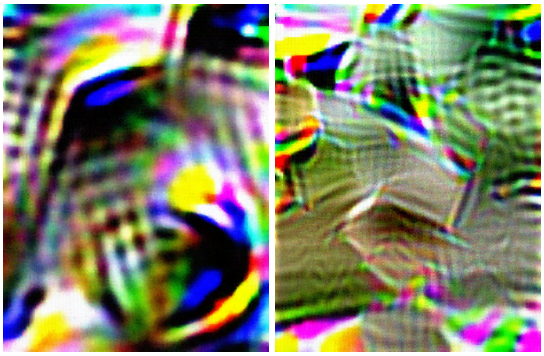


Figure 5.8: Attack performances as a function of the patch ratio on "shrinking" attacks.

and shrinking attack (2nd row in Fig. 5.4), respectively.



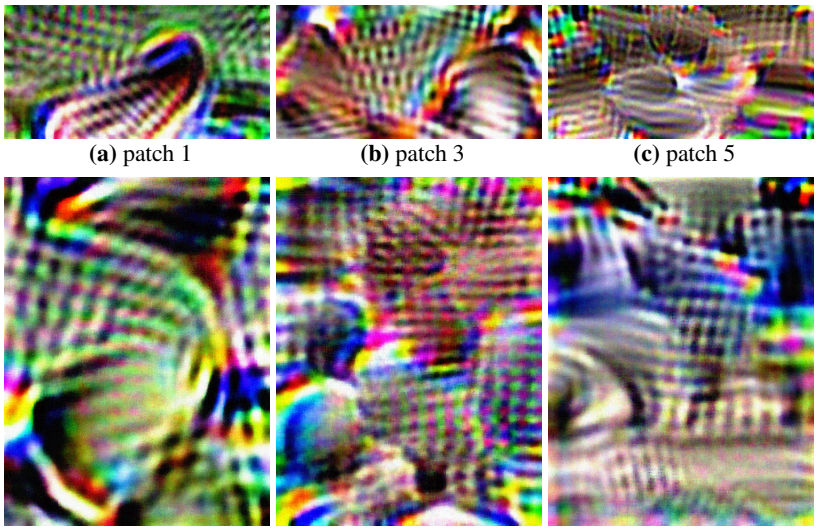
(a) Dilation patch **(b)** Shrink patch

Figure 5.9: Examples of adversarial patches in digital scenes.

In the real-world scene attacks, corresponding to Fig. 5.5, adversarial patches have been displayed in Fig. 5.10. To be more specific, Fig. 5.10a, Fig. 5.10b and Fig. 5.10c denote the patches from row 1, row 3 and row 5 in Fig. 5.5, respectively; and Fig. 5.10d, Fig. 5.10e and Fig. 5.10f represent the patches from row 2, row 4 and row 6 in Fig. 5.5, respectively.

5.6 Conclusion

This Chapter studies adversarial attacks on a combination of the three fundamental tasks: the matching, classification and regression tasks. As the first attempt,



(a) patch 1 **(b)** patch 3 **(c)** patch 5

(d) patch 2 **(e)** patch 4 **(f)** patch 6

Figure 5.10: Examples of adversarial patches in real-world scenes.

we study universal physically feasible attacks against single object tracking. To generate an effective patch, we propose the MTD loss to effectively de-match the template and search frames in hierarchical feature levels. We then propose two shape attacks to misguide visual trackers in a more controllable way. Finally we evaluate different optimization strategies to make the patch universal to different instances within a category and more robust to practical environments. Experimental results demonstrate that the proposed methods can significantly degrade advanced visual trackers' performances in the physically feasible attack setting. Our exploration on physically feasible attacks raises security concerns to be addressed by model designers in real-world visual tracking scenarios.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

Establishing secure, reliable and trustworthy human-aiding models for machine/deep learning tasks is an essential field of study for machine learning researchers and forensic analysts. Towards this goal, this thesis relies on the adversarial deep learning scheme, which involves a competing game, played by two-players: a specific model for a task and a virtual adversary. Under the paradigm of such a scheme, this thesis plays the role as an attacker. The thesis investigates the vulnerabilities of three dominant types of digital media analysis tasks: matching, classification and regression. As representative case study examples, this thesis selects four typical models in these tasks. The four models we have studied are: image hashing in image retrieval and authentication (as a representative matching task), GAN-generated fake face imagery forensic detection (as a representative binary classification task), deep learning-based image classification (as a representative multi-class classification task) and single object tracking (as a representative composite task which combines matching, classification and regression tasks). From the methodology perspective, we propose different approaches to expose potential threats that can negatively affect the models we are investigating. The effectiveness of our proposed methods has been verified with extensive experiments on different datasets. Major findings and contributions can be summarized as follows.

Firstly, we study the privacy issue of image hashing models by exploring

whether it is feasible to obtain images inverted from given image hashes and having good perceptual quality. We denote this problem as: the feasibility of perceptual image hashing inversion. We formulate the image de-hashing problem as a nonlinear mapping function approximation problem. To sufficiently approximate this image hashing reversion function, we leverage the functional expressiveness of deep neural networks, and we develop RevHashNet, the first image de-hashing network. To verify the effectiveness of RevHashNet, firstly we choose four typical image hashing algorithms as the models to be attacked, i.e. the victim models. We then perceptually revert image hashes on MNIST and MIX grayscale datasets, respectively. Experimental results demonstrate that, given real-valued image hashes (obtained from certain image hashing methods for image retrieval and authentication), the proposed RevHashNet can automatically reconstruct images that are perceptually similar to the original ones. Nevertheless, we observe that RevHashNet cannot reconstruct images with high visual quality in certain scenarios, e.g., image de-hashing when real-valued image hashes were quantized into shorter hash bits, or de-hashing large-resolution color images. To resolve the image de-hashing difficulty in such challenging scenarios, we then design PyLRR-Net which leverages deep residual learning to de-hash images in a progressive manner. At each image scale, we insert the proposed LRR block to refine the coarse image reconstruction. Our experimental results confirm the superiority of PyLRR-Net over RevHashNet in these more challenging image de-hashing scenarios. To be specific, PyLRR-Net produces much improved visual quality of de-hashed images than RevHashNet when quantized image hashes (i.e., hashes with shorter hash bits) was used. Moreover, PyLRR-Net can also successfully de-hash color images with large image resolution where RevHashNet will fail (e.g., de-hashing color images in ImageNet dataset). To summarize this study, the proposed RevHashNet and PyLRR-Net have demonstrated the feasibility of perceptual image hashing inversion. Through our exploration of image hashing reversion, we hope our work can raise the potential security and privacy issue for image hashing model developers.

Secondly, we explore adversarial attacks on fake face image detectors. Fake images that escape an adversarial attacked detector are usually degraded versions of original images. We analyze the visual degradation in such face images, and show how to design attacks that result in visually imperceptible adversarial images. Com-

pared with general adversarial attacks, the anti-forensic method for GAN-generated fake face imagery detection has its unique characteristics. In face images, facial and background regions are often smooth, even small perturbation could cause noticeable perceptual impairment which could be easily spotted through sanity checks. Therefore any attack that results in such a change in an image, renders existing adversarial attacks ineffective. In such circumstances, a smart adversary may turn to develop more imperceptible yet transferable adversarial attacks to be more practical. To this end, as a virtual adversary, firstly we analyze the perturbation residues from existing attacks; the perturbation analysis reveals the intuitive reason of how perceptual degradation results by applying existing attacks. By considering visual perception, we then propose a novel adversarial attack method, better suitable for image anti-forensics, in the transformed color domain. To demonstrate the effectiveness of the proposed method, we evaluate the perceptual quality improvement using three different visual metrics and one subjective study on StyleGAN, StyleGAN2 datasets, respectively. Our experiments reveal that, the proposed method significantly outperforms baseline attacks in terms of visual quality, at comparable attacks success rates. Besides, we observe that adversarial attack-based anti-forensics can also fool some non-deep learning based forensic detectors with a high attack success rate. Furthermore, we observe that attacking real face images are generally more difficult than attacking fake face images. Finally, this study also serves to raise the security awareness of fake face forensic model developers in the presence of imperceptible and transferable attacks.

Thirdly, we study how to fool image classifiers into making wrong decisions when the input is an image of good visual quality. We investigate the possibility to incorporate visual perceptual models into adversarial attacks on image classification models under the black-box setting. Specifically, to improve the perceptual image quality of black-box adversarial examples, we propose structure-aware adversarial attacks based on psychological perceptual models. That is, we allow higher perturbations to affect the perceptually insignificant image regions, while assigning lower or no perturbations on visually sensitive ones. In addition to spatial domain attacks, we also propose a novel structure-aware frequency adversarial attack method in the discrete cosine transform (DCT) domain. This framework is general and is compatible with all gradient-based attack methods. We confirm the effectiveness

of the two proposed methods via experiments on the ImageNet validation dataset. Experimental results show that, for comparable success rates, the images in the adversarial examples produced by our proposed methods can generally have imperceptible degradations or higher natural visual quality than images employed in the original attack methods. Moreover, with comparable image perceptual quality, the proposed methods produce higher attack success rates than baseline methods.

Finally, we study universal physically feasible attacks against single object tracking. An adversarial attack on an object being tracked involves changing the appearance of the tracked object so that it cannot be corrected tracked by the object tracker. This change can be done virtually (e.g. by changing the pixel values in the image) or physically (e.g. by changing the physical parts of the scene) . Existing attacks against visual tracking takes place in the digital domain, leaving it unexplored for visual tracking attacks in the physical world. More importantly, physical attacks pose more threats to real-world object tracking, and they are much more challenging than digital attacks. Therefore, we made the first step towards physically feasible adversarial attacks against visual tracking in real scenes. We blind the object tracker by pasting a printed universal patch somewhere on the object's surface. In our analysis, we realize that the essence of Siamese-based single object tracking lies in the feature matching between the object template and the searched frames. To generate a patch that can obstruct the object tracker, we propose the MTD loss to effectively de-match the template and search frames in hierarchical feature levels. We then propose two shape attacks to misguide visual trackers in a more controllable way. Finally we evaluate different optimization strategies to make the patch universal to different instances within a category and more robust to practical environments. To verify the effectiveness of the proposed attacks, we conducted experiments in both physically feasible digital scenes and physical scenes. Experimental results demonstrate that the proposed methods can significantly degrade advanced visual trackers' performances in the physically feasible attack setting. Our findings on physically feasible attacks raise the security concerns on adversarial vulnerabilities of the real-world visual tracking models.

To summarize, this thesis studies the vulnerabilities of four typical machine learning models, under the framework of adversarial deep learning. For each of these models, this thesis proposes virtual attack methods to reveal potential threat

vulnerability inherent in the model with reasonable threat modeling assumptions. By exposing the weakness of the examined models, we hope such studies could raise security concerns to be addressed by model designers when developing more secure, reliable and trustworthy machine learning models for digital media security and forensics.

6.2 Future work

In the future work, I plan to investigate three main directions which are of crucial importance yet less explored in digital media security: investigating perceptual image hashing in the presence of property attacks, developing more secure image hashing algorithms, and establishing secure defenses against adversarial attacks.

6.2.1 Property attacks on perceptual hashing

In this thesis, we have explored privacy vulnerabilities of image hashing algorithms by performing model inversion attacks. The vulnerability of two inherent properties, however, remains unexplored on perceptual image hashing — robustness to content-preserving manipulations and discrimination ability to malicious operations. As a virtual adversary, I plan to target these two properties, and propose $\mathscr{R}$-attack and $\mathscr{D}$-attack, respectively.

Given an image x_p and its content-preserving distorted version x'_p, a perceptual hashing algorithm $h_l, (l = 1, \cdots, L)$, then the robustness property states that h_l should produce two identical hashes with high probability,

$$Pr\Big(h_l(x_p) \approx h_l(x'_p)\Big) \geqslant 1 - \tau, \quad 0 \leqslant \tau < 1$$

In contrast, for an image x_p and a perceptually different image x_q, the discrimination property requires that h_l should yield two different hashes with high probability,

$$Pr\Big(h_l(x_p) \neq h_l(x_q)\Big) \geqslant 1 - \delta, \quad 0 \leqslant \delta < 1$$

The robustness and the discrimination properties have been depicted in Fig. 6.1. Here we describe our attack scheme which could be potentially exploited by adversaries in applications such as near duplicate detection [170, 175], and image

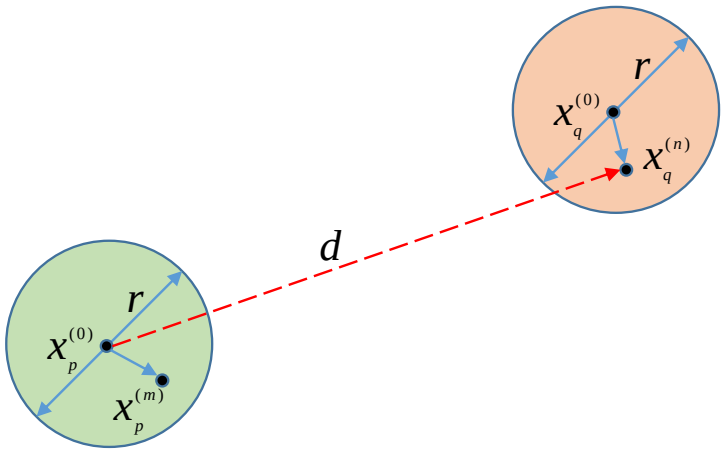


Figure 6.1: Illustration of the robustness and discrimination properties in perceptual image hashing. A perceptual image hashing function is supposed to produce similar hashes for an image within the r-robustness radius wrt the authentic image, and give very different hashes for an image distinctly different from the authentic image ($d \gg r$).

forgery detection [51, 154]. Given an authentic image $x_p^{(0)}$, a manipulated image $\forall x_p^{(m)} \in U(x_p^{(0)}, r)$, the objective of the $\mathscr{R}$-attack is to find a perturbation Δ,

$$\begin{aligned} \text{maximize} \quad & \sum_{l=1}^{L} \mathrm{D}_h\Big(h_l(x_p^{(0)}), h_l(x_p^{(m)} + \Delta)\Big) \\ \text{s.t.,} \quad & x_p^{(m)} + \Delta \in U(x_p^{(0)}, r) \end{aligned}$$

where $\mathrm{D}_h(\cdot)$ denotes some distance metrics in image hashing. Solving the optimization problem can be challenging in several aspects. Firstly, it may be infeasible to directly use the gradient-based perturbation attack method since some conventional hashing functions were not formulated as a gradient-based optimization problem. Secondly, for binary discrete hashing, the step function is non-smooth which may make it intractable to adopt back propagation for optimization. Thirdly, the objective function necessitates a perceptual metric to guarantee that the perturbed image $x_p^{(m)}$ is still within the perceptual vicinity of the authentic image $x_p^{(0)}$.

We plan to explore several strategies to resolve the challenges above. For the non-DL based hashing methods, we can establish surrogate DL-based models

which make them easier to optimize. To deal with the step function in binary codes, we can smooth such functions with the $tanh$ function to back propagate gradients [141]. Moreover, we can incorporate some perceptual constraints utilized in this thesis to tackle the third challenge.

Given the authentic image $x_p^{(0)}$, a perceptually different image $\forall x_q^{(n)} \in U(x_q^{(0)}, r)$, similarly, we can design the objective of the $\mathscr{D}$-attack,

$$\begin{aligned} \text{minimize} \sum_{l=1}^{L} \mathrm{D}_h\Big(h_l(x_p^{(0)}), h_l(x_q^{(n)} + \Delta)\Big) \\ \text{s.t.,} \quad x_q^{(n)} + \Delta \in U(x_q^{(0)}, r) \end{aligned}$$

In the future, we will conduct experiments to verify the feasibility of the proposed property attacks on perceptual image hashing algorithms. Also, we could extend such concepts to related security-critical areas such as property attacks on perceptual video hashing [113].

6.2.2 Establishing more secure image hashing

As a virtual adversary, our ultimate goal is to establish secure, reliable and trustworthy machine learning algorithms on digital media security & forensics. Therefore, after having explored worst-case vulnerabilities of image hashing, we plan to amend the exposed weakness and develop more secure hashing approaches.

Generally, there are two types of attacks against image hashing: privacy attacks (as explored in Chapter 2) and property attacks in Section 6.2.1. In privacy attacks, the success of adversaries mainly relies on the injection mapping between an image and a hash representation, which permits RevHashNet/PyLRR-Net to approximate the inverse mapping to reconstruct images from image hashes. To resolve the reversion issue, we propose Group-hashing with the randomization mechanism. Departing from the image-hash scheme, we propose to utilize multiple images to generate a hash, i.e., turning one-to-one mapping to many-to-one mapping. Specifically, firstly, we need to define several content-preserving operations to create a group of images from the authentic image. We then randomly select K images from the image pool, and generate image hashes individually. Finally, an image hash can be created from multiple hashes (e.g., simple average). Since the

hash was embedded from a group of images, it becomes more challenging to revert the image from the hash. Moreover, incorporating randomness into the group selection further increases the inversion difficulty since adversaries do not know which images/operations have been utilized to produce the image hash.

To counter property attacks, we propose to replace the vanilla hashing objective with the robust optimization-based version [156]. Given a pair of images $x_p,\ x_q \in \mathbb{R}^{M\times N}$, perturbation budget ϵ, let us define $y = 1$ if $x_p,\ x_q$ are perceptually similar; and $y = 0$ otherwise. The objective function of the proposed robust image hashing can be expressed as,

$$\min_{\theta} \max_{||\Delta||_p \leqslant \epsilon} \mathbb{E}_{(x_p, x_q, y)\sim\mathcal{D}} \sum_{l=1}^{L} y \cdot \mathrm{D}_h\Big(h_l^{\theta}(x_p), h_l^{\theta}(x_q+\Delta)\Big) + (y-1)\cdot \mathrm{D}_h\Big(h_l^{\theta}(x_p), h_l^{\theta}(x_q+\Delta)\Big)$$

where h_l^{θ} denotes the output of a hashing network parameterized by θ.

We will perform experiments to verify the security of the proposed image hashing methods, and check whether the proposed secure hashing could achieve comparable performance with vanilla hashing methods in terms of image authentication or retrieval purposes.

6.2.3 Establishing defenses on adversarial attacks

Adversarial attacks pose severe threats to deep learning models, e.g., image classification [46, 69], forensic detection [5], object detection [27, 257] and visual tracking [259, 261]. Therefore, it is crucial to establish defenses to counter such attacks.

Recall that in Section 1.1.5, depending on when attacks take place, the threat modeling can be generally categorized into two stages: attacks during the training phase and the inference phase, respectively. Correspondingly, defending approaches can be broadly divided into two categories based on the defense timing: training-based defenses and inference-based defenses, respectively. Training-based defenses include typical methods e.g., adversarial re-training [69], model distillation [182], provable defenses [33], robust architecture search [75] , and adopting robust training losses [179]. Amongst methods in this category, only adversarial re-training and

provable defenses are empirically shown to be robust to secondary adaptive attacks [2]. Nevertheless, both methods generally require intensive computational cost during model training, which renders them difficult to scale to large networks and image datasets. By contrast, inference-based defenses resort to directly protect deployed (trained) models from potential adversaries [43, 73, 150, 162, 163, 212]. In work [43], the authors propose to stochastically prune a subset of model activations to improve model robustness. This work achieves certain amount of robustness at the cost of loss of accuracy. Works [150, 163] created DNN-based detectors to distinguish adversarial examples with benign images. In works [162, 212], the authors propose to detect adversarial examples and map them back to the legitimate manifold using an DNN-based autoencoder. Unfortunately, these defending methods are broken by secondary attacks [2, 22].

We propose a two-stage secure defense scheme. Since adversarial perturbations can be considered the noise injection to clean images, therefore, firstly we will employ image denoising methods (e.g., median filtering, BM3D [35]) to remove certain amount of noise. In general, denoising operations may slightly influence model's predictions. In the second stage, we will further purify the denoised image by reconstructing it based on a private large-scale clean image dataset. To be specific, firstly we can divide the image into patches. We will then retrieve similar patches from the hash pool built upon the private dataset. Finally, the retrieved patches will be stitched together to form a denoised copy of the input image. Importantly, image hashes should be generated in a secure way (e.g., with a key for each image patch). By doing so, we should be able to defend secondary adaptive attacks in the white-box setting even though model parameters and our retrieval/stitching scheme can be transparent to adversaries. Indeed, the security of the proposed method directly relies on the key-protected secure hashing/stitching and the utilization of a private dataset, which can explicitly break the forward pass which is a requirement in the adaptive attack (a SOTA white-box attack for defenses) proposed in [2]. By contrast, the defense [73] merely adopts a image quilting method for adversarial perturbation purification, which was shown easily bypassed by [2].

We will conduct experiments to numerically verify the feasibility of the proposed secure defending approach over strong white-box attacks.

Bibliography

[1] Y. Adi, C. Baum, M. Cisse, B. Pinkas, and J. Keshet. Turning your weakness into a strength: Watermarking deep neural networks by backdooring. In *27th {USENIX} Security Symposium ({USENIX} Security 18)*, pages 1615–1631, 2018. → page 23

[2] A. Athalye, N. Carlini, and D. Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *arXiv preprint arXiv:1802.00420*, 2018. → page 148

[3] A. Athalye, L. Engstrom, A. Ilyas, and K. Kwok. Synthesizing robust adversarial examples. In *International conference on machine learning*, pages 284–293, 2018. → pages 117, 126

[4] S. Baluja. Hiding images in plain sight: Deep steganography. *Advances in Neural Information Processing Systems*, 30:2069–2079, 2017. → page 23

[5] 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)*, pages 8286–8290. IEEE, 2019. → pages 21, 71, 147

[6] B. Bayar and M. C. Stamm. Design principles of convolutional neural networks for multimedia forensics. *Electronic Imaging*, 2017(7):77–86, 2017. → page 11

[7] B. Bayar and M. C. Stamm. On the robustness of constrained convolutional neural networks to jpeg post-compression for image resampling detection. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2152–2156. IEEE, 2017. → page 23

[8] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr. Fully-convolutional siamese networks for object tracking. In *European*

conference on computer vision, pages 850–865. Springer, 2016. → pages 3, 16, 17, 119, 120

[9] A. N. Bhagoji, S. Chakraborty, P. Mittal, and S. Calo. Analyzing federated learning through an adversarial lens. In *International Conference on Machine Learning*, pages 634–643. PMLR, 2019. → page 21

[10] B. Biggio and F. Roli. Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognition*, 84:317–331, 2018. → page 24

[11] B. Biggio, G. Fumera, F. Roli, and L. Didaci. Poisoning adaptive biometric systems. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pages 417–425. Springer, 2012. → pages 20, 23

[12] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Šrndić, P. Laskov, G. Giacinto, and F. Roli. Evasion attacks against machine learning at test time. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 387–402. Springer, 2013. → pages 19, 24

[13] B. Biggio, P. Russu, L. Didaci, F. Roli, et al. Adversarial biometric recognition: A review on biometric system security from the adversarial machine-learning perspective. *IEEE Signal Processing Magazine*, 32(5): 31–41, 2015. → pages 19, 20, 22

[14] C. M. Bishop. *Pattern recognition and machine learning*. springer, 2006. → page 17

[15] J. A. Boyd, D. B. Harris, D. D. King, and H. Welch Jr. Electronic countermeasures. *elcm*, 1978. → page 24

[16] A. Brock, J. Donahue, and K. Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018. → pages 2, 8

[17] J. Bunk, J. H. Bappy, T. M. Mohammed, L. Nataraj, A. Flenner, B. Manjunath, S. Chandrasekaran, A. K. Roy-Chowdhury, and L. Peterson. Detection and localization of image forgeries using resampling features and deep learning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1881–1889. IEEE, 2017. → page 23

[18] E. Candès. The restricted isometry property and its implications for compressed sensing. *Comptes Rendus Mathematique*, 346(9-10):589–592, 2008. → page 35

[19] J. Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, 8(6):679–698, 1986. → page 97

[20] Y. Cao, X. Chen, L. Yao, X. Wang, and W. E. Zhang. Adversarial attacks and detection on reinforcement learning-based interactive recommender systems. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1669–1672, 2020. → page 23

[21] Z. Cao, M. Long, J. Wang, and P. S. Yu. Hashnet: Deep learning to hash by continuation. *arXiv preprint arXiv:1702.00758*, 2017. → pages 5, 11

[22] N. Carlini and D. Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 3–14, 2017. → page 148

[23] N. Carlini and D. Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. IEEE, 2017. → pages 92, 95

[24] N. Carlini, A. Athalye, N. Papernot, W. Brendel, J. Rauber, D. Tsipras, I. Goodfellow, and A. Madry. On evaluating adversarial robustness. *arXiv preprint arXiv:1902.06705*, 2019. → pages 19, 94

[25] J. Chen, X. Kang, Y. Liu, and Z. J. Wang. Median filtering forensics based on convolutional neural networks. *IEEE Signal Processing Letters*, 22(11): 1849–1853, Nov 2015. ISSN 1070-9908. doi:10.1109/LSP.2015.2438008. → pages 11, 23

[26] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017. → pages 3, 11

[27] S.-T. Chen, C. Cornelius, J. Martin, and D. H. P. Chau. Shapeshifter: Robust physical adversarial attack on faster r-cnn object detector. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 52–68. Springer, 2018. → pages 2, 21, 22, 23, 117, 147

[28] T. Chen, J. Liu, Y. Xiang, W. Niu, E. Tong, and Z. Han. Adversarial attack and defense in reinforcement learning-from ai security view. *Cybersecurity*, 2(1):11, 2019. → page 23

[29] X. Chen, X. Yan, F. Zheng, Y. Jiang, S.-T. Xia, Y. Zhao, and R. Ji. One-shot adversarial attacks on visual tracking with dual attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10176–10185, 2020. → page 121

[30] M. Cheng, T. Le, P.-Y. Chen, J. Yi, H. Zhang, and C.-J. Hsieh. Query-efficient hard-label black-box attack: An optimization-based approach. *arXiv preprint arXiv:1807.04457*, 2018. → page 22

[31] Y.-J. Chin and T. Berger. A software-only videocodec using pixelwise conditional differential replenishment and perceptual enhancements. *IEEE Transactions on Circuits and Systems for Video Technology*, 9(3):438–450, 1999. → page 96

[32] C.-H. Chou and Y.-C. Li. A perceptually tuned subband image coder based on the measure of just-noticeable-distortion profile. *IEEE Transactions on circuits and systems for video technology*, 5(6):467–476, 1995. → pages 92, 93, 95, 96, 97

[33] M. Cisse, P. Bojanowski, E. Grave, Y. Dauphin, and N. Usunier. Parseval networks: Improving robustness to adversarial examples. *arXiv preprint arXiv:1704.08847*, 2017. → page 147

[34] F. Croce and M. Hein. Sparse and imperceivable adversarial attacks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4724–4732, 2019. → page 95

[35] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on image processing*, 16(8):2080–2095, 2007. → pages 44, 148

[36] E. G. Dada, J. S. Bassi, H. Chiroma, A. O. Adetunmbi, O. E. Ajibuwa, et al. Machine learning for email spam filtering: review, approaches and open research problems. *Heliyon*, 5(6):e01802, 2019. → page 23

[37] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 886–893, 2005. → page 15

[38] N. Dalvi, P. Domingos, S. Sanghai, and D. Verma. Adversarial classification. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 99–108, 2004. → pages 19, 21, 23, 24

[39] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *Proceedings of the twentieth annual symposium on Computational geometry*, pages 253–262. ACM, 2004. → page 47

[40] R. Datta, D. Joshi, J. Li, and J. Z. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys (Csur)*, 40(2):5, 2008. → pages 5, 29

[41] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009. → pages 3, 8, 11, 34

[42] E. L. Denton, S. Chintala, a. szlam, and R. Fergus. Deep generative image models using a laplacian pyramid of adversarial networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1486–1494, 2015. → page 39

[43] G. S. Dhillon, K. Azizzadenesheli, Z. C. Lipton, J. Bernstein, J. Kossaifi, A. Khanna, and A. Anandkumar. Stochastic activation pruning for robust adversarial defense. *arXiv preprint arXiv:1803.01442*, 2018. → page 148

[44] X. Ding, Y. Wang, Z. Xu, W. J. Welch, and Z. J. Wang. Cc{gan}: Continuous conditional generative adversarial networks for image generation. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=PrzjugOsDeE. → page 8

[45] C. Dong, C. C. Loy, K. He, and X. Tang. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):295–307, 2016. → pages 11, 34, 40, 56

[46] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9185–9193, 2018. → pages 2, 21, 22, 23, 70, 73, 78, 147

[47] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9185–9193, 2018. → pages 92, 95, 99, 103, 104, 105, 106, 107, 110

[48] Y. Dong, H. Su, B. Wu, Z. Li, W. Liu, T. Zhang, and J. Zhu. Efficient decision-based black-box adversarial attacks on face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7714–7722, 2019. → pages 22, 23

[49] A. Dosovitskiy and T. Brox. Inverting visual representations with convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4829–4837, 2016. → page 31

[50] A. Dosovitskiy, J. T. Springenberg, M. Tatarchenko, and T. Brox. Learning to generate chairs, tables and cars with convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(4):692–705, 2017. → page 34

[51] L. Du, A. T. Ho, and R. Cong. Perceptual hashing for image authentication: A survey. *Signal Processing: Image Communication*, 81:115713, 2020. → page 145

[52] 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*, pages 7890–7899, 2020. → page 70

[53] S. Escalera and M. Weimer. *The NIPS'17 Competition: Building Intelligent Systems*. Springer, 2018. → pages 104, 105

[54] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song. Robust physical-world attacks on deep learning visual classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1625–1634, 2018. → page 117

[55] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song. Robust physical-world attacks on deep learning visual classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1625–1634, 2018. → pages 92, 95

[56] H. Fan, L. Lin, F. Yang, P. Chu, G. Deng, S. Yu, H. Bai, Y. Xu, C. Liao, and H. Ling. Lasot: A high-quality benchmark for large-scale single object tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5374–5383, 2019. → page 128

[57] X. Fang. Understanding deep learning via backtracking and deconvolution. *Journal of Big Data*, 4(1):40, 2017. → page 52

[58] M. Fiaz, A. Mahmood, S. Javed, and S. K. Jung. Handcrafted and deep trackers: Recent visual object tracking approaches and trends. *ACM Computing Surveys (CSUR)*, 52(2):1–44, 2019. → pages 14, 117

[59] J. Fridrich. *Steganography in digital media: principles, algorithms, and applications*. Cambridge University Press, 2009. → pages 23, 24

[60] K. Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36:193–202, 1980. → page 10

[61] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423, 2016. → page 123

[62] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *International Conference on Learning Representations*, 2019. → page 123

[63] A. Gionis, P. Indyk, R. Motwani, et al. Similarity search in high dimensions via hashing. In *Vldb*, pages 518–529, 1999. → pages 5, 29, 41, 47

[64] A. Gleave, M. Dennis, C. Wild, N. Kant, S. Levine, and S. Russell. Adversarial policies: Attacking deep reinforcement learning. *arXiv preprint arXiv:1905.10615*, 2019. → page 23

[65] I. Goodfellow. A research agenda: Dynamic models to defend against correlated attacks. *arXiv preprint arXiv:1903.06293*, 2019. → page 19

[66] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2672–2680, 2014. → pages 2, 7, 39, 68, 70

[67] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. → page 120

[68] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. The MIT Press, 2016. ISBN 0262035618, 9780262035613. → page 11

[69] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. *International Conference on Learning Representations*, 2015. → pages 2, 21, 22, 23, 70, 71, 72, 92, 95, 99, 100, 103, 105, 147

[70] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. *International Conference on Learning Representations*, 2015. → page 120

[71] K. Grosse, N. Papernot, P. Manoharan, M. Backes, and P. McDaniel. Adversarial examples for malware detection. In *European Symposium on Research in Computer Security*, pages 62–79. Springer, 2017. → page 21

[72] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, pages 5767–5777, 2017. → page 7

[73] C. Guo, M. Rana, M. Cisse, and L. Van Der Maaten. Countering adversarial images using input transformations. *arXiv preprint arXiv:1711.00117*, 2017. → page 148

[74] C. Guo, J. R. Gardner, Y. You, A. G. Wilson, and K. Q. Weinberger. Simple black-box adversarial attacks. *nternational Conferenceon Machine Learning*, 2019. → page 100

[75] M. Guo, Y. Yang, R. Xu, Z. Liu, and D. Lin. When nas meets robustness: In search of robust architectures against adversarial attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 631–640, 2020. → page 147

[76] Q. Guo, X. Xie, F. Juefei-Xu, L. Ma, Z. Li, W. Xue, W. Feng, and Y. Liu. Spark: Spatial-aware online incremental attack against visual tracking. *arXiv preprint arXiv:1910.08681*, 2019. → pages 21, 23, 117, 120, 121, 123

[77] S.-H. Han and C.-H. Chu. Content-based image authentication: current status, issues, and challenges. *International Journal of Information Security*, 9(1):19–32, 2010. → pages 3, 5, 29, 30, 49

[78] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. → pages 2, 3, 11, 13, 34, 67, 92, 104

[79] 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*, pages 770–778, 2016. → page 54

[80] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. → page 92

[81] D. Held, S. Thrun, and S. Savarese. Learning to track at 100 fps with deep regression networks. In *European Conference on Computer Vision*, pages 749–765. Springer, 2016. → pages 3, 121

[82] G. Hinton and R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504 – 507, 2006. → page 10

[83] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. → page 63

[84] G. E. Hinton, S. Osindero, and Y.-W. Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006. → page 10

[85] I. Hontsch and L. J. Karam. Locally adaptive perceptual image coding. *IEEE Transactions on Image Processing*, 9(9):1472–1483, 2000. → page 96

[86] N. J. Hopper, J. Langford, and L. Von Ahn. Provably secure steganography. In *Annual International Cryptology Conference*, pages 77–92. Springer, 2002. → page 24

[87] K. Hornik, M. Stinchcombe, H. White, et al. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989. → page 10

[88] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural

networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. → page 14

[89] D. Hu, L. Wang, W. Jiang, S. Zheng, and B. Li. A novel image steganography method via deep convolutional generative adversarial networks. *IEEE Access*, 6:38303–38314, 2018. → page 23

[90] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. → pages 12, 13

[91] L. Huang, A. D. Joseph, B. Nelson, B. I. Rubinstein, and J. D. Tygar. Adversarial machine learning. In *Proceedings of the 4th ACM workshop on Security and artificial intelligence*, pages 43–58, 2011. → pages 19, 21, 23

[92] L. Huang, C. Gao, Y. Zhou, C. Xie, A. L. Yuille, C. Zou, and N. Liu. Universal physical camouflage attacks on object detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 720–729, 2020. → pages 2, 22, 117

[93] D. Hubel and T. Wiesel. Receptive fields, binocular interaction, and functional architecture in the cat's visual cortex. *Journal of Physiology*, 160:106–154, 1962. → page 10

[94] D. H. Hubel and T. N. Wiesel. Receptive fields of single neurons in the cat's striate cortex. *Journal of Physiology*, 148:574–591, 1959. → page 10

[95] D. H. Hubel and T. N. Wiesel. Receptive fields and functional architecture of monkey striate cortex. *Journal of Physiology (London)*, 195:215–243, 1968. → page 10

[96] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and< 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016. → pages 13, 14

[97] A. Ilyas, L. Engstrom, A. Athalye, and J. Lin. Black-box adversarial attacks with limited queries and information. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*, July 2018. → page 22

[98] N. Inkawhich, W. Wen, H. H. Li, and Y. Chen. Feature space perturbations yield more transferable adversarial examples. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7066–7074, 2019. → pages 118, 123

[99] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017. → pages 2, 8

[100] M. Jagielski, A. Oprea, B. Biggio, C. Liu, C. Nita-Rotaru, and B. Li. Manipulating machine learning: Poisoning attacks and countermeasures for regression learning. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 19–35. IEEE, 2018. → page 20

[101] N. Jayant, J. Johnston, and R. Safranek. Signal compression based on models of human perception. *Proceedings of the IEEE*, 81(10):1385–1422, 1993. → pages 92, 94, 95, 96, 98

[102] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678. ACM, 2014. → page 41

[103] X. Jin, Y. Su, L. Zou, Y. Wang, P. Jing, and Z. J. Wang. Sparsity-based image inpainting detection via canonical correlation analysis with low-rank constraints. *IEEE Access*, 6:49967–49978, 2018. → page 2

[104] Z. Jin, C. Li, Y. Lin, and D. Cai. Density sensitive hashing. *IEEE Transactions on Cybernetics*, 44(8):1362–1371, 2014. → pages 30, 32, 35, 36

[105] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016. → page 123

[106] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016. → pages 2, 11

[107] D. Kahn. *The Codebreakers: The comprehensive history of secret communication from ancient times to the internet*. Simon and Schuster, 1996. → page 24

[108] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *International Conference on Learning Representations*, 2018. → pages 7, 8, 9, 68, 70, 74, 84

[109] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019. → pages 1, 2, 8, 68, 70, 74, 76, 87

[110] 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*, pages 8110–8119, 2020. → pages 1, 2, 8, 9, 68, 70, 74, 76

[111] H. Kato and T. Harada. Image reconstruction from bag-of-visual-words. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 955–962, 2014. → page 31

[112] S. Katzenbeisser and F. A. Petitcolas. Defining security in steganographic systems. In *Security and Watermarking of Multimedia Contents IV*, volume 4675, pages 50–56. International Society for Optics and Photonics, 2002. → page 24

[113] F. Khelifi and A. Bouridane. Perceptual video hashing for content identification and authentication. *IEEE Transactions on Circuits and Systems for Video Technology*, 29(1):50–67, 2017. → page 146

[114] D. Kim, H.-U. Jang, S.-M. Mun, S. Choi, and H.-K. Lee. Median filtered image restoration and anti-forensics using adversarial networks. *IEEE Signal Processing Letters*, 25(2):278–282, 2017. → pages 8, 23

[115] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR), May*, 2015. → page 56

[116] J. Kodovsky, J. Fridrich, and V. Holub. Ensemble classifiers for steganalysis of digital media. *IEEE Transactions on Information Forensics and Security*, 7(2):432–444, 2012. → page 68

[117] W. Kong and W.-J. Li. Isotropic hashing. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1646–1654, 2012. → page 30

[118] P. Korshunov and S. Marcel. Deepfakes: a new threat to face recognition? assessment and detection. *arXiv preprint arXiv:1812.08685*, 2018. → page 2

[119] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1097–1105, 2012. → pages 3, 11, 12, 13, 34, 67, 77, 92

[120] B. Kulis and T. Darrell. Learning to hash with binary reconstructive embeddings. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1042–1050, 2009. → pages 5, 30, 32, 35, 36

[121] B. Kulis and K. Grauman. Kernelized locality-sensitive hashing for scalable image search. In *IEEE 12th International Conference on Computer Vision (ICCV)*, pages 2130–2137, 2009. → pages 32, 41, 47, 48

[122] B. Kulis, P. Jain, and K. Grauman. Fast similarity search for learned metrics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(12):2143–2157, 2009. → page 47

[123] K. Kulkarni, S. Lohit, P. Turaga, R. Kerviche, and A. Ashok. Reconnet: Non-iterative reconstruction of images from compressively sensed measurements. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 449–458, 2016. → pages 11, 35

[124] A. Kurakin, I. Goodfellow, and S. Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016. → pages 23, 117

[125] A. Kurakin, I. Goodfellow, and S. Bengio. Adversarial examples in the physical world. *International Conference on Learning Representations Workshop*, 2017. → page 95

[126] H. Lai, Y. Pan, Y. Liu, and S. Yan. Simultaneous feature learning and hash coding with deep neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3270–3278, 2015. → pages 5, 11, 30

[127] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pages 2278–2324, 1998. → pages 10, 11

[128] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553): 436–444, 2015. → page 11

[129] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 105–114, July 2017. → pages 8, 39

[130] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu. High performance visual tracking with siamese region proposal network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8971–8980, 2018. → pages 3, 16, 17, 117, 120

[131] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan. Siamrpn++: Evolution of siamese visual tracking with very deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4282–4291, 2019. → pages 2, 3, 16, 17, 117, 120, 128

[132] H. Li, H. Chen, B. Li, and S. Tan. Can forensic detectors identify gan generated images? In *2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 722–727. IEEE, 2018. → pages 3, 68

[133] H. Li, B. Li, S. Tan, and J. Huang. Identification of deep network generated images using disparities in color components. *Signal Processing*, page 107616, 2020. → pages 69, 77, 89

[134] Y. Li and J. Zhou. Fast and effective image copy-move forgery detection via hierarchical feature point matching. *IEEE Transactions on Information Forensics and Security*, 14(5):1307–1322, 2018. → page 2

[135] Y. Li, X. Yang, P. Sun, H. Qi, and S. Lyu. Celeb-df: A large-scale challenging dataset for deepfake forensics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3207–3216, 2020. → page 92

[136] B. Lim, S. Son, H. Kim, S. Nah, and K. Mu Lee. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 136–144, 2017. → page 54

[137] K. Lin, H. F. Yang, J. H. Hsiao, and C. S. Chen. Deep learning of binary hash codes for fast image retrieval. In *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 27–35, June 2015. doi:10.1109/CVPRW.2015.7301269. → pages 5, 11

[138] W. Lin and C.-C. J. Kuo. Perceptual visual quality metrics: A survey. *Journal of visual communication and image representation*, 22(4):297–312, 2011. → pages 93, 94, 96

[139] B. Liu and C.-M. Pun. Deep fusion network for splicing forgery localization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 0–0, 2018. → page 2

[140] C. Liu, B. Zoph, M. Neumann, J. Shlens, W. Hua, L.-J. Li, L. Fei-Fei, A. Yuille, J. Huang, and K. Murphy. Progressive neural architecture search. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 19–34, 2018. → page 14

[141] H. Liu, R. Wang, S. Shan, and X. Chen. Deep supervised hashing for fast image retrieval. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2064–2072, 2016. → page 146

[142] Q. Liu, P. Li, W. Zhao, W. Cai, S. Yu, and V. C. Leung. A survey on security threats and defensive techniques of machine learning: A data driven view. *IEEE access*, 6:12103–12117, 2018. → pages 19, 20

[143] W. Liu, J. Wang, S. Kumar, and S.-F. Chang. Hashing with graphs. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 1–8. Citeseer, 2011. → pages 5, 30

[144] W. Liu, J. Wang, R. Ji, Y.-G. Jiang, and S.-F. Chang. Supervised hashing with kernels. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2074–2081, 2012. → pages 5, 29, 30

[145] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016. → page 3

[146] X. Liu, S. Si, X. Zhu, Y. Li, and C.-J. Hsieh. A unified framework for data poisoning attack to graph-based semi-supervised learning. *arXiv preprint arXiv:1910.14147*, 2019. → page 23

[147] Y. Liu, X. Chen, C. Liu, and D. Song. Delving into transferable adversarial examples and black-box attacks. *International Conference on Learning Representations*, 2017. → pages 22, 70

[148] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. → pages 3, 11

[149] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. → page 15

[150] J. Lu, T. Issaranon, and D. Forsyth. Safetynet: Detecting and rejecting adversarial examples robustly. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 446–454, 2017. → page 148

[151] A. Lucas, M. Iliadis, R. Molina, and A. K. Katsaggelos. Using deep neural networks for inverse problems in imaging: beyond analytical methods. *IEEE Signal Processing Magazine*, 35(1):20–36, 2018. → page 67

[152] B. Luo, Y. Liu, L. Wei, and Q. Xu. Towards imperceptible and robust adversarial example attacks against neural networks. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. → page 95

[153] Y. Luo, H. Zi, Q. Zhang, and X. Kang. Anti-forensics of jpeg compression using generative adversarial networks. In *2018 26th European Signal Processing Conference (EUSIPCO)*, pages 952–956. IEEE, 2018. → page 23

[154] X. Lv and Z. J. Wang. Perceptual image hashing based on shape contexts and local feature points. *IEEE Transactions on Information Forensics and Security*, 7(3):1081–1093, 2012. → pages 3, 5, 6, 29, 30, 35, 49, 145

[155] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. *International Conference on Learning Representations*, 2018. → pages 70, 71, 73

[156] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. *International Conference on Learning Representations*, 2018. → pages 92, 95, 147

[157] F. Marra, D. Gragnaniello, and L. Verdoliva. On the vulnerability of deep learning to adversarial attacks for camera model identification. *Signal Processing: Image Communication*, 65:240–248, 2018. → page 71

[158] 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)*, pages 506–511. IEEE, 2019. → page 69

[159] S. M. Marvasti-Zadeh, L. Cheng, H. Ghanei-Yakhdan, and S. Kasaei. Deep learning for visual tracking: A comprehensive survey. *arXiv preprint arXiv:1912.00535*, 2019. → pages 14, 117

[160] S. McCloskey and M. Albright. Detecting gan-generated imagery using color cues. *arXiv preprint arXiv:1812.08247*, 2018. → pages 69, 77

[161] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov. Exploiting unintended feature leakage in collaborative learning. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 691–706. IEEE, 2019. → page 21

[162] D. Meng and H. Chen. Magnet: a two-pronged defense against adversarial examples. In *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, pages 135–147, 2017. → page 148

[163] J. H. Metzen, T. Genewein, V. Fischer, and B. Bischoff. On detecting adversarial perturbations. *arXiv preprint arXiv:1702.04267*, 2017. → page 148

[164] Y. Mirsky and W. Lee. The creation and detection of deepfakes: A survey. *arXiv preprint arXiv:2004.11138*, 2020. → page 2

[165] A. Mittal, R. Soundararajan, and A. C. Bovik. Making a "completely blind" image quality analyzer. *IEEE Signal processing letters*, 20(3):209–212, 2012. → pages 79, 104

[166] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida. Spectral normalization for generative adversarial networks. *International Conference on Learning Representations*, 2018. → page 70

[167] H. Mo, B. Chen, and W. Luo. Fake faces identification via convolutional neural network. In *Proceedings of the 6th ACM Workshop on Information Hiding and Multimedia Security*, pages 43–47. ACM, 2018. → pages 3, 68, 69, 77

[168] V. Monga et al. Robust and secure image hashing via non-negative matrix factorizations. *IEEE Transactions on Information Forensics and Security*, 2 (3):376–390, 2007. → pages 5, 30, 32, 49

[169] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard. Universal adversarial perturbations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1765–1773, 2017. → page 22

[170] L. Morra and F. Lamberti. Benchmarking unsupervised near-duplicate image detection. *Expert Systems with Applications*, 135:313–326, 2019. → page 144

[171] M. Muller, A. Bibi, S. Giancola, S. Alsubaihi, and B. Ghanem. Trackingnet: A large-scale dataset and benchmark for object tracking in the wild. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 300–317, 2018. → page 128

[172] K. P. Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012. → page 17

[173] Y. Nagai, Y. Uchida, S. Sakazawa, and S. Satoh. Digital watermarking for deep neural networks. *International Journal of Multimedia Information Retrieval*, 7(1):3–16, 2018. → page 23

[174] A. Netravali. *Digital pictures: representation and compression*. Springer Science & Business Media, 2013. → page 98

[175] X. Nie, W. Jing, C. Cui, J. Zhang, L. Zhu, and Y. Yin. Joint multi-view hashing for large-scale near-duplicate video retrieval. *IEEE Transactions on Knowledge and Data Engineering*, 2019. → page 144

[176] M. Norouzi, D. J. Fleet, and R. R. Salakhutdinov. Hamming distance metric learning. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1061–1069, 2012. → pages 29, 30

[177] E. Nowroozi, A. Dehghantanha, R. M. Parizi, and K.-K. R. Choo. A survey of machine learning techniques in adversarial image forensics. *Computers & Security*, page 102092, 2020. → page 21

[178] H. Palangi, R. Ward, and L. Deng. Distributed compressive sensing: A deep learning approach. *IEEE Transactions on Signal Processing*, 64(17): 4504–4518, Sept 2016. ISSN 1053-587X. → page 35

[179] T. Pang, C. Du, Y. Dong, and J. Zhu. Towards robust detection of adversarial examples. In *Advances in Neural Information Processing Systems*, pages 4579–4589, 2018. → page 147

[180] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami. The limitations of deep learning in adversarial settings. In *2016 IEEE European symposium on security and privacy (EuroS&P)*, pages 372–387. IEEE, 2016. → pages 92, 95

[181] N. Papernot, P. McDaniel, A. Sinha, and M. Wellman. Towards the science of security and privacy in machine learning. *arXiv preprint arXiv:1611.03814*, 2016. → pages 19, 20

[182] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 582–597. IEEE, 2016. → page 147

[183] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems*, pages 8026–8037, 2019. → page 128

[184] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8024–8035, 2019. → pages 13, 103

[185] A. Peng, H. Zeng, X. Lin, and X. Kang. Countering anti-forensics of image resampling. In *2015 IEEE International Conference on Image Processing (ICIP)*, pages 3595–3599. IEEE, 2015. → page 23

[186] A. Piva. An overview on image forensics. *ISRN Signal Processing*, 2013, 2013. → page 69

[187] Z. Qian and X. Zhang. Improved anti-forensics of jpeg compression. *Journal of Systems and Software*, 91:100–108, 2014. → pages 23, 69

[188] Y. Qin, N. Carlini, I. Goodfellow, G. Cottrell, and C. Raffel. Imperceptible, robust, and targeted adversarial examples for automatic speech recognition. *nternational Conferenceon Machine Learning*, 2019. → pages 92, 95

[189] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *International Conference on Learning Representations*, 2016. → pages 8, 9, 70, 77

[190] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. → page 3

[191] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. → pages 2, 3, 92

[192] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. → page 120

[193] M. Rigaki and S. Garcia. A survey of privacy attacks in machine learning. *arXiv preprint arXiv:2007.07646*, 2020. → page 21

[194] 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*, pages 234–241. Springer, 2015. → pages 3, 11

[195] R. Rosenblatt. The perceptron–a perceiving and recognizing automaton. *Technical Report, Cornell Aeronautical Laboratory Report*, 1957. → page 9

[196] 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 International Conference on Computer Vision*, pages 1–11, 2019. → page 2

[197] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986. → page 10

[198] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3): 211–252, 2015. → page 104

[199] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, 2018. → pages 14, 77

[200] R. Satter. Experts: Spy used ai-generated face to connect with targets. *The Washington Post*, 2019. → pages 1, 68

[201] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio. Robust object recognition with cortex-like mechanisms. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(3):411–426, Mar. 2007. ISSN 0162-8828. doi:10.1109/TPAMI.2007.56. URL http://dx.doi.org/10.1109/TPAMI.2007.56. → page 10

[202] S. D. Servetto, C. I. Podilchuk, and K. Ramchandran. Capacity issues in digital image watermarking. In *Proceedings 1998 International Conference on Image Processing. ICIP98 (Cat. No. 98CB36269)*, volume 1, pages 445–449. IEEE, 1998. → page 24

[203] S. Shalev-Shwartz and S. Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014. → page 17

[204] M. Sharif, S. Bhagavatula, L. Bauer, and M. K. Reiter. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *Proceedings of the 2016 acm sigsac conference on computer and communications security*, pages 1528–1540, 2016. → pages 22, 23

[205] M. Sharif, S. Bhagavatula, L. Bauer, and M. K. Reiter. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *Proceedings of the 2016 acm sigsac conference on computer and communications security*, pages 1528–1540, 2016. → page 126

[206] Y. Sharma, G. W. Ding, and M. Brubaker. On the effectiveness of low frequency perturbations. *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, 2019. → page 101

[207] H. R. Sheikh and A. C. Bovik. Image information and visual quality. *IEEE Transactions on Image Processing*, 15(2):430–444, Feb 2006. ISSN 1057-7149. doi:10.1109/TIP.2005.859378. → page 57

[208] R. Shokri, M. Stronati, C. Song, and V. Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18. IEEE, 2017. → page 21

[209] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. → pages 3, 11, 12, 13, 34, 68, 77

[210] A. Sinha, D. Kar, and M. Tambe. Learning adversary behavior in security games: A pac model perspective. *arXiv preprint arXiv:1511.00043*, 2015. → page 19

[211] L. N. Smith and N. Topin. Deep convolutional neural network design patterns. *arXiv preprint arXiv:1611.00847*, 2016. → page 39

[212] Y. Song, T. Kim, S. Nowozin, S. Ermon, and N. Kushman. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. *arXiv preprint arXiv:1710.10766*, 2017. → page 148

[213] M. C. Stamm, S. K. Tjoa, W. S. Lin, and K. R. Liu. Anti-forensics of jpeg compression. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1694–1697. IEEE, 2010. → page 23

[214] M. C. Stamm, S. K. Tjoa, W. S. Lin, and K. R. Liu. Anti-forensics of jpeg compression. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1694–1697. IEEE, 2010. → page 24

[215] M. C. Stamm, M. Wu, and K. R. Liu. Information forensics: An overview of the first decade. *IEEE access*, 1:167–200, 2013. → page 69

[216] R. C. Streijl, S. Winkler, and D. S. Hands. Mean opinion score (mos) revisited: methods and applications, limitations and alternatives. *Multimedia Systems*, 22(2):213–227, 2016. → pages 104, 105

[217] O. Suciu, S. E. Coull, and J. Johns. Exploring adversarial examples in malware detection. In *2019 IEEE Security and Privacy Workshops (SPW)*, pages 8–14. IEEE, 2019. → page 21

[218] M. Sun, J. Tang, H. Li, B. Li, C. Xiao, Y. Chen, and D. Song. Data poisoning attack against unsupervised node embedding methods. *arXiv preprint arXiv:1810.12881*, 2018. → pages 21, 23

[219] A. Swaminathan, Y. Mao, and M. Wu. Robust and secure image hashing. *IEEE Transactions on Information Forensics and Security*, 1(2):215–230, June 2006. ISSN 1556-6013. → pages 3, 5, 6, 30

[220] M. D. Swanson, B. Zhu, and A. H. Tewfik. Transparent robust image watermarking. In *Proceedings of 3rd IEEE International Conference on Image Processing*, volume 3, pages 211–214. IEEE, 1996. → pages 23, 24

[221] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014. URL http://arxiv.org/abs/1312.6199. → page 92

[222] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *International Conference on Learning Representations*, 2014. → page 70

[223] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *International Conference on Learning Representations*, 2014. → pages 24, 117, 120

[224] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015. → pages 11, 13, 34

[225] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016. → pages 3, 12, 13, 103

[226] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-first AAAI conference on artificial intelligence*, 2017. → pages 12, 104

[227] M. Tan and Q. V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *arXiv preprint arXiv:1905.11946*, 2019. → page 14

[228] 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*, pages 81–87. ACM, 2018. → page 68

[229] S. Thys, W. Van Ranst, and T. Goedemé. Fooling automated surveillance cameras: adversarial patches to attack person detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. → page 2

[230] S. Thys, W. Van Ranst, and T. Goedemé. Fooling automated surveillance cameras: adversarial patches to attack person detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. → pages 2, 21, 23

[231] R. Tolosana, R. Vera-Rodriguez, J. Fierrez, A. Morales, and J. Ortega-Garcia. Deepfakes and beyond: A survey of face manipulation and fake detection. *arXiv preprint arXiv:2001.00179*, 2020. → page 2

[232] V. Tolpegin, S. Truex, M. E. Gursoy, and L. Liu. Data poisoning attacks against federated learning systems. In *European Symposium on Research in Computer Security*, pages 480–501. Springer, 2020. → page 21

[233] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel. Ensemble adversarial training: Attacks and defenses. *International Conference on Learning Representations*, 2018. → page 104

[234] J. A. Tropp. Algorithms for simultaneous sparse approximation: Part II: Convex relaxation. *Signal Process.*, 86(3):589–602, Mar. 2006. ISSN 0165-1684. → page 35

[235] J. A. Tropp, A. C. Gilbert, and M. J. Strauss. Algorithms for simultaneous sparse approximation: Part I: Greedy pursuit. *Signal Process.*, 86(3): 572–588, Mar. 2006. ISSN 0165-1684. → page 35

[236] Y. Tsuzuku and I. Sato. On the structural sensitivity of deep convolutional networks to the directions of fourier basis functions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 51–60, 2019. → page 100

[237] L. G. Valiant. A theory of the learnable. *Communications of the ACM*, 27 (11):1134–1142, 1984. → page 18

[238] L. Verdoliva. Media forensics and deepfakes: an overview. *arXiv preprint arXiv:2001.06564*, 2020. → page 2

[239] Y. Vorobeychik and M. Kantarcioglu. Adversarial machine learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 12(3): 1–169, 2018. → page 19

[240] J. Wang, W. Liu, S. Kumar, and S.-f. Chang. Learning to hash for indexing big data - a survey. *Proceedings of the IEEE*, 104(1):34–57, Jan 2016. ISSN 0018-9219. → pages 5, 29, 47

[241] Q. Wang, L. Zhang, L. Bertinetto, W. Hu, and P. H. Torr. Fast online object tracking and segmentation: A unifying approach. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1328–1338, 2019. → pages 16, 17, 92

[242] Q. Wang, L. Zhang, L. Bertinetto, W. Hu, and P. H. Torr. Fast online object tracking and segmentation: A unifying approach. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1328–1338, 2019. → pages 3, 16, 17, 117, 120, 128

[243] 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 Conference on Computer Vision and Pattern Recognition, volume 7, 2020. → pages 3, 69, 77

[244] X. Wang, K. Pang, X. Zhou, Y. Zhou, L. Li, and J. Xue. A visual model-based perceptual image hash for content authentication. *IEEE Transactions on Information Forensics and Security*, 10(7):1336–1349, 2015. → pages 29, 49

[245] Y. Wang, H. Palangi, Z. J. Wang, and H. Wang. RevHashNet: Perceptually de-hashing real-valued image hashes for similarity retrieval. *Signal processing: Image communication*, 68:68–75, 2018. → pages 51, 56, 57, 58, 67, 92

[246] Z. Wang, E. P. Simoncelli, and A. C. Bovik. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, volume 2, pages 1398–1402. Ieee, 2003. → page 104

[247] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. → pages 46, 55, 57

[248] P. Weinzaepfel, H. Jégou, and P. Pérez. Reconstructing an image from its local descriptors. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 337–344, 2011. → page 31

[249] Y. Weiss, A. Torralba, and R. Fergus. Spectral hashing. In *Advances in Neural Information Processing Systems*, pages 1753–1760, 2009. → page 61

[250] Y. Weiss, A. Torralba, and R. Fergus. Spectral hashing. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1753–1760, 2009. → pages 5, 30, 32, 35, 36

[251] L. Weng, L. Amsaleg, A. Morton, and S. Marchand-Maillet. A privacy-preserving framework for large-scale content-based information retrieval. *IEEE Transactions on Information Forensics and Security*, 10(1): 152–167, 2015. → page 30

[252] R. R. Wiyatno and A. Xu. Physical adversarial textures that fool visual object tracking. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4822–4831, 2019. → pages 2, 121

[253] X. Wu, X. Wang, X. Zhou, and S. Jian. Sta: Adversarial attacks on siamese trackers. *arXiv preprint arXiv:1909.03413*, 2019. → page 121

[254] Y. Wu, J. Lim, and M.-H. Yang. Online object tracking: A benchmark. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2411–2418, 2013. → pages 14, 15, 117

[255] Y. Wu, W. Abd-Almageed, and P. Natarajan. Busternet: Detecting copy-move image forgery with source/target localization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 168–184, 2018. → page 2

[256] Z.-H. Wu, M. C. Stamm, and K. R. Liu. Anti-forensics of median filtering. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3043–3047. IEEE, 2013. → page 23

[257] C. Xie, J. Wang, Z. Zhang, Y. Zhou, L. Xie, and A. Yuille. Adversarial examples for semantic segmentation and object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1369–1378, 2017. → pages 2, 117, 120, 147

[258] C. Xie, Z. Zhang, Y. Zhou, S. Bai, J. Wang, Z. Ren, and A. L. Yuille. Improving transferability of adversarial examples with input diversity. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2730–2739, 2019. → pages 92, 93, 95, 99, 103, 104, 105, 108

[259] B. Yan, D. Wang, H. Lu, and X. Yang. Cooling-shrinking attack: Blinding the tracker with imperceptible noises. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 990–999, 2020. → pages 21, 22, 23, 117, 120, 121, 123, 125, 147

[260] C.-P. Yan, C.-M. Pun, and X.-C. Yuan. Quaternion-based image hashing for adaptive tampering localization. *IEEE Transactions on Information Forensics and Security*, 11(12):2664–2677, 2016. → page 30

[261] X. Yan, X. Chen, Y. Jiang, S.-T. Xia, Y. Zhao, and F. Zheng. Hijacking tracker: A powerful adversarial attack on visual tracking. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2897–2901. IEEE, 2020. → pages 2, 22, 23, 120, 121, 147

[262] J. Yang, J. Wright, T. S. Huang, and Y. Ma. Image super-resolution via sparse representation. *IEEE transactions on image processing*, 19(11): 2861–2873, 2010. → pages 40, 56

[263] X. Yang, W. Ling, Z. Lu, E. P. Ong, and S. Yao. Just noticeable distortion model and its applications in video coding. *Signal Processing: Image Communication*, 20(7):662–680, 2005. → pages 92, 94, 95, 96, 97

[264] N. Yu, L. S. Davis, and M. Fritz. Attributing fake images to gans: Learning and analyzing gan fingerprints. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7556–7566, 2019. → page 69

[265] X. Yuan, P. He, Q. Zhu, and X. Li. Adversarial examples: Attacks and defenses for deep learning. *IEEE transactions on neural networks and learning systems*, 30(9):2805–2824, 2019. → page 19

[266] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. *CoRR*, abs/1311.2901, 2013. → page 11

[267] Y. Zhan, Y. Chen, Q. Zhang, and X. Kang. Image forensics based on transfer learning and convolutional neural network. In *Proceedings of the 5th ACM Workshop on Information Hiding and Multimedia Security*, pages 165–170. ACM, 2017. → page 11

[268] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena. Self-attention generative adversarial networks. In *International Conference on Machine Learning*, pages 7354–7363. PMLR, 2019. → page 70

[269] L. Zhang, L. Zhang, X. Mou, and D. Zhang. FSIM: A feature similarity index for image quality assessment. *IEEE Transactions on Image Processing*, 20(8):2378–2386, 2011. → page 57

[270] L. Zhang, L. Zhang, X. Mou, and D. Zhang. Fsim: A feature similarity index for image quality assessment. *IEEE transactions on Image Processing*, 20(8):2378–2386, 2011. → pages 79, 104

[271] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. → page 79

[272] T. Zhang and Z. Zhu. Interpreting adversarially trained convolutional neural networks. *International conference on machine learning*, 2019. → page 123

[273] X. Zhang, W. Lin, and P. Xue. Improved estimation for just-noticeable visual distortion. *Signal Processing*, 85(4):795–808, 2005. → pages 93, 94, 96, 98, 104

[274] X. Zhang, X. Zhou, M. Lin, and J. Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6848–6856, 2018. → page 14

[275] Y. Zhang, R. Jia, H. Pei, W. Wang, B. Li, and D. Song. The secret revealer: generative model-inversion attacks against deep neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 253–261, 2020. → page 21

[276] Y. Zhao, H. Zhu, R. Liang, Q. Shen, S. Zhang, and K. Chen. Seeing isn't believing: Towards more robust adversarial attack against real world object detectors. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pages 1989–2004, 2019. → pages 23, 118

[277] K. Zhou, J. C. Doyle, K. Glover, et al. *Robust and optimal control*, volume 40. Prentice hall New Jersey, 1996. → page 24

[278] P. Zhou, X. Han, V. I. Morariu, and L. S. Davis. Learning rich features for image manipulation detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1053–1061, 2018. → page 2

[279] H. Zhu, M. Long, J. Wang, and Y. Cao. Deep hashing network for efficient similarity retrieval. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, pages 2415–2421. AAAI Press, 2016. → pages 5, 11, 30

[280] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017. → pages 2, 8

[281] D. Zügner, A. Akbarnejad, and S. Günnemann. Adversarial attacks on neural networks for graph data. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2847–2856, 2018. → pages 21, 23